



Gateway 6.17.3 User Guide



Copyright ©2023 Axway- All rights reserved.

No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, Axway.

This document, provided for informational purposes only, may be subject to significant modification. The descriptions and information in this document may not necessarily accurately represent or reflect the current or planned functions of this product. Axway may change this publication, the product described herein, or both. These changes will be incorporated in new versions of this document. Axway does not warrant that this document is error free.

Axway recognizes the rights of the holders of all trademarks used in its publications.

The documentation may provide hyperlinks to third-party web sites or access to third-party content. Links and access to these sites are provided for your convenience only. Axway does not control, endorse or guarantee content found in such sites. Axway is not responsible for any content, associated links, resources or services associated with a third-party site.

Axway shall not be liable for any loss or damage of any sort associated with your use of third-party content.

Table Of Contents

What's new in this version	16
Gateway 6.17.3 SP10	16
Gateway 6.17.3 SP9	16
Gateway 6.17.3 SP5	17
Gateway 6.17.3 SP4	17
Gateway 6.17.3 SP3	18
Gateway 6.17.3 SP2	18
Gateway 6.17.3 SP1	20
Gateway 6.17.3	20
Gateway 6.17.2 SP2	21
Gateway 6.17.2	22
Gateway 6.17.1	23
Gateway 6.16.1	24
Gateway 6.16.0	25
 Overview	 27
Definition of Axway Gateway	27
Principal functions	27
Network Architecture	28
Local Area Networks	28
Linking your local network to the Internet	29
Using EDIINT protocols	31
Linking your local network to a WAN	32
Interconnecting applications over multiple local and remote networks	33
File Transfers and Transfer Requests	34
File Transfers	35
Transfer Requests	35
Transfer-related objects	36
Transfer Sites	37
Site characteristics	37
Dynamic Sites	38
Applications	39
Specifying file characteristics for Transfers	41
Default Applications	41
Connection Gates	42
CGateGroups	42
CGates	42
SGates and SGateGroups	43

Diffusion Lists	43
Virtual File Directories	44
Example	45
Trading Partners	46
Event management	48
Triggering Events	48
Triggered Processes	48
Configuring Gateway to trigger processes	49
Routing	50
Models	51
Transfer Models	51
XMS Models	53
Security Features	53
Monitoring Gateway	53
Mailbox	53
Connections and Connected Users	54
Logs	54
Audit	54
Traces	54
Statistics	55
Process monitoring	55
Sentinel	56
DMZ deployment	56
Axway Secure Relay (formerly XSR)	57
Paired Axway Gateways	58
User interfaces	61
GUI and command line interfaces	61
Choosing a user interface	61
Using the Graphical User Interface (GUI)	62
About the Gateway GUI	62
Starting and stopping the GUI	62
How to set and modify your password	63
About the main window	63
Online commands	70
Overview	70
Before using the Command Line Interface	71
Online commands and utilities	71
Syntax and conventions	72
Return values and error reporting	75
Displaying online help	77
Index of online commands	82
Index of tasks	110

Catalog of online commands	124
Online commands in Perl	147
Configuration	152
Overview	152
Platform support	152
Network clients	153
Configuration: procedure	154
About configuration	154
Configuring Gateway	154
Configuring Gateway parameters	155
Gateway parameters	155
Gateway > Database	165
Gateway > Connection control	170
Trace file archiving parameters	174
Configuring trading Partner parameters	175
Configuration: connectivity parameters	176
Activating/deactivating protocols and connectors	176
Configure Passport with SecureRelay	176
Connectivity > Navigator server	177
Connectivity > Secure Relay	178
Configuration of Secure Relay	180
About configuring protocols	180
Supported protocols and network interfaces	180
About configuring protocols	181
Configuring protocols	182
IPv6 support	187
Configuring TCP/IP parameters	188
Configuring protocols: SAP parameters	194
Configuring FTP options	196
Configuring protocols: HTTP options	196
Configuring protocols: TCP advanced options	198
Configuring X.400 O/R address	200
Configuring protocols: X.400 advanced settings	203
Configuring SNA LU 6.2 (on Windows)	209
Using RAS	215
Network configuration for UNIX platforms	220
Configuring connectors	221
Configuring connectors	221
Connectivity > Axway Connectivity > Integrator	222
Connectivity > Axway Connectivity > PassPort	223
Connectivity > Axway Connectivity > Sentinel	227
Connectivity > Axway Connectivity > Messaging	228

Connectivity > Axway Connectivity > JMS	228
Connectivity > Axway Connectivity > MFT Navigator	228
Connectivity > Axway Connectivity > SWIFTNet	229
Connectivity > Axway Connectivity > X.400	235
Modifying conffile parameters	236
Modifying pelsetup.def	237
SWIFTNet parameters in the pelsetup.def file	237
Configure where log files are stored in the pelsetup.def file	238
Application Service Profiles (ASP)	239
Introduction	239
ASP package	240
ASP lifecycle	240
More information	240
Relationship Management Application (RMA)	241
Introduction	241
Gateway services for Gateway RMA	241
Summary of Gateway RMA features	241
Housekeeping	241
Access control	242
Checking and importing RMA distribution files	242
Importing an ASP package	242
RMA filtering exit	242
RMA Server administration commands	242
About RMA	242
Introduction	242
RMA authorizations	243
RMA filtering	248
RmaQuery and RmaAnswer	249
More information	249
Importing an ASP package	249
Introduction	249
Downloading the latest ASP package	249
Importing (or updating) ASP profiles	250
Housekeeping: starting and stopping the RMA Server	251
About RMA housekeeping scripts	251
Running the RMA housekeeping scripts	251
Alternative methods of starting the RMA Server on Windows	252
Installing (or removing) the RMA Server Windows service	252
Starting and stopping the RMA Server in Windows service mode	252
Purging log, audit and trace files	253
Configuring access control for Gateway RMA	253
Overview	253

Importing required data into PassPort	253
Creating users in PassPort	255
Updating PassPort properties	255
Checking and importing an RMA distribution file	255
Introduction	255
Checking an RMA distribution file	256
Importing an RMA distribution file	257
Using the RMA filtering exit	257
Introduction	257
Prerequisites	257
Contents of the user exit configuration file	258
Logging	258
RMA Server administration commands	258
Introduction	259
Configuration file	259
Command syntax	260
Command line actions	260
Parameter list	261
Command line errors	262
Managing transfers	263
About transfers and transfer requests	263
Transfer types	263
Transfer objects	263
Submitting transfers	268
About transfers	268
Transfer requests	269
Application messages	299
Acknowledgment messages	303
Transfer icons	306
Transfer states in Gateway	309
Working with Transfers (command line)	314
Virtual directories	376
About Virtual File Directories	376
Working with Virtual File Directories	383
Working with Virtual File Directories (command line)	387
Parameters (applications and models)	423
Parameters (applications and models)	424
About applications	424
Managing models	459
Monitoring Transfers	513
Mailbox	513
Connections and Connected Users	514
Logs	515

Statistics	515
Automatic maintenance operations	515
Axway Sentinel	515
Monitoring Transfers	516
Viewing and managing Mailbox contents	518
Monitoring Connections	535
Monitoring Connected Users	538
Automating maintenance operations	542
Viewing Mailbox contents (command line)	543
Log files	578
Transfer examples	597
Executing a basic PeSIT transfer	597
About this example	597
Prerequisites	598
Executing the Transfer Request	602
Transfer results	603
Executing a basic FTP transfer	604
About this example	605
Prerequisites	605
Configuring the client/sender (machine A)	605
Configuring the server/receiver (machine B)	606
Executing the FTP transfer to server	609
Transfer results	610
Executing an SFTP transfer	611
About this example	612
Prerequisites	613
Configuring the SFTP client/sender (machine A)	613
Configuring the SFTP server/receiver (machine B)	616
Executing the SFTP transfer	620
Transfer results	621
Validating keys	622
Executing an FTP transfer using TLS and certificates	623
About this example	623
Prerequisites	623
Configuring the client/sender (machine A)	623
Configuring the server/receiver (machine B)	626
Executing the FTP transfer	629
Transfer results	630
Validating certificates	631
Executing an EDIINT AS2 transfer	632
About this example	632
Before you execute this example	634
Configuring the EDIINT distributor machine	634

Submitting an AS2 Transfer Request	640
After the transfer	641
Managing partners	642
Managing site objects	642
About Site objects	642
Working with Local Sites	645
Remote Sites	649
Command line operations for Local Sites	697
Proxy objects	752
Proxy objects introduction	752
Proxies and Gateway	752
Working with Proxies	754
Working with Proxies (command line)	758
	764
Managing Connection Gate objects	764
About CGates and CGateGroups	764
CGates: Protocol connection phase	767
CGates: Transfer phase	774
About SGates	778
Working with CGates and CGateGroups	781
Working with SGates	794
Command line operations	796
About diffusion lists	819
Introduction	819
Diffusion List object	819
Using a template site with a diffusion list	819
Example	820
About diffusion lists	821
Working with Diffusion Lists	822
Working with Diffusion Lists (command line)	826
Trading Partner objects	835
Trading Partner objects	836
Working with Trading Partner objects	837
Working with Trading Partner objects (command line)	851
Managing events on Gateway	890
About Event Management on Gateway	890
Decision Rule objects	890
Rule Table objects	890
Basic example	891
Types of triggering event and conditions	891
Types of triggered processes	895
Rule Table exits	895

Working with Rule Tables and Decision Rules	896
Creating a Rule Table	897
Creating a Decision Rule	898
Viewing a Rule Table or a Decision Rule	902
Modifying a Rule Table or a Decision Rule	903
Deleting a Rule Table or a Decision Rule	903
Copying a Rule Table or a Decision Rule	904
Refreshing the Decision Rule/Rule Table list display	904
Selecting one or more Decision Rules / Rule Tables from a displayed list	905
Creating an Axway Messaging connector	906
About Axway Messaging connectors	906
Creating an Axway Messaging IN connector	907
Creating an Axway Messaging OUT connector	908
Defining Decision Rule conditions	910
About Decision Rule conditions	911
Condition parameters	912
Operators	922
SMTP values in Decision Rule conditions	924
Scheduling Decision Rule condition parameters	925
Directory Scanning Decision Rule condition parameters	926
Completing the Value field	926
Symbolic variables for Directory Scanning Decision Rules	936
Using user functions	936
Case-insensitive conditions	942
New Rule Table: General tab	943
New Rule Table: General tab	945
New Rule Table: Linked rules tab	948
New Rule Table: Scheduling Event tab	949
New Rule Table: File Event tab	955
New Rule Table: Event tab	955
New Rule Table: XMS Event tab	956
Working with Decision Rules (command line)	957
Creating a Decision Rule: peladm create_decisionrule	957
Updating Decision Rules: peladm update_decisionrule	965
Deleting a Decision Rule: peladm delete_decisionrule	970
Displaying Decision Rule properties: peldsp display_decisionrule	971
Displaying all Decision Rule names: peldsp select_decisionrule	971
Working with Decision Rules (command line)	972
Working with Rule Tables (command line)	987
Decision Rules and Rule Tables: Parameters List	1013
Command syntax for Scheduling Event Models	1023
Executing scripts as command lines from Decision Rules	1029
Decision Rule usage example: Routing	1031
Model-based routing via the GUI	1031

Configuring Gateway for routing	1031
Results	1036
Decision Rule usage example: Routing	1037
Decision Rule usage example: Directory Scanning	1042
Decision Rule usage example: Scheduling	1050
Managing security	1056
Protocols	1057
About Axway Gateway protocols	1057
Supported protocols	1057
	1058
FTP	1058
About FTP	1058
FTP: Configuring the environment	1061
General FTP command attributes	1065
Using FTP in Initiator mode	1068
Using FTP in Responder mode	1076
Using an FTP proxy	1088
Monitoring an FTP Transfer Request	1091
Transferring multiple files with FTP	1091
FTP specific user exits	1095
HTTP	1099
About HTTP	1099
HTTP: Configuring the environment	1100
Using HTTP in server mode	1103
Using HTTP in client mode	1138
Using HTTP/HTTPS proxies	1150
Adding non-standard HTTP headers	1151
OFTP (Odette)	1153
About OFTP	1153
How OFTP processing works	1160
Configuring Gateway for OFTP	1165
OFTP identification code	1169
Submitting an OFTP Transfer Request	1170
Monitoring an OFTP transfer	1176
OFTP trace example	1179
Using PassPort PM with OFTP	1184
Transfer to an existing file name	1192
About PEL	1195
PEL	1195
How PEL processing works	1198
Configuring Gateway for PEL	1201
PEL request counters for LOTS transfers	1203

Submitting a PEL Transfer Request	1204
PeSIT Hors SIT	1210
About PeSIT Hors SIT	1210
How PeSIT Hors SIT processing works	1215
Configuring the PeSIT Hors SIT environment	1225
Submitting a PeSIT Hors SIT Transfer Request	1233
Monitoring a PeSIT Hors SIT transfer	1239
PeSIT specific user exits	1242
SFTP	1242
SFTP Introduction	1242
Using SFTP in Gateway	1243
Gateway-specific SFTP commands	1248
Configuring SFTP	1254
SFTP security	1257
Submitting an SFTP Transfer Request	1259
Transferring multiple files with SFTP	1259
SMTP/POP3	1262
SMTP/POP3 Introduction	1262
Using SMTP in Gateway	1263
Configuring the SMTP environment	1263
Configuring security for SMTP	1267
SMTP/POP3: sending an email	1267
Using POP3 in Axway Gateway	1272
Configuring the POP3 environment	1274
Processing email after retrieval	1278
Diagnostics and Troubleshooting SMTP/POP3	1279
EDIINT	1281
EDIINT Introduction	1282
General operating principles	1283
Generic EDIINT transfer example	1290
Configuring and using EDIINT	1291
RosettaNet	1298
About the RosettaNet protocol	1298
Gateway/RosettaNet message processing	1299
Configuring the RosettaNet environment	1301
Using RosettaNet	1306
Starting and stopping the server	1311
Server startup overview	1311
Before you start the Gateway server	1311
IPC Queues	1311
TCP/IP subsystems	1312
SNA LU 6.2 subsystems	1312
Starting and stopping the Gateway server	1312

Windows environments: Starting	1312
Windows environments: Stopping	1312
UNIX environments: Starting and stopping	1313
Starting and stopping the Navigator server	1313
Starting and stopping the Axway MFT Navigator server	1313
After you start the Gateway server	1314
Windows environments	1314
UNIX environments	1314
After you stop the Gateway server	1316
Checking the status of processes	1316
Starting and stopping the server	1317
Server startup overview	1317
Before you start the Gateway server	1318
Starting and stopping the Gateway server	1318
Starting and stopping the Navigator server	1319
Starting and stopping the Axway MFT Navigator server	1320
After you start the Gateway server	1320
After you stop the Gateway server	1322
Importing and exporting objects	1323
Overview	1323
Using the pelbase command	1323
pelbase subcommands	1324
Upgrading the Mailbox	1359
Upgrading the Mailbox: pelmig	1359
Virtual Sites	1361
Executing server traces	1362
pelctl trace_entity	1362
pelctl trace_process	1363
Managing rotation and archiving of trace files	1364
Modifying trace rotation parameters	1364
Modifying trace archiving parameters	1365
Triggering trace rotation	1366
About gatestatus messages	1366
GateController and gatestatus	1367
Possible gatestatus responses	1367
List of gatestatus messages	1367
Resetting the status prompted by the gatestatus command to the default value	1370
Reloading the profile	1370
Messages and codes	1371
Structure of log messages	1371
Message identification (column 2)	1371
Message text (column 3)	1373
Examples of message text	1373

Access Management (AM)	1373
EDI errors	1375
Exits (EXT)	1377
File Transfer Protocol (FTP)	1380
FTP reply codes	1407
FTP code explanations	1407
Second digit definitions	1408
Third digit definitions	1408
FTP reply code list	1408
Command-reply correspondence table	1414
Hypertext Transfer Protocol (HTTP)	1415
HTTP reply codes	1429
Introduction to HTTP code	1429
HTTP reply codes detail	1432
Axway Integrator messages	1442
JMS	1444
Job Control Table (JCT)	1448
Dynamic processes	1448
User processes	1451
LU 6.2 SNA network (SNAC)	1454
Axway Messaging connector routing (XMSC)	1458
MGET and MPUT messages (MXF)	1466
Monitor checking (CHK)	1477
Network module (NET and NDSP)	1480
OFTP protocol log messages (ODT)	1506
Payload integrity	1518
Diagnostic codes	1518
Errors	1519
PEL protocol (PEL)	1519
PeSIT HS protocol (FPSIT)	1537
Post Office Protocol (POP3)	1564
Process Monitoring messages (GC)	1576
Protocol Connection (PCNX)	1592
Reason codes and messages	1596
Routing (NOTIF)	1599
Routing Axway Messaging (RXMS)	1613
Rule Table evaluation	1618
SECS messages	1645
Simple Mail Transfer Protocol (SMTP)	1712
SSH File Transfer Protocol (SFTP)	1717
Statistics and script resolution modules (SR STATS)	1735
Supervisor module (SUP)	1738
SWIFTNet (SNET)	1809
TCP Server Daemon (TSD)	1878

Time Driver (TDRV)	1889
Trade messages (TRADE)	1890
Trading Partner Management (TPM)	1904
User Request (USR)	1909
VFD errors	1909
X.400 (X420) log messages	1930

Axway Gateway connectors 1943

Axway Messaging connector	1943
Axway Gateway and Axway Messaging	1943
Operational overview	1944
Axway Messaging connector	1945
Linking Axway Gateway to Axway Messaging: Overview	1946
Axway Messaging connector	1947
Linking Axway Gateway to Axway Messaging	1950
Linking Axway Gateway to Axway Messaging (command line)	1953
Axway Sentinel	1964
Configuring Gateway for Sentinel monitoring	1964
Sentinel: XFBTransfer Tracked Object	1966
XFBTransfer Sentinel/Gateway correspondence	1988
Sentinel: XFBLog Tracked Object	1995
Managing .pmq files with the pmqctl command	1996
Axway Integrator / AMTrix	1998
About Axway Integrator	1998
About AMTrix	1998
Axway Integrator connector	1999
Axway PassPort PM connector	2003
About PassPort PM	2003
Optimizing performance between Gateway and PassPort PM	2004
Axway PassPort PM connector	2004
Axway PassPort PS connector	2006
Axway PassPort AM connector	2007
SWIFTNet	2008
SWIFTNet Introduction	2008
SWIFTNet connector	2010
Backup remote sites for SWIFTNet	2037
SWIFTNet input channels	2075
SWIFTNet Header Info	2120
SWIFTNet Signature List	2137
SWIFTNet ASP check commands	2163
SWIFTNet HA overview	2171
JMS	2181
About Java Message Service (JMS)	2181
JMS connector	2184

Configuring the Axway Gateway/JMS environment	2187
Working with Axway Gateway/JMS	2189
X.400	2194
About X.400	2194
Addressing	2194
X.400 components	2195
X.400 protocols	2195
X.400 connector	2197
Configuring Gateway for X.400	2198
Managing X.400 partners	2200
Working with X.400	2205
TARGET2	2206
About TARGET2	2206
About QT2	2207
SWIFT documentation	2207
TARGET2	2207
Checking the TARGET2 package installation prerequisites	2220
Imported Decision Rules	2223
Imported Models	2223
Imported Trading Partners	2223
Imported Sites	2224
Remote Sites	2252
Trading Partner	2252
Models	2253
Decision Rules	2253
Remote Sites	2253
Trading Partner	2253
Models	2254
Decision Rules	2254
Pre-built test cases	2256
Customizing Gateway processes	2266
User exits	2266
About User exits	2266
Internal and External exits	2267
Exit families	2268
Exit-related files	2268
Exit management overview	2269
Internal user exits	2269
External user exits	2305
Compiling C user exits	2352
C API primitives	2353
Interface header	2353
List of API primitives	2353

C API primitives	2355
C API usage examples	2364
C API Transfer and Selection parameters	2370
Code conversion and file formatting	2382
Overview	2383
Defining code conversion and formatting parameters	2383
Code conversion	2384
Formatting and padding	2391
Examples	2394
Managing Audit files	2396
How to read the audit file	2396
Content of the audit file after export	2396
Actions not logged to the audit	2397
Actions logged into audit without user action	2397
Actions for Audit files	2398
Working with Audits (command line)	2398
pelbase export_audit	2398
pelctl archive_audit: Archive the audit	2399
pelauditexport: export audit file when Gateway is stopped	2400
	2402
Viewing and managing Audit	2402
Viewing Audits	2402
Viewing Archived Audits	2403
Filtering the Audit message display	2404
Refreshing the Audit display	2406
Exporting the Audit display	2407
Exporting the Archived Audit display	2407
Archiving audit files automatically	2407

What's new in this version

Gateway 6.17.3 SP10

What's new	Refer to
Option to set the selection order of the Key Exchange Algorithm : <i>client</i> or <i>server</i> preferred order.	<i>Security guide</i> > Managing SSH Security Profiles > Key exchange algorithms (requires login)
Connection Limiter option in Secure Relay , relating to Gateway.	<i>User guide</i> > Configuring Secure Relay advanced options

Gateway 6.17.3 SP9

What's new	Refer to
Gateway supports multiple PKI structures, as well as easy certificate renewal across all levels of the certification path.	<i>Security guide</i> > Managing TLS (requires login)
Local certificates on negotiated cipher: only if compatible	<i>Security guide</i> > Managing TLS > Certification path building mode

Gateway 6.17.3 SP5

What's new	Refer to
Gateway now offers a mechanism to protect against Cross-Site Request Forgery (CSRF) attacks	Using HTTP in server mode on page 1103
For Virtual File Directories, you can set a limit to the number of concurrent listings for each protocol	User guide > VFD concurrent listing on page 382
Gateway's response for a non existing file can be switched from an empty listing to 550 (<i>Request action not taken</i>).	To configure this behavior you can set <code>peluconf [ft_ftp] alter_list_response</code> parameter to 1

Gateway 6.17.3 SP4

What's new	Refer to
SFTPV6 protocol can now be disabled through configuration at remote site level.	Axway Gateway: Managing Partners on page 677 Axway Gateway: Managing Partners on page 710
HTTP authentication protocols NTLM and NTLMv2 are now supported for Gateway Client.	Using HTTP in client mode on page 1138

Gateway 6.17.3 SP3

What's new	Refer to
X-Priority SMIME headers now have alphanumeric values starting with the 1902 SEPAMAIL standard. Numeric values are encoded as 1 = HIGHEST, 2 = HIGH, 3 = NORMAL, 4 = LOW, 5 = LOWEST.	- Managing partners > ... > Axway Gateway: Managing Partners on page 677 - Managing partners > ... > Axway Gateway: Managing Partners on page 710
-	

Gateway 6.17.3 SP2

What's new	Refer to
Improved cache for Passport: Gateway can be configured to cache user authentication and privileges when PassPort AM is configured as access manager.	Configuring connectors on page 221
You can set a policy to archive audit files automatically	Managing audits > Viewing and managing Audit on page 2402
New option to avoid restart on broken statuses caused by vital processes	Messages and codes > Process Monitoring messages (GC) on page 1576

What's new	Refer to
Gateway now chooses local certificate only if its key is compatible with the key exchange algorithm of the negotiated cipher. You can thus configure a security profile with multiple certificates, to use connections with different requirement levels: old software that only supports RSA ciphers, as well as new software supporting elliptic curve cryptography.	<i>Security guide > Managing TLS > Certification path building mode</i>
Command line option to disable the use of MD5 hash algorithms	<i>Security guide > TLS Cypher suites Security guide > Using SSH</i>
Delete only temporary files when purging, option to keep mailbox record	<i>Managing transfers > ... > Axway Gateway: Managing Transfers on page 506 Managing transfers > ... > Viewing and managing Mailbox contents on page 518</i>
To preserve security, the Browse button has been removed from Navigator.	<i>All references to the Browse button removed in documentation.</i>
Gateway supports SFTP versions 0, 1, 2, 3 and 6	<i>Protocols > SFTP protocol</i>

Gateway 6.17.3 SP1

What's new	Refer to
Documentation To protect the security of your data, documentation content previously contained in the <i>User guide</i> under the <i>Managing security</i> chapter has been moved to the <i>Security guide</i> > <i>Security administration</i>. The reason is that access to the Security guide is restricted on the internet to users with Axway IDs, which excludes malevolent hackers looking for insights into any sensitive information.	<i>Security guide</i> > Security administration (requires a login)
The connection to Integrator/XIB can be secured using TLS	Configuration > Configuring connectors

Gateway 6.17.3

What's new	Refer to
TLS extensions: When connecting as a client, Gateway uses the host name of the remote partner as the value for the Server Name Indication extension, as specified by RFC6606 from the IETF.	See restrictions: Security guide > SSL and TLS protocols > TLS Extensions
Native support for 64-bit architecture on AIX and Windows	Installation guide > Installation prerequisites
SWIFT: Compliance with SWIFT CSP	
Multiple backup sites for SwiftNet remote sites. Gateway now allows up to 4 backup sites for each SWIFTNet remote site.	Site objects: Parameters List

What's new	Refer to
Longer keys: maximum length of certificates with RSA keys imported in Gateway is raised to 8192 bits (formerly 4096). PEM and DER formats are supported. Only RSA-based signature algorithms are supported. Two hash algorithms are available and can be associated with RSA: MD5 and SHA-1.	Security guide > Certificate Restrictions Security guide > Managing TLS
Updated OpenSSL: Cryptographic operations now rely on OpenSSL version 1.0.2k	
JMS logging now supported by Apache Log4j 2	
No longer supported in this version: • ETEBAC 3 Client & Bank and ETEBAC 5 Client & Bank • Infozip and infounzip tools	

Gateway 6.17.2 SP2

What's new	Refer to
Cipher suites configuration It is now possible to configure - both in GUI and CLI - up to 32 TLS ciphers in TLS security profiles (server and client).	Security guide > Managing TLS Security Profiles
Audit support Gateway stores the actions performed by users on the Gateway configuration (global and objects configuration) on a dedicated file, providing visibility of the changes done on the product. This functionality cannot be disabled.	Managing Audit files on page 2396

What's new	Refer to
Payload integrity You can enable the payload integrity check to assure that payload haven't changed between the moment it has been received by Gateway and further actions: routing to Integrator or routing to a third party. The protocols for which the signature is computed in Gateway, for incoming transfers are: SWIFTNet, PeSIT and JMS.	Security guide > Payload Security

Gateway 6.17.2

What's new	Refer to
Encryption of password information in export files	Security guide > Managing SSH Security Profiles (command line) Security guide > Managing TLS Security Profiles (command line) User guide > Importing and exporting objects on page 1323
ECDSA support with PassPort and XSR Termination ECDSA keys and certificates can now be used with PassPort and XSR Termination. Please make sure you use Passport 4.6 SP12 P2 and SecureRelayRA 2.7.0.	Security guide > TLS cipher suites
Improved cache for TLS sessions An improvement in TLS cache now allows the use of cache between control and data sessions on FTP.	
Certificate chain size on TLS The limit of chain size sent on TLS was increased from 16k to 64k.	

What's new	Refer to
Native support for AIX-64 bit operating systems With this release, Gateway includes native 64-bit support for AIX Operating System. Besides the new installation package, a cross-platform migration tool is available to handle the Gateway 6.17.2 AIX 32-bit to Gateway 6.17.2 AIX 64-bit migration.	Upgrade guide > Upgrading to 6.17
New signature for transfer user exits ExitXfer* (perl and C) To support passing custom user data in subsequent ssh exit calls and to be able to differentiate between ssh and tls context, the transfer exits <code>ExitXfer*</code> (perl and C) have a new signature.	Signatures: refer to User guide > Customizing Gateway processes > User exits > External user exits on page 2305 Upgrade: Upgrade guide > Upgrading to 6.17
Length of file label PeSIT transfer parameter The limit for transfer parameter <code>file label</code> has increased from 80 chars to 256 chars.	Refer to User guide > Managing transfers > Submitting transfers > Transfer requests > Axway Gateway: Managing Transfers on page 283

Gateway 6.17.1

What's new	Refer to
Upgrade to Perl 5.24 Perl version 5.24.0 is required to compile the Perl exits or to run the Perl scripts.	Refer to User guide > User interfaces > Online commands > Online commands in Perl on page 147
Password authorizing operations on Gateway is secured The password used for authorizing operations on Gateway through commands (command line utility, exists, scripts) when access management is enabled is now stored in encrypted form.	Security guide > Password management

Gateway 6.16.1

What's new	Refer to
ECDSA keys support for SSH The ECDSA keys supported are the ones required by RFC5656 (nistp256, nistp384 and nistp521). The key exchange algorithm list has been updated with new entries specific to this key format.	Security guide > SSH user exits
ECDSA certificates support for TLS You can now import in Gateway ECDSA certificates. The supported curve format for ECDSA certificates are the ones specified in rfc4492.	Security guide > Managing Keys and Certificates in SSH
New TLS ciphers support Next cipher suites are now available: DHE - ECDHE-RSA - ECDHE-ECDSA - AES-GCM	Security guide > TLS cipher suites
Weak ciphers disabled On the new Gateway installations the weak ciphers are not present on TLS profiles. User will still have the possibility to enable them using a configuration option, [tls] enable_weak_ciphers set on 'yes', but this is strongly discouraged.	Security guide > TLS cipher suites
Secured the connection between Gateway and Sentinel using TLS Gateway can now use TLS when connecting to Sentinel. TLS connection is activated by default on new installations. You must make sure that Sentinel exposes a TLS-secured port, in order to have a successful connection between the two products.	Configuring connectors on page 221
Upgrade to OpenSSL version 1.0.2 For the cryptographic operations the OpenSSL version 1.0.2 is being used.	

What's new	Refer to
Manually force Gateway to re-read PassPort PM Communication Protocol Profiles (DEA) <code>pelctl</code> was extended with a new subcommand to invalidate all cache used for the communication with Passport PM, or to invalidate the cache per DEA (<code>pelctl invalidate_tpm_cache</code>). Changes should reflect immediately on incoming/outgoing exchanges.	PassPort PM cache on page 2005

Gateway 6.16.0

What's new	Refer to
Managing TLS Security Profiles New commands to create and modify security profiles using the command line UI.	- Security guide > Managing TLS Security Profiles - Security guide > Managing TLS Security Profiles (command line)
Support for TLS 1.1 and 1.2 Configure and use TLS1.1 and TLS1.2 with Gateway	Security guide > TLS cipher suites
Discontinued support for CSR sample certificates	Configuration: connectivity parameters on page 176

What's new	Refer to
Secure Relay IP filtering Configuring IP filtering on Secure Relay Master Agent, from Gateway, using black lists and white lists	• Configuration: Secure Relay advanced options • Configuring protocols: SAP parameters on page 194
PassPort configuration Configuring Gateway's connector to PassPort for services: PKI, access management, partner management	• Axway PassPort PM connector on page 2004 • Axway PassPort AM connector on page 2007 • Axway PassPort PS connector on page 2006 • Security guide > Working with PassPort AM
Support for FIPS Configuring Gateway correctly to establish FIPS-compliant secure connections	• Security guide > About FIPS

Axway Gateway: Overview

Overview

Definition of Axway Gateway

Axway Gateway (formerly XFB Gateway) is an Axway product in the Managed File Transfer category, that serves as a communication gateway. It provides a secure front end for handling data exchange between partners that are connected via public networks such as the Internet. It supports a wide range of open standard protocols that includes, but is not restricted to:

- FTP(s)
- JMS
- HTTP(s)
- Ssh-FTP
- PeSIT
- EDIINT AS1, AS2 and AS3
- OFTP (Odette)
- SWIFTNet

Principal functions

Gateway enables you to simplify and automate your file transfer processes. You can use Gateway for the following functions:

- **File server function**

Gateway centralizes file flow between generic workstations and Axway Transfer production.

- **Gateway router function**

Gateway provides general protocol and network gateway access, typically between corporate networks and Internet and extranet networks. It isolates the local network from the Internet to guarantee data integrity and confidentiality.

- **Axway Transfer function**

You can use Gateway to create, submit, receive and control file Transfer Requests.

You can integrate Gateway into your security solution. Gateway provides the following standardized security functions:

- Firewalls and proxies
- DMZ deployments (using Axway Secure Relay or, alternatively, using paired DMZ Gateways)
- Partner identification (internal database or external database via LDAP protocol)

- SSL/TLS one-way or mutual certification
- Optional encryption of temporary files stored on the gateway

[Next topic](#)

Axway Gateway: Overview

Network Architecture

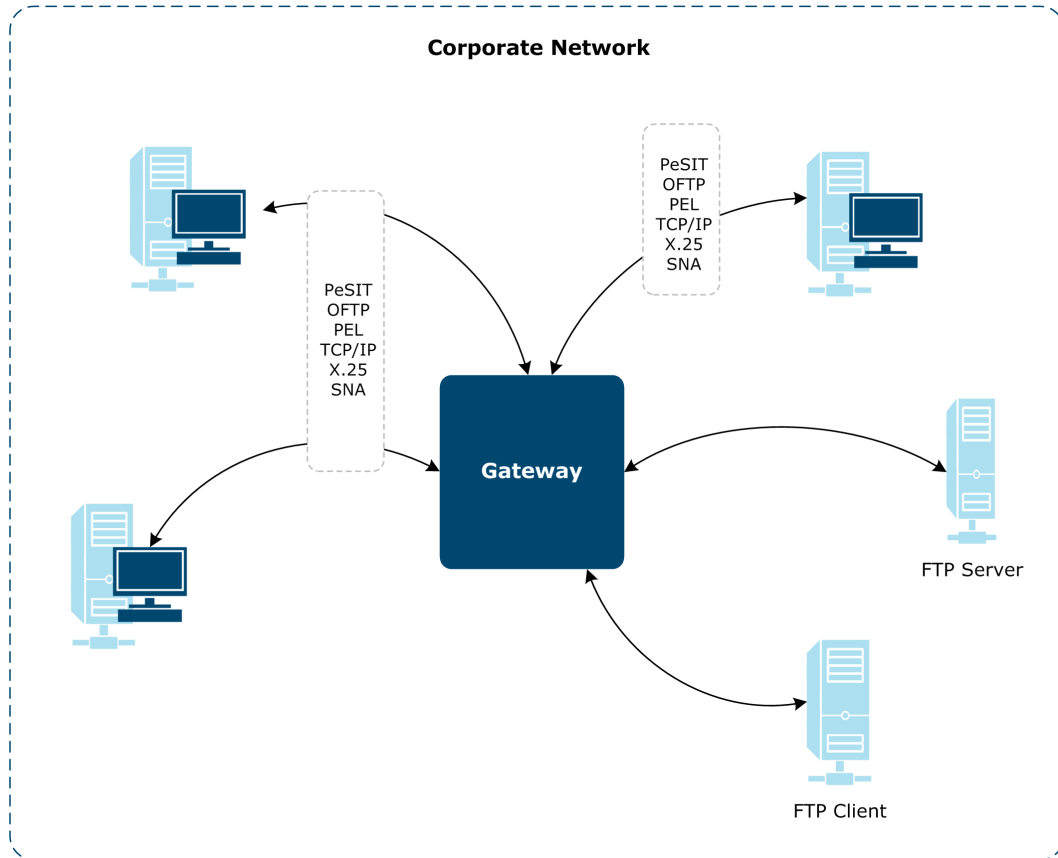
As described in the previous topic, Gateway enables you to transfer different types of file between different applications, both within a local network and over the Internet.

You can integrate Gateway into a large variety of network architectures. The architecture you select depends on the Gateway functions you want to use.

Local Area Networks

Inter-platform file transfers

You can deploy Gateway in your local corporate network, to make use of its capacity to transfer variously-formatted files between different platforms using a variety of different protocols.



Sentinel monitoring

You can enhance your ability to track the activities of your Axway network applications by adding Sentinel monitoring.

Linking your local network to the Internet

Axway Secure Relay option

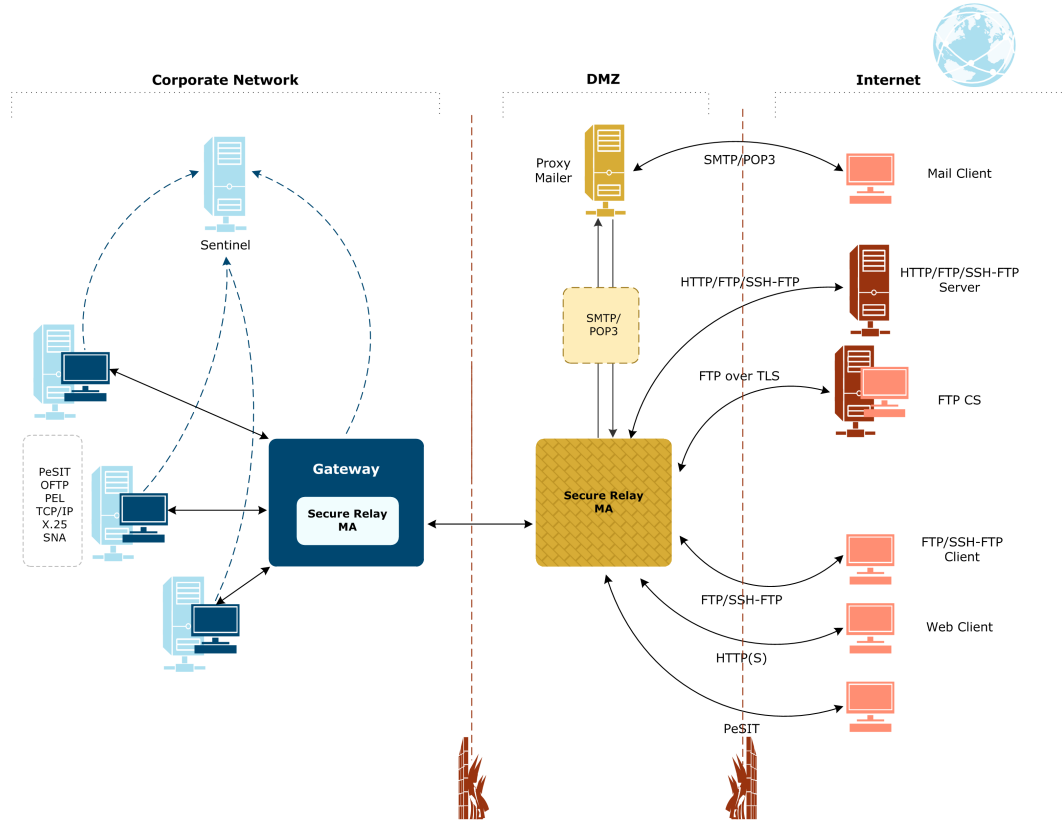
You can use your Gateway as an Internet gateway. The Gateway product includes Secure Relay components that enable you to deploy Gateway in a secured DMZ configuration.

Secure operation is assured by:

- A Secure Relay Master Agent that you install and run natively on the Gateway
- One or more Secure Relay Router Agents, delivered as part of the Gateway package, that you install in the DMZ

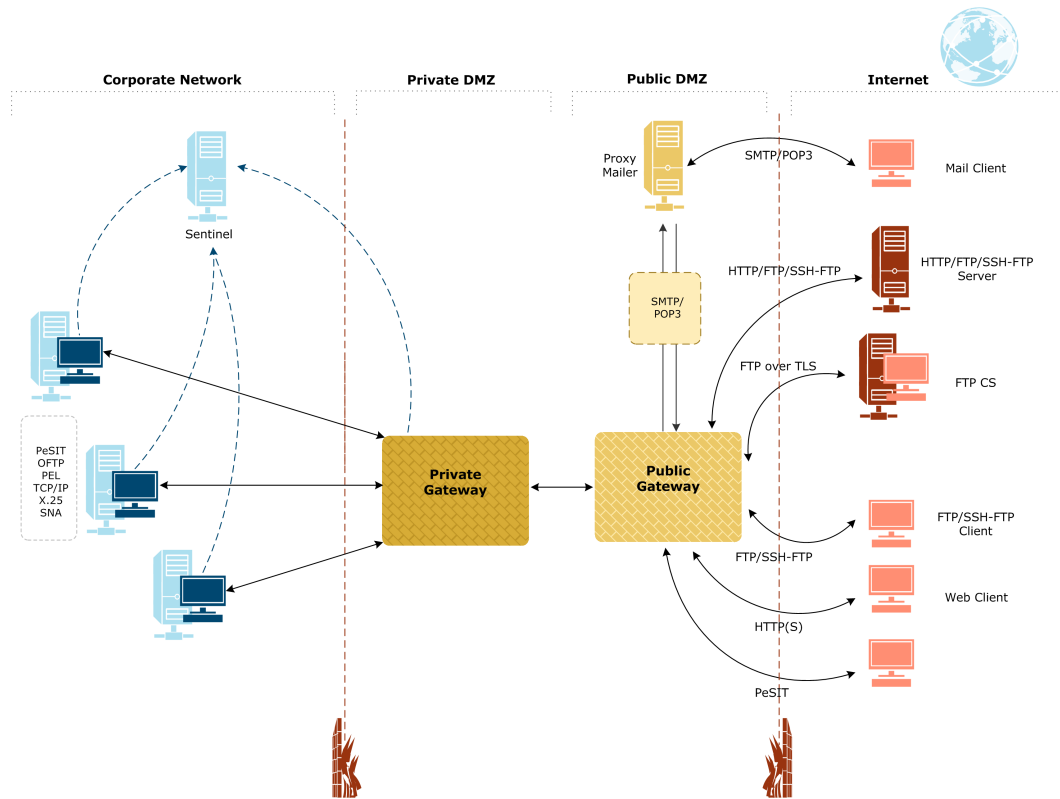
Together, these two components:

- Manage connections to the outside world for file exchanges, preventing external partners from directly accessing Gateway
- Handle streaming of data flows, to and from the Internet



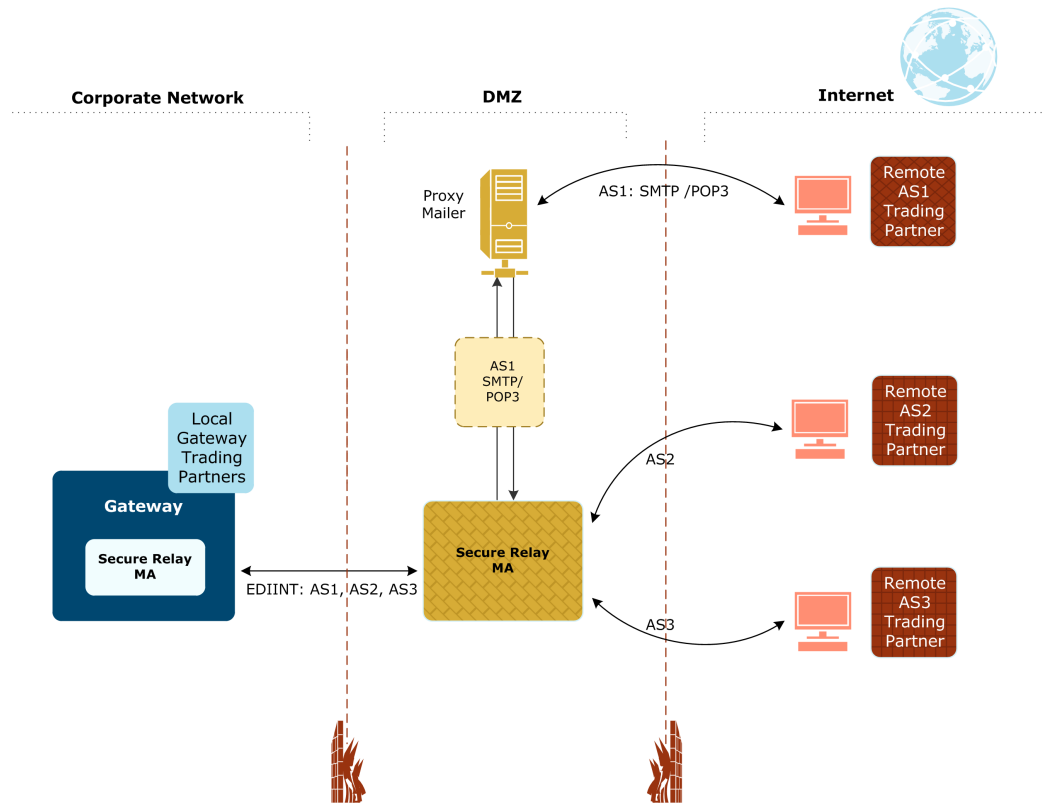
Gateway-to-Gateway DMZ option

Alternatively, you can deploy two Gateways in a bridged private/public DMZ configuration.



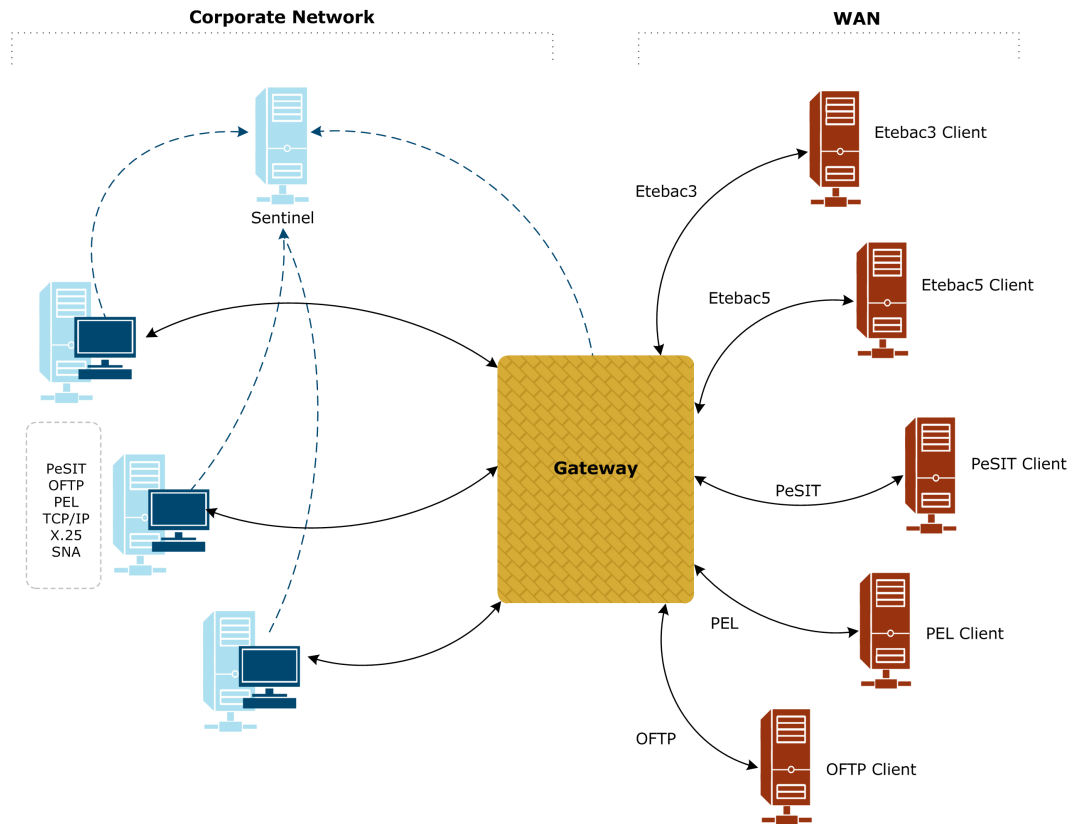
Using EDIINT protocols

You can deploy Gateway as an EDIINT file transfer tool, creating links to your business trading partners via the Internet.



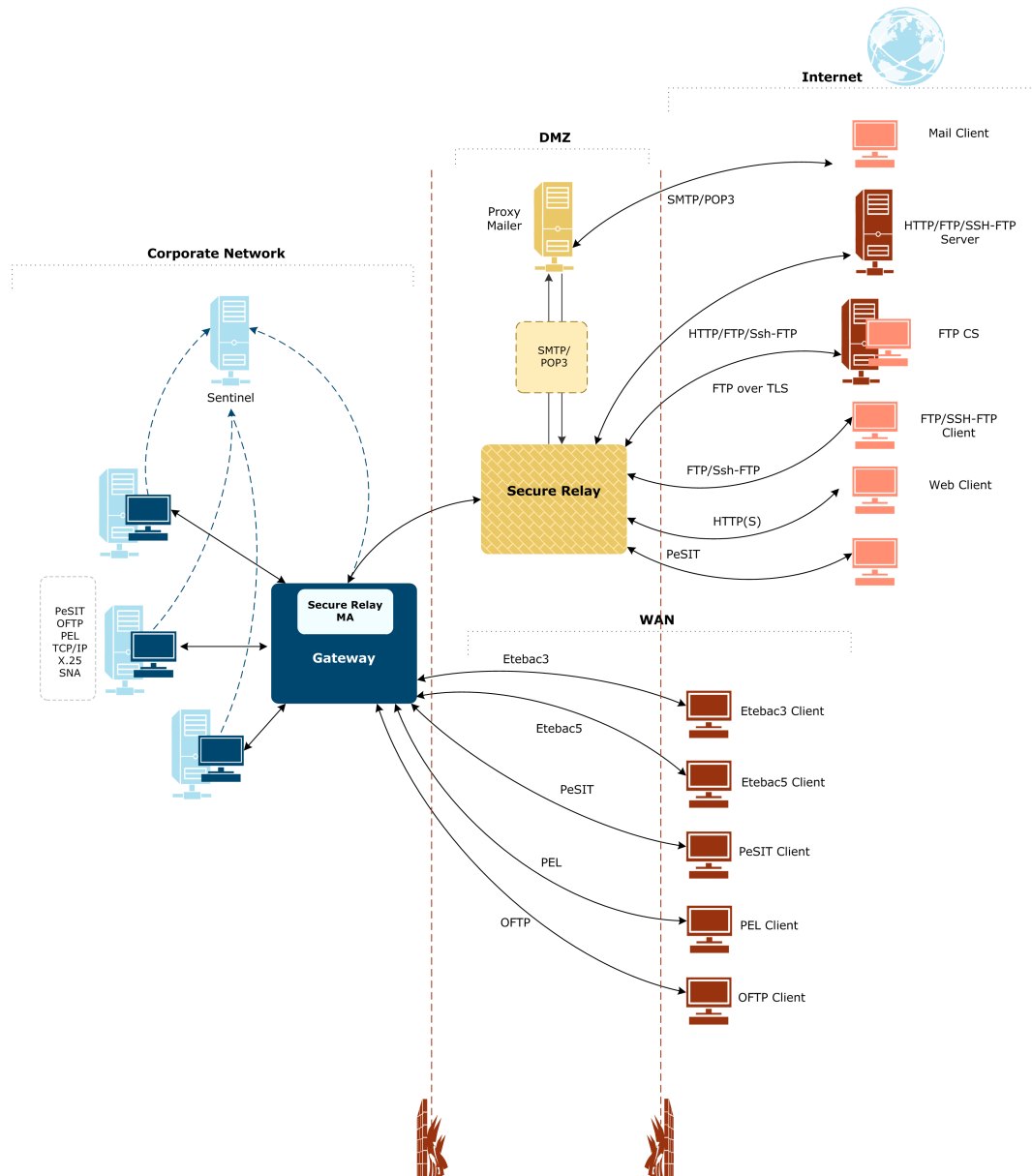
Linking your local network to a WAN

You can use Gateway to link a local network to an external corporate network.



Interconnecting applications over multiple local and remote networks

You can make use of the multiple protocol handling capacities of Gateway to deploy the product for exchanges over multiple networks.



Axway Gateway: Overview

File Transfers and Transfer Requests

As described in the [previous topic](#), Gateway facilitates the exchange of files between diverse platforms and applications, over local networks and across the Internet.

File Transfers

Most of the tasks you execute with Gateway involve configuring and executing *file transfers*. File transfer is the process of displacing a file from one computer platform to another.

In any file exchange, one computer is the sender, another is the receiver. The two computers are linked together by a network, either TCP/IP or SNA. An inter-platform file transfer also requires the use of a high-level file transfer protocol, independent of the supporting network type and the hardware and software architecture of the host computers.

Supported protocols

Gateway enables automatic and secured file transfers over the following supporting protocols:

- FTP
- HTTP
- OFTP (Odette)
- PEL
- PeSIT HS D
- PeSIT HS E
- SFTP (SSH-FTP)
- SMTP
- POP3
- EDIINT
- RosettaNet
- SWIFTNet
- JMS
- X.400

Transfer Requests

A file transfer consists of sending:

- The file itself
- Technical parameters associated with the file. For example: connected partners, local file storage format, and so on.

Transfer Request object

The *Transfer Request object* contains the complete set of information required to trigger a Gateway file transfer:

- Name of file to transfer
- Physical file attributes
- Protocol used for the transfer
- Compression or encryption options
- Behavior in case of transfer failure
- Transfer scheduling, restarting and routing

Mailbox

Gateway stores all data related to each Transfer Request in a permanent file, named the *Mailbox*. During the transfer process, Gateway updates the Transfer Request status and information in the Mailbox. This information allows transfer restart upon failure or transfer completion.

Transfer acknowledgments

Depending on the protocol used for a file transfer, Gateway can handle both positive and negative acknowledgments.

A *positive acknowledgment* informs the sender that the recipient correctly received or processed the file. The following protocols support positive acknowledgments:

- PeSIT HS E
- OFTP (Odette)
- SMTP
- POP3
- FTP (via a proprietary mechanism, only when communicating with another Gateway)

A *negative acknowledgment* indicates that errors occurred during the processing of a received file, or that file transfer failed during routing. The following protocols support negative acknowledgments:

- PeSIT HS E (via the proprietary `nack_option` parameter)
- SMTP
- POP3

Transfer-related objects

The **Transfer Request** object and the [Site](#) object together usually contain all data required to process transfers. Additionally, some transfers require an [Application](#) object or a [Trading Partner](#) object.

[Next topic](#)

[Managing Transfers](#)

[About Axway Gateway protocols](#)

Transfer Sites

As described in the [previous topic](#), Gateway operations generally involve the transfer of files between two or more computers. In order to process a Transfer, Gateway needs information about the originating and destination computers. In Gateway terminology, originating and destination computers are known as *Sites*. You describe the characteristics of sites to Gateway by creating *Site objects*.

Each Transfer Request requires the definition of two Site objects:

- **Local Site object** - provides the characteristics of the computer that hosts your Gateway. In many cases, Gateway automatically collects the necessary information about your local Site from information you provide during installation and configuration.
- **Remote Site object** - provides the characteristics of the computer with which the local computer is exchanging the file or files. In most cases, you specify the remote site that is the partner in a transfer, and specify the properties of the site in the Remote Site object.

Site characteristics

Site alias

You attribute a unique name to every Site object you create. This unique name is known as the *Site Alias*.

Site role

In a given Transfer, one of the two sites is the Sender site and the other is the Receiver site. Additionally, one of the two sites is the Initiator (client) and the other is the Responder (server).

Therefore, for a given transfer, a Remote Site can fill one of four possible roles:

- **Initiator/Sender:** The Remote Site acts as a client and initiates a transfer towards the local Gateway.
- **Initiator/Receiver:** The Remote Site acts as a client, requesting the transfer from the local Gateway.
- **Responder/Sender:** The Remote Site acts as a server responding to a request for a transfer from the local Gateway.

- **Responder/Receiver:** The Remote Site acts as a server, accepting a transfer from the local Gateway.

Site protocol

In a given Transfer, the sending and receiving sites have to agree on a protocol to use for the Transfer.

Remote Site

You specify the protocol you want to use for a Transfer in the Remote Site object. The Remote Site must support the protocol you specify, or the transfer attempt will fail.

Local Site

You must also configure your Local Site to handle the specified protocol. Typically this means that you must activate the protocol in the configuration menu and, if your local machine is acting as a server, in a CGate (Connection Gate) object. You do not need to specify the protocol to use in the Local Site object.

Additional Site object characteristics

In addition to the basic information described above, you can specify other characteristics of Site objects. These parameters can improve the performance and security of your transfers. The information you can enter varies according to the type of transfer you choose to execute.

Dynamic Sites

To exchange files in client mode, Gateway requires a complete description (including network parameters such as IP address and SAP) of the connection partner and the file exchange process.

However, when multiple client stations exist, the task of redefining a new Remote Site for each and every file exchange partner may become long and time-consuming. Gateway provides the *dynamic site creation* feature that enables you to accelerate the site definition process and exchange files with unrecognized client stations.

When you submit a Transfer Request for which no defined Remote Site exists, Gateway dynamically creates a site that is based on one of the *Template Sites* provided with the product. A Template Site is like any Remote Site, but some of its protocol and network attributes remain empty.

When Gateway dynamically creates a Site, the lifetime of the Site continues as long as there is a reference to the Site in an active Request or in one of the records contained in the Mailbox (repository of transfer records). If no Requests or Request records are associated with this Site at the end of the session, Gateway automatically deletes the dynamic Site.

Gateway automatically creates several Template Sites when you install the product. For example:

- TFTP (for generic access from FTP clients)
- THTTP (for generic access from HTTP clients)
- TFTP_HTTP (for generic access from clients using either HTTP or FTP protocol)
- TODT (for generic access from OFTP clients)
- TAS2 (for generic access from AS2 clients)

Additionally, Gateway provides a set of Template Sites for generic access to servers (for example, when using PassPort PM):

- AS2
- AS3
- ETB3
- ETB5
- FTP
- HTTP
- ODT
- PEL
- PHSD
- PHSE
- SFTP
- RosettaNet

At the protocol connection level, Gateway analyzes the CGateGroup definition and checks whether a Template Site is specified. If the connection requires the use of a Template Site, Gateway:

- Dynamically creates a Remote Site based on the relevant Template definition
- Uses the received connection parameters to complete the Site definition

[Next topic](#)

[About Site objects](#)

Axway Gateway: Overview

Applications

As described in the [previous topic](#), a file transfer occurs between two or more computers. Gateway recognizes the computers involved in a transfer via information stored in Site objects. To successfully transfer a file, Gateway also requires information describing the format of the file that is the object of the transfer.

Application object

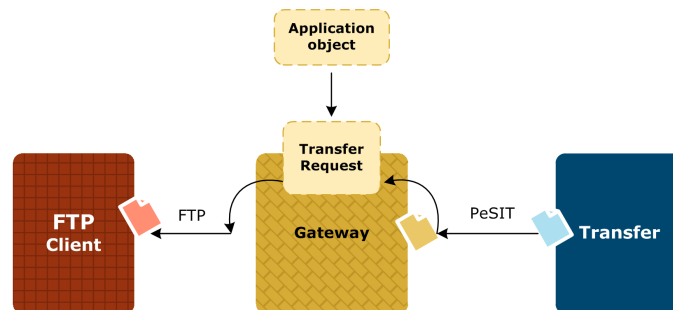
For some transfers, the applications that process the transferred file on the sending and receiving machines cannot handle files of the same format. If this is the case, you can create an *Application object* in which you specify the differences between the attributes of the file as it resides on your local machine, and the file that is transferred. Gateway then uses this information to transform the format of the transferred file.

You can enter the following types of information in an Application object:

- Local file system attributes such as name, file structure and type
- Record format
- Data encryption and padding
- Online record attributes
- Compression mode

Example

Gateway receives a file via the PeSIT protocol from an Axway Transfer product hosted on an MVS platform. An FTP client on a Windows platform then requests the transfer of the file from Gateway.



The file that Axway Transfer deposits to Gateway has the following characteristics:

- Record format: stream
- Record length: 180
- Data code: EBCDIC

The FTP client handles files with the following characteristics:

- Record format: Variable text
- Record length: 1024
- Data code: ASCII

You create an Application object that specifies the characteristics of the file as it is deposited to Gateway, and as it must be received by the FTP client. In the Transfer Request for the delivery to the FTP client you specify this Application object.

During the transfer process, Gateway transforms the file to the format required by the FTP client.

Specifying file characteristics for Transfers

When you have created an Application object for a specific type of transfer, you can then link the Application to the Transfer when you create the Transfer Request. The Application that you link must belong to the same group as the Site that you specify in the **Destination alias** field of the Transfer Request. By default, when you create new Sites and Applications, the group value is set to GDEFAULT.

It is not necessary to link an Application to every Transfer Request. There are two alternative ways that you can provide Gateway with application file system information for a Transfer:

- **Explicitly enter file information in the Transfer Request**

You can enter information about file format directly in the Transfer Request in the **Attributes** tab. Then, if you do not specify an Application in the Transfer Request, Gateway takes the **Attributes** tab values. In this case, you must specify the Record format and the Record length.

- **Enter the file format information in a Model**

You can enter values for file format attributes in a Model object. You then attach the Model to the Transfer Request. Gateway then takes the values from the Model object.

If you set the same parameters in both the Transfer Request and the Model objects, the Transfer Request parameters override the parameters set in the Model.

Default Applications

Gateway provides a set of default Applications. When Gateway operates in Responder mode, it assigns an Application to the incoming transfer as follows:

- **PeSIT transfers:** The Application name is the incoming protocol filename. The default Application names are DEFAULT_B (Binary), DEFAULT_A (ASCII) and DEFAULT_E (EBCDIC), depending on the type of data to be transferred. If Gateway cannot find the required Application (from those listed above), it uses the Application DEFAULT.
- **FTP/SFTP transfers:** The client station can define Application names with an FTP SITE command before sending the STOR or the RETR command. The default Application names are FTP_B (Binary), FTP_A (ASCII) and FTP_E (EBCDIC) depending on the FTP data type. Since ASCII and EBCDIC are meaningless in SFTP, SFTP transfers use the default Application FTP_B.
- **HTTP transfers:** Since the client station does not provide the Application name, Gateway uses the default FTP Application for file reception.

[Next topic](#)

Axway Gateway: Overview

Connection Gates

For security reasons, when a remote machine makes an attempt to connect to your local machine, you need to control whether or not to allow the connection. Gateway uses a combination of two different objects to filter connection attempts at the protocol level:

- Connection Gate Groups (CGateGroups)
- Connection Gates (CGates)

For a given connection attempt, Gateway scans the properties listed in CGateGroups and CGates to determine whether or not the remote partner is allowed to connect.

CGates and CGateGroups are only useful when Gateway is operating in *server mode*.

CGateGroups

A CGateGroup can contain other CGateGroups or CGates. The CGates contained in a CGateGroup share the properties of that CGateGroup.

When you create a CGateGroup, you give the group a name and specify the protocol or protocols that connecting machines can use to connect to Gateway. You must also enter a value for the called SAP (either a specific value or the universal variable '*').

Additionally, you may require other filtering criteria at the group level, such as an HTTP hostname. You may choose to specify directory access rights for HTTP and/or FTP connections for the CGateGroup.

CGates

Use CGates to:

- Activate and deactivate individual protocol CGates
- Associate IDs and passwords with protocol connection procedures - you can allow users to modify their own passwords
- Link TLS security functions to the connection
- Create remotely-accessible file directories and manage directory access rights for connected users (HTTP and/or FTP, SFTP)
- Specify a home page (HTTP only)
- Set upload limits (FTP / SFTP only)
- Exclude connections from specified calling addresses (TCP/IP supported protocols only)

SGates and SGateGroups

Gateway now supports the use of *Super CGates* (SGates) that offer a higher capacity than the standard CGates.

[Next topic](#)

[About CGates and CGateGroups](#)

[About SGates](#)

Axway Gateway: Overview

Diffusion Lists

A file transfer involves the exchange of a single file from an originating computer to one or more destination computers (Sites). Gateway enables you to create lists of computers that you want to use as destinations for a given type of file. When you create a Transfer Request, you can then specify the list as the destination. Gateway attempts to deliver the file to each computer on the list.

Diffusion List object

The *Diffusion List object* enables you to create a list of destination Sites.

A Diffusion List object comprises:

- One or more Sites and/or one or more Diffusion Lists, **or**
- A protocol-level destination, if the Diffusion List is associated with a Template Site (only useful if you are using Gateway in Responder mode).

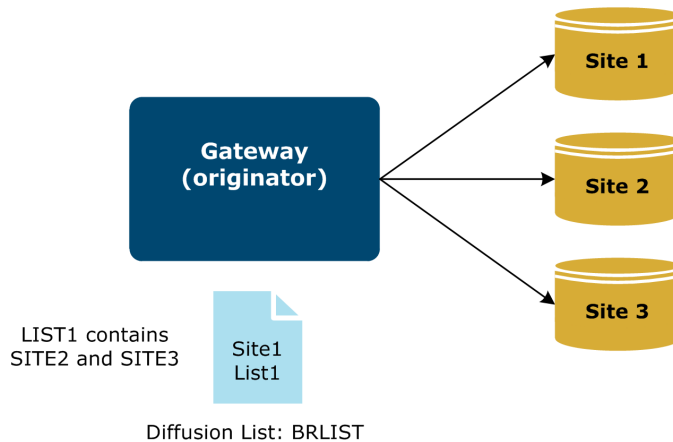
When you submit a Transfer Request with a Diffusion List name as the destination alias, Gateway:

- Creates a parent Request (Diffusion List Request)
- Creates one Transfer Request for each Site referenced in the Diffusion List

Example

Create a Diffusion List BRLIST and use the list as the destination alias in a Transfer Request.

Gateway transfers the file to each Site of BRLIST (SITE1) as well as to each Site contained in each list of BRLIST (SITE2 and SITE3).



[Next topic](#)

[About Diffusion Lists](#)

Axway Gateway: Overview

Virtual File Directories

Gateway can act as a file server for FTP, SFTP or HTTP transfers. As a server, Gateway securely stores the physical files it receives, away from direct viewing by users. Gateway administrators can control the views that users have of files from client machines. For individual users, you can create virtual views of the system of stored files, while keeping the real contents inaccessible.

This user-adapted view of the file system is known as a *Virtual File Directory (VFD)*. A VFD appears in the GUI to users with the appropriate rights. Users to whom you have granted VFD rights can access only the set of directories that you organize for their use. Users are unaware of any directories and files that are not expressly intended for their viewing.

Gateway creates the file contents of VFDs from the contents of Transfer Request records contained in the Mailbox. Unless you select a mounting option, the virtual files represented in the VFD do not link to real files stored on the Gateway host machine.

You can set restrictions to user actions within the directories. You can specify:

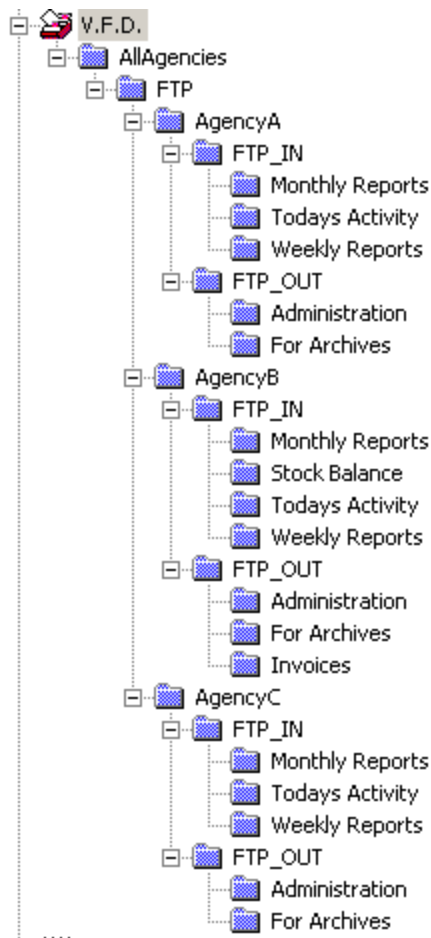
- The type of file to which users have access
- The type of operations users can perform on VFD directories:
 - List
 - Create
 - Remove

- The type of operations users can perform on VFD files:
 - Read
 - Delete
 - Write

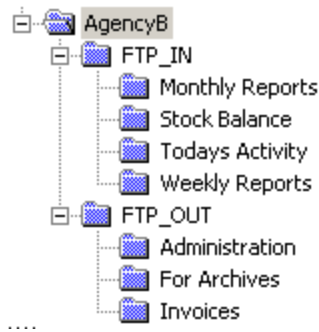
Example

You can create a VFD to handle the file deposits and requests of FTP clients connecting from Agency A, Agency B and Agency C. You can create an individual VFD for each agency so each agency has a view only of the file directories with which it is concerned.

Your administrator view of the complete VFD directory structure might appear as follows:



You can manage the FTP access to the VFD for each Agency. For example, you might restrict the Agency B access so that when the agency connects the Gateway with an FTP client, the agency view of the directory structure appears as follows.



[Next topic](#)

[About Virtual File Directories](#)

Axway Gateway: Overview

Trading Partners

As described in previous topics, you can use Gateway to transfer files between two or more machines using many different protocols. The processes that Gateway uses to execute a transfer vary according to the protocol that you specify for the transfer. Gateway uses the information contained in different objects to carry out these processes.

Gateway includes a special set of services for the transfer of files via EDIINT (Electronic Data Interchange over the INternet), S/MIME, RosettaNet, OFTP R2.0 and SWIFTNet.

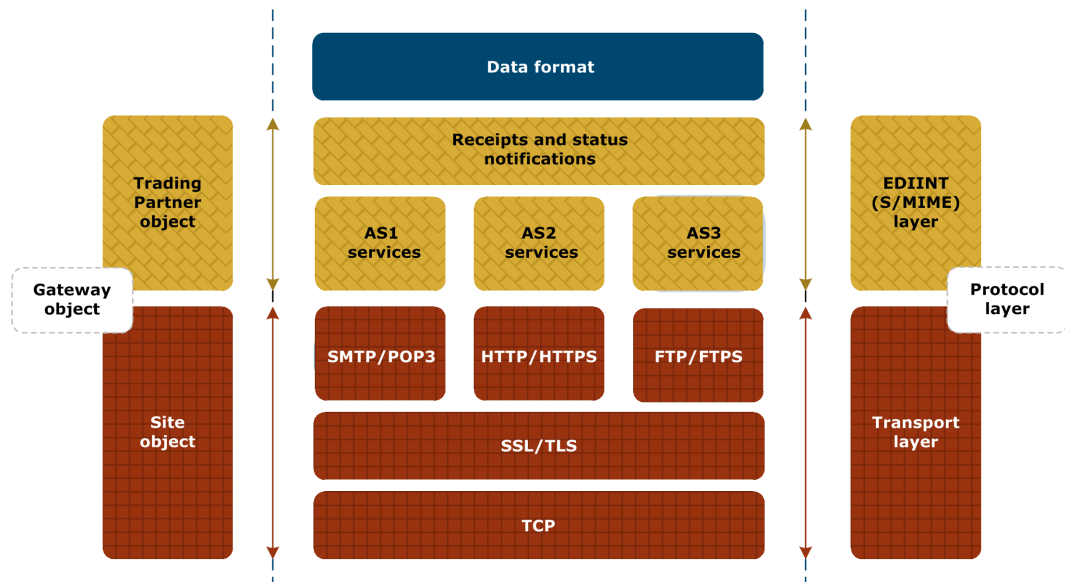
To transfer a file using EDIINT, the Gateway that originates the transfer performs a set of operations defined by the transport level protocol, and additionally performs compression, signature, encryption and encapsulation operations defined by the S/MIME protocol.

The Gateway that receives the transfer performs the reverse operations to extract the information contained in the body of the message. If necessary, the receiving Gateway then sends a reception acknowledgment known as a *receipt* to the originating Gateway.

To provide Gateway with the information necessary for the transport layer services that support higher level EDIINT and S/MIME services you create Site objects. These objects are identical to the Site objects you might create for standard SMTP/POP3, HTTP or FTP transfers.

Additionally, to provide the information to Gateway necessary for compression, signature, encryption and encapsulation operations you create a *Trading Partner object*.

The following illustration shows the relationship between the services provided at different protocol layers, and shows the Gateway objects that provides information for those services.



To execute an EDIINT or S/MIME transfer between two Gateways you require Trading Partner objects for both the originating and receiving machines.

For each Trading Partner object you create, you can specify the Gateway behavior on sending or receiving an EDIINT or S/MIME transfer.

For locally originating transfers, use the Trading Partner object to specify:

- Originating machine name
- Message format:
 - S/MIME
 - EDIINT: AS1, AS2 or AS3
 - RosettaNet
 - OFTP (for OFTP R2.0 security services)
 - SWIFTNet
- Security (signing, encryption type, certificate)
- Compression characteristics
- Receipt characteristics

For the reception of EDIINT or S/MIME transfers, use the Trading Partner object to specify:

- Confirmation of receipt delivery
- Obligatory signing (yes/no option)
- Obligatory encryption (yes/no option)

[Next topic](#)

[About EDIINT](#)

[Trading Partner objects](#)

Event management

Gateway includes a set of tools that enable you to program the triggering of tasks by events that occur on the Gateway. You can use this triggering mechanism for such tasks as:

- Routing
- Scheduled maintenance
- Scheduled transfers
- Directory polling
- Execution of script-based processes

Triggering Events

A broad range of events can trigger Gateway processes. Gateway groups possible triggering events into four categories:

Triggering Category	Description	Example
Transfer Change State	Enables you to use one or more attribute values of a Transfer to trigger a Gateway process.	The reception of a Transfer from a specified Axway Transfer product triggers the deposit of a new Transfer Request with a Diffusion List as destination.
Directory Scanning	Enables you to use the detection of a specific file or file type in a directory of the Gateway host machine to trigger a Gateway process.	The deposit of <code>example.txt</code> in the directory <code>C:\xwb_poll*</code> triggers the deposit of a Transfer Request that forwards the file to a specified Axway Transfer product.
Scheduling	Enables you to set the times when a specified action is triggered.	Gateway automatically archives a specified log file every day at the same time.
XMS	An Axway Messaging message transfer triggers the deposit of a Transfer Request.	Gateway detects a transfer over the Axway Messaging connector and applies the appropriate Model to the transfer.

Triggered Processes

When you select a triggering event, you link the event to a task that is executed by Gateway. The following table summarizes the types of task that Gateway can execute.

Triggered Process	Description
Model	Gateway applies a routing definition as specified in a Model object.
AMTrix/XIB	Gateway redirects a Transfer to AMTrix/ Axway Integrator for processing.
Perl Script	Gateway executes a Perl script.
Batch command	Gateway executes a specified online command.
Purge	Gateway purges a Mailbox or log file.
No action	Gateway takes no action.

Configuring Gateway to trigger processes

To create processing triggers and link them to events, you create *Decision Rule* objects and *Rule Table* objects.

- **Decision Rule** objects contain the definitions of the events that you want to trigger a specific action. They also specify the action that is carried out when the event occurs. You can create as many Decision Rules as you require.
- **Rule Table** objects are ordered lists of one or more Decision Rules. For each Decision Rule that it contains, the Rule Table displays the following parameters:
 - Order
 - Decision Rule name
 - Action to be executed
 - Condition parameters

The order in which a Decision Rule appears in the Rule Table list indicates its priority. Gateway analyzes all events and when an event corresponds to an event type of a Rule Table, it scans the table list in order, top to bottom. If it finds a rule whose criteria completely match the event, it executes the specified process specified by the rule.

[Next topic](#)

[Managing Events on Axway Gateway](#)

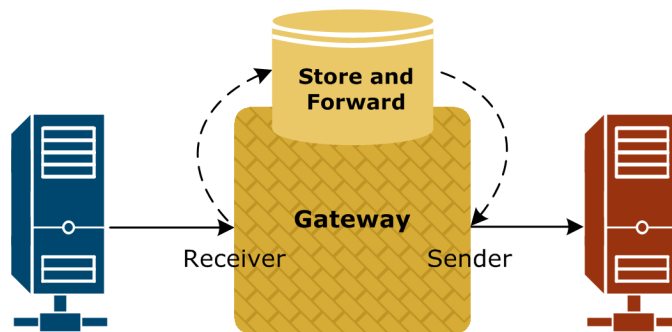
Axway Gateway: Overview

Routing

As described in the [previous topic](#), the Event management functions in Gateway enable you to use the occurrence of events to trigger processes. One important application of Event management functions is for routing.

Routing is the process of passing a message from one stage in a predefined processing path to the next stage. In most Gateway routing scenarios, Gateway acts as intermediate machine between an originating machine and the final recipient.

In its role as a gateway, Gateway operates as an intermediary site in a Store-and-Forward process, between the file sender and the file receiver. This means that, for each exchange that transits Gateway, at least two file transfer processes are involved, as illustrated in the following figure. Note that the sender and the receiver are represented in Gateway by Remote Site objects.



In Gateway, you automate routing by creating and managing Decision Rule objects. In the Decision Rule, you define a set of conditions to apply to the current transfer. For example:

- Protocol used for the current transfer = FTP
- Destination = Site A
- Destination is not the same as the originating Site
- ...

When all the Decision Rule conditions match those of the current transfer, Gateway applies the execution type specified for this Decision Rule to the transfer.

Example: To redirect an incoming transfer to Axway Integrator, create a Decision Rule that applies an XIB-type Model to the incoming transfer type. When Gateway detects an incoming transfer of the specified type, it applies the Model, which in turn submits a new Transfer Request to the desired Axway Integrator Site.

You can restrict routing to a limited set of transfers. For example, you could restrict the routing to a certain Site to transfers with the following characteristics:

- Transfer type: TRANS
- Transfer status: ENDED or ACKED

[Next topic](#)

[Managing Events on Axway Gateway](#)

Models

A *Model object* is a logical container that holds a set of reusable attributes. Models are useful when you need to repeatedly reuse a specific set of information to accomplish a series of tasks.

Gateway provides Model objects that enable you to group sets of attributes for the creation of:

- Transfer Requests
- Axway Messaging connector objects for Axway Messaging incoming and Axway Messaging outgoing events

Transfer Models

You can create Models that provide sets of attributes that complete Transfer Requests. You can specify the name of a Model when you create a Transfer Request or a Decision Rule. Then, when Gateway executes the Transfer Request or Decision Rule, it extracts attributes from the named Model.

Use the Model object to define parameters such as:

- Sender and receiver
- Application name
- Transfer validity date and time
- Scheduling priority
- User information
- ...

There are two types of Model for use with Transfer Requests and Decision Rules:

- **General Model**

The General Model defines a set of parameters that can be reused in a Transfer Request with any protocol. When you create a Model, you begin by creating a General Model.

There are some parameters that you can only define via the General Model, in particular, the **destination alias** (`dest_alias`), **remote agent** (`remote_agent`) and **originator alias** (`org_alias`) parameters.

In general, a General Model contains sufficient information for outgoing transfers.

- **Protocol Model**

After having defined a General Model, you can optionally define a Protocol Model with the same name. The Protocol Model contains a set of protocol-specific parameters.

A Protocol Model may be useful when you are submitting a Transfer Request that has a Diffusion List as its destination. If the members of the Diffusion List are serviced by different protocols, Gateway can use the Protocol Model to apply the appropriate protocol to each destination Site.

When you specify the Model name to use in a Transfer Request or a Decision Rule, Gateway first extracts the parameters specified in the Protocol Model and then completes the definition with parameters taken from the General Model with the same name.

Note: If you set the same parameters in both the Transfer Request and the Model objects, the Transfer Request parameters override the parameters you set in the Model.

Using Models in Transfer Requests

To accelerate the process of creating a Transfer Request, you can specify the name of a Model to use as one of the Transfer Request parameters.

Using Models in Decision Rules

As described in [Event management](#), when you can create Decision Rule objects to link Gateway processes to triggering events, you can set the execution type to *Model* and then specify the name of the Model to apply. You can specify the name of an existing Model or a Model that is not yet defined. If the Model is not already defined, complete the object definition before executing the Decision Rule.

When you implement Model-based routing via Decision Rules, you must specify the final destination for the transfer either by entering a value for the **destination alias** (`dest_alias`) or the **remote agent** (`remote_agent`). You can **only** define these two Transfer parameters in a General Model, and not in a Protocol Model.

When you use Gateway to route files via a Model object:

- The Model name that you specify must correspond to a General Model definition and can optionally correspond to a Protocol Model with the same name
- The specified General Model must include a final destination
- Either the General Model or the Protocol Model must contain values for the parameters: **file component** (`-file_component`) and **application** (`-appli`)

If you set the execution type to *Model* in a Decision Rule, it is recommended that you use symbolic variables in the Model definition. For example:

```
file_component = &(dir_path)/&(file_component)
application = &(appli)
destination_alias = &(dest_alias)
```

This ensures that when the Decision Rule invokes the Model to process a Transfer Request, the attributes that the Model uses for the transfer correspond to the attributes that are contained in the transfer that is being handled.

XMS Models

The Axway Messaging product serves as Messaging-Oriented Middleware. Axway Messaging components (servers, clients and administrative tools) provide an environment in which applications can exchange messages without directly communicating with each other. The Axway Messaging environment functions using Axway proprietary protocols.

Use the XMS Model object to define the set of connector attributes to support either:

- Incoming Axway Messaging messages originating from a specified Axway Messaging server and queue
- Outgoing messages to be sent to a specified Axway Messaging server message queue

[Next topic](#)

[About Models](#)

[Working with Transfer Models](#)

[Working with XMS Models](#)

Security Features

Information previously in this section have now been moved to the [Gateway Security guide](#) (requires a login).

Axway Gateway: Overview

Monitoring Gateway

The following features enable you to supervise and analyze Gateway transfer operations and processes.

Mailbox

The *Mailbox* is a permanent file in which Gateway stores all data relating to Transfer Requests. During the transfer process, all Transfer Request statuses and data are updated in the Mailbox. This stored information enables Gateway to restart and complete transfers in the case of failure during the transfer process.

To view and manage the contents of the Mailbox, use either the GUI or online commands.

Connections and Connected Users

To monitor the activities of the Gateway server, you supervise current connections. These are connections that are still open, whether they are active (transferring data) or not. To retrieve information about connections, use Connection and Connected User objects.

To view Connection and Connected User objects, use either the GUI or online commands.

Logs

Gateway stores traces of all monitoring, protocol, and network events in a *Log* file. You can configure Gateway to write different levels of detail to the file.

To consult log contents, either use the GUI or open the Log file.

Audit

Gateway stores the actions performed by users relating to the Gateway configuration in a dedicated audit file, to provide traceability of changes done to the product's configuration. This concerns changes both to global and to object configuration.

This functionality cannot be disabled, and the audit file can only be written by the audit process inside Gateway.

For more information, see [Managing Audit files on page 2396](#).

Traces

Gateway lets you activate the traces of Gateway processes without restarting the product. Traces settings are displayed as dynamic parameters in the unified configuration and changing their values using the `peluconf` utility or Gateway Navigator does not require a product restart. The only difference is for Gatecontroller, whose trace level needs to be set using the new `pelmon` subcommand `traceset`. For Gatecontroller, the updates made with the command `pelmon traceset` are only active until restart. There is no link between trace set in Navigator and trace set from the command line. The Gatecontroller trace parameter set in Navigator requires restart.

The list of trace level parameters is available in the advanced parameter configuration Gateway Navigator or using the `pelmon get` command, filtering by trace.

Some processes (`ipelsock`) may take some time to be applied requiring up to 120s to acknowledge the configuration change.

For Secure Relay you can now activate the traces from Gateway without restarting the product:

- Using the `peluconf` utility:
 - Set `-s xsr log_level N`, where $N\ 0 \leq N \leq 3$. If N is numerical but other than 0, 1, 2 or 3, an error message is displayed.
- In Gateway Navigator (**Log levels > Secure Relay**)

To verify that the trace level was set, look in the `log.properties` file in the `xsr` directory (`install_dir/run_time/xsr`). The line should appear as: `log4j.rootLogger=ERROR, XSR_ROLLINGFILE`.

Enabling root cause analysis

If a process fails to send heartbeats to Gatecontroller for a number of 5 configured heartbeat periods and the process is considered "vital", then Gatecontroller will restart Gateway. After the 4th missed heartbeat for one vital process, Gatecontroller will automatically trigger two actions:

- activate level 4 trace for the faulty process
- if configured to do so, will launch a debugger attached to the faulty process.

The configuration of the debugger command to launch is made through the static parameter `[gatecontroller]gc_debugger_cmd`

If between the 4th and the 5th heartbeat period, the process unblocks and sends its heartbeat to Gatecontroller, the root cause analysis procedure is reversed:

- trace is set back to level 0 on that process
- the debugger process is stopped

Limitation

For APIs used by Gateway to interoperate with other products (Sentinel API, Integrator API, X.400, X.420, Messaging), the dynamic trace level is not currently supported.

Statistics

If you select the corresponding option in the Gateway configuration menu, each transfer termination (correct ending or cancellation) triggers the deposit of information to a *Statistics file*. You can use the information in your Statistics files to analyze transfer performance.

Process monitoring

Gateway provides a centralized control of all its processes. GateController is a process that is always present. This process is responsible for starting, monitoring and stopping each permanent process within Gateway. It is able to give a reliable status of the product at any time. GateController is the only process you need to monitor in order to know the status of Gateway.

GateController starts and stops permanent processes. It is notified of starts and stops of any other process, and monitors every process of Gateway. Every permanent process sends heartbeats, so that GateController can detect crashes, loops, or any other loss of service. It can restart the entire Gateway application, or an individual process.

The `gatestatus` command enables you to monitor Gateway. It is a simple reference tool designed for the system administrator to check the functioning of Gateway. `gatestatus` messages don't necessarily indicate that there is problem, merely a condition that can only be cleared with user intervention.

Sentinel

Sentinel is a Business Activity Monitoring (BAM) product of the Axway Platform.

If your Axway product package includes both Gateway and Sentinel, you can configure Sentinel to monitor Gateway transfer activities.

[Next topic](#)

[Monitoring transfers](#)

[Monitoring Connections](#)

[Monitoring Connected Users](#)

[Log files](#)

[Managing Statistics files \(command line\)](#)

[About gatestatus messages](#)

[About Axway Sentinel](#)

[Configuring Gateway parameters on page 155](#)

Axway Gateway: Overview

DMZ deployment

When you link your enterprise network to the Internet, you can enhance local isolation and security by deploying Gateway in a DMZ configuration.

Axway offers two solutions for deploying Gateway in DMZ architectures:

- [Axway Secure Relay](#)
- [Paired Axway Gateways](#)

Axway Secure Relay (formerly XSR)

The components of Secure Relay, delivered with Gateway, enable you to deploy Gateway in a DMZ architecture. Secure Relay is a secure, multi-protocol, front-end product that prevents external partners from directly accessing Gateway for file exchange. Secure Relay processes data streaming of external connections from the outside world to Gateway without any data storage (even temporary) in the DMZ. Secure Relay is compatible with any Gateway supported protocol that uses a TCP/IP network.

When you use Gateway with Secure Relay, you derive the following specific benefits:

- All of your data is stored in your local network, protected by your firewall
- Connections from outside the protected area are refused
- Configuration data, including private server keys, are never stored in the DMZ
- No decryption occurs within the DMZ
- You can adapt the connection parameters of your Gateway to conform to the policies you establish in the DMZ

Distribution of functions

Native Gateway functions work together with the specific Secure Relay functions to provide DMZ security.

In a Secure Relay / DMZ configuration:

- Gateway:
 - Manages streaming connections with the Secure Relay Router Agent situated in the DMZ
 - Manages user access to file resources (workspaces, downloads and uploads)
 - Handles the file transfer protocol dialog with partners
 - Manages file transfer scheduling and monitoring
 - Can terminate transport security communication (either SSL or SSH for SFTP) to ensure a transparent link between partner identities (X.509 certificate) and file operations
 - Generates links with internal IT system applications by either routing files to other destinations or by direct integration
 - Handles physical file management operations
- Secure Relay:
 - Manages connections with the outside world for file exchange, depending on the Gateway directives
 - Handles data streaming of flows to and from the Internet
 - Can terminate transport security communication (either SSL or SSH for SFTP) to ensure a transparent link between partner identities (X.509 certificate) and file operations

Secure Relay architecture

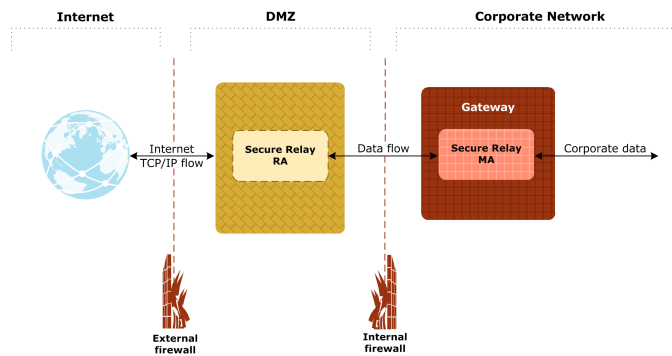
Secure Relay is divided into two basic components parts:

- **Secure Relay Master Agent**

The Secure Relay Master Agent is a component that resides with and is fully integrated in the Gateway product. All of the processes that support Secure Relay Master Agent functions are automatically included in Gateway on installation.

- **Secure Relay Router Agent**

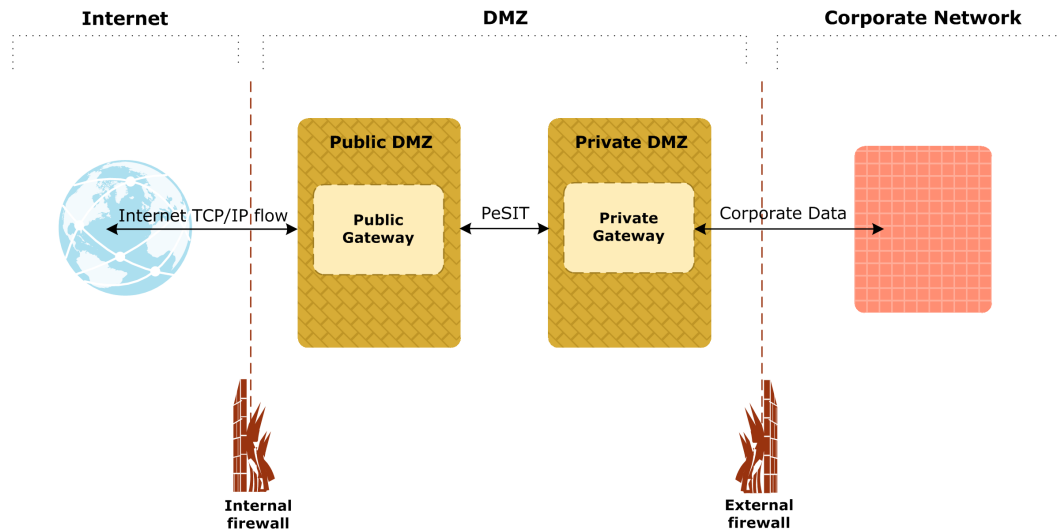
The Secure Relay Router Agent is a stand-alone component. You install the Secure Relay Router Agent on a machine located in the DMZ, independently of Gateway.



Paired Axway Gateways

Alternatively, to secure your local network when connecting to the Internet, you can deploy paired Gateways in a bridged private/public DMZ configuration.

The following diagram illustrates the use of Gateways in a DMZ bridge configuration.



With this configuration you can:

- Filter and control access attempts from the Internet
- Ensure secure end-to-end file data integrity and confidentiality on both sending and receiving transfers (transfer acknowledgments and monitoring)
- Scan received files for viruses
- Route files from the sender to the final destination

Operating principles

The paired Gateway DMZ operates according to the following principles:

Private Gateway

A Gateway serving as a *private* Gateway contains definitions that enable the applications situated in your corporate network to act as servers and clients for exchanges with Internet partners.

The files that the private Gateway receives from Internet partners are decrypted and scanned for viruses before being forwarded to the destination application in your corporate intranet.

In your private Gateway, to control the identity of your corporate partners, you create a Remote Site object definition for each partner.

In the private Gateway you create a Decision Rule that handles all transfer flow originating from your corporate network and directed towards the Internet. You use this Decision Rule apply a Routing Model that directs the transfer to the public Gateway via a secured protocol.

In the private Gateway you can also create a Decision Rule to handle the reception of files forwarded from the Internet by the public Gateway.

Public Gateway

A Gateway serving as a *public* Gateway contains definitions of all external Internet partners.

The temporary files that the public Gateway stores and all Gateway internal files (administration and Mailbox files) are ciphered to protect against malicious intrusion.

In your public Gateway, to establish the identities and control possible connection attempts of your external Internet partners, you create:

- Remote Site object definitions for Internet partners acting as servers
- CGate object definitions for Internet partners acting as clients

Additionally, in your public Gateway, you create a Decision Rule that handles all transfer flow originating from Internet partners. You use this Decision Rule to apply a Routing Model that directs the transfer to the private Gateway via a secured protocol.

Exchanges between Gateways

The public and private Gateways communicate via the PeSIT protocol (secured by SSL/TLS) or via SFTP.

Axway Gateway: User Interfaces

User interfaces

[Gateway user interfaces](#)

[Choosing a user interface](#)

GUI and command line interfaces

Gateway provides you with two interfaces that enable you to manage Gateway processes:

- **Windows Graphical User Interface**

The [Graphical User Interface \(GUI\)](#), also referred to as Gateway Navigator, provides frames, menus and icons you can use to manage the processes of your Gateway from a Windows client platform.

- **Command line interface**

You can directly enter [online commands](#) to manage Gateway processes from a UNIX session or from a Windows system command session.

This documentation includes descriptions of how to use both of these interfaces.

Choosing a user interface

You can execute most Gateway management tasks using either the GUI or the command line interface. For example, you can perform all tasks relating to the management of Transfer Requests, routing and security processes from either interface. One exception is that configuration can only be carried out using the GUI.

There are a certain number of tasks that you can only perform using the command line interface. These include commands that control server behavior (starting, stopping, load balancing, Mailbox management and so on) as well as the configuration of [user exits and APIs](#).

For quick reference to any online command described in this documentation, refer to:

- [Index of online commands](#)
- [Index of tasks](#)

[Using the Graphical User Interface \(GUI\)](#)

[About online commands](#)

[Index of online commands](#)

[Index of tasks](#)

[Catalog of online commands](#)

Axway Gateway: User Interfaces

Using the Graphical User Interface (GUI)

[About the Gateway GUI](#)

[Starting and stopping the GUI](#)

[How to set and modify your password](#)

[About the main window](#)

About the Gateway GUI

Gateway Navigator is the Gateway GUI (Graphical User Interface) provided for Windows platforms. Use this interface to manage the behavior and performance of your Gateway:

- Configure Gateway
- Create, manage and delete the objects that determine the Gateway processes
- Manage User Profiles and User Groups
- Consult log files
- Monitor connections
- Link events to Gateway processes
- Define parameters for Transfer Requests
- Manage Models for Transfer Requests
- Manage security features

Before reading and applying the contents of this topic, it is recommended that you are familiar with Gateway. For a short introduction, refer to Overview: About Axway Gateway.

Starting and stopping the GUI

How to start the Windows GUI

To start the Gateway GUI:

1. From the Windows **Start** menu, select **Programs > Axway Software > Axway [ProjectName] > Gateway > Start Navigator**.

Gateway opens the GUI main window.

2. In the left frame of the main window, double-click the name of your Local Site displayed in the directory structure.

Gateway displays the login dialog window.

3. Enter your user name and password, and click **Connection**.

In the left frame directory tree structure, Gateway displays the categories of objects that exist for your configuration, or that you can create. Use the GUI tools to create, manage and delete these objects.

How to stop the GUI

To stop the Gateway GUI:

1. From the **File** menu, select **Exit**.

Gateway displays a confirmation window.

2. Select **Yes** to exit and close the GUI.

How to set and modify your password

For local installations, by default the Gateway server is not protected by a password. To create a password for access as a local client, or modify an existing password:

1. From the Windows **Start** menu, select **Programs > Axway Software > Axway [ProjectName]> Start Navigator**

Alternatively, if you are using the Axway Gateway Navigator:

Select **Programs > Axway Software > Axway [ProjectName] > Gateway > Start Navigator**

Gateway opens the GUI main window.

2. In the left frame of the main window, double-click the name of your Local Site displayed in the directory structure.

Gateway displays the login dialog window.

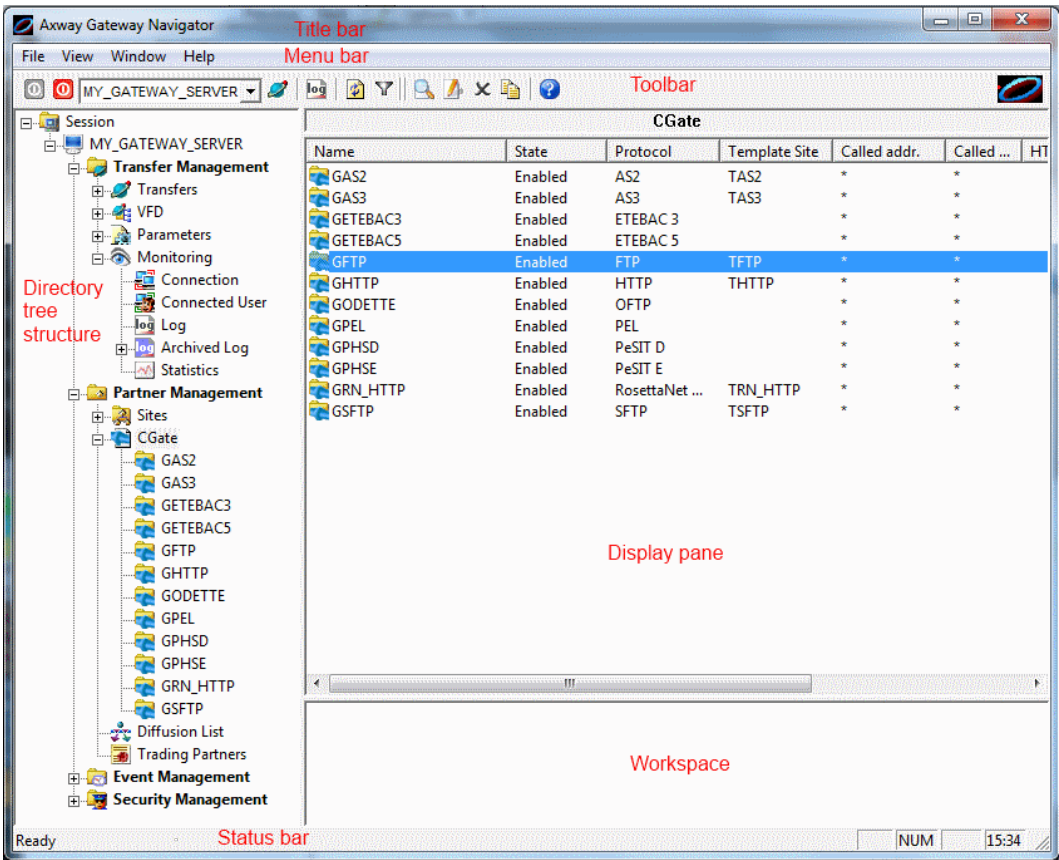
3. In the login dialog window:

- Enter a user name in the **User** field
- Enter a password in the **New password** field
- Reenter the same password in the **Confirm password** field
- Click **Connection**.

Gateway registers the new password and opens a GUI session. The new password is required for the next time you log in with the name you specified in the **User** field.

About the main window

When you start and log in to the GUI, Gateway displays the main window:



Title bar

The Title bar identifies the product you are using: **Axway Gateway Navigator**.

Menu bar

The Menu bar contains the following menus.

[File](#) [View](#) [Window](#) [Help](#)

File menu

The following table introduces the commands in the **File** menu.

Use the command...	To...
Print...	Print a Gateway formatted copy of the information visible in the display pane.

Use the command...	To...
Print preview	Preview the page layout of the information visible in the display pane.
Print setup...	Open the Windows printer configuration dialog box.
Save current settings	Conserve the settings that you choose from the <i>Menu bar</i> menus.
Exit	Close the Gateway GUI.

View menu

The **View** menu contains commands that enable you to customize the display of the information in the display pane.

Use the command...	To...
List	Display the information in the display pane in the form of a list of small icons accompanied by labels.
Details	Display the information in the display pane in the form of a columned list. Each column of the list contains information concerning the type of object you are viewing. The columns displayed (in other words, the details) vary according to the type of object you select for viewing.
List font...	Open the font selection window. Use this window to specify the size, color and font type of the text that is visible in the display pane.

Window menu

The **Window** menu contains commands that enable you to select the elements that are visible in the main window.

Use the command...	To...
Toolbar	Display, or alternatively, hide the Toolbar.
Status bar	Display, or alternatively, hide the Status bar.

Use the command...	To...
Workspace	Display, or alternatively, hide the Workspace.
Close all	Simultaneously close all of the dialog windows that are currently open in the display pane.

Below the list of commands in the **Window** menu, Gateway lists the names of the dialog windows and display windows that you have opened from the display pane.

Help menu

The following table introduces the commands in the **Help** menu.

Use the command...	To...
Contents...	Open this online help.
About...	Open the About... window. This window provides information about the product you are using. Additionally, the window displays your: • Available disk space • Product serial number When you finish reading the information in this window, click: • Close to close the window • Help to close the window and open the online help

Toolbar

The Toolbar displays a test field and a set of buttons that you can use to handle the objects and messages visible in the display pane.

You can choose to display or hide the Toolbar by using the **Toolbar** command of the [Window](#) menu.

The following table describes the elements of the **Toolbar**.

Element	Use
 Connect	Click to open the connection dialog box for the server displayed in the <i>Server selection/ display box</i> .

Element	Use
 Disconnect	Click to disconnect from the server displayed in the <i>Server selection/ display box</i> .
Server selection/ display box	Displays the name of the Gateway server to which you are applying the toolbar tools.
 Display transfer requests	Displays the contents of the Mailbox for the server displayed in the <i>Server selection/ display box</i> .
 Display log messages	Displays current log messages for the server displayed in the <i>Server selection/ display box</i> .
 Print	Prints the current contents of the display pane .
 Refresh	Refreshes the display of the display pane with most recently received data. Alternatively, click anywhere inside the display pane and enter F5 .
 Select and display	Opens the <i>Select Transfer</i> dialog box, enabling you to filter the display pane contents according to various criteria.
 View	Enables you to select the format in which the display pane content is presented.
 Modify	Opens an editing window that enables you to modify the element you select in the display pane .
 Delete	Deletes the element(s) you select in the display pane .
 Copy	Copies the elements you select in the display pane .
 About	Opens the Axway Gateway product information dialog box. This box displays: • Version and copyright information • Available disk space • Your product serial number • A Help button, that provides access to this online help

Directory tree structure

The left pane of the main window contains the *Directory tree structure*.

Expanding and reducing the tree structure

Click the  icons in this pane to expand, or alternatively, reduce the displayed development of the directory tree.

Viewing folder contents

Each folder and subfolder in the directory tree is represented by an icon accompanied by the folder or subfolder name. Folders at the lowest (expanded) level of each branch of the *Directory tree structure* can contain objects or messages. Click one of these folders, to view the objects or messages it contains in the display pane.

Example 1

When you click the **Log** folder in the **Transfer Management > Monitoring** subdirectory of a server directory, Gateway displays a list of log messages collected for that server in the display pane.

Example 2

When you click the **Local Site** folder of the **Partner Management > Sites** subdirectory of a server directory, Gateway displays a list of the Local Site objects that contain Site definitions for that server.

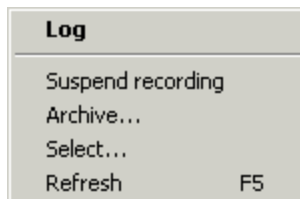
Managing folder contents: context menus

Folders at the lowest (expanded) level of each branch of the *Directory tree structure* can contain objects or messages.

Right-click any of these folders, to view a context menu of possible actions you can perform for the folder. The content of the context menus varies according to the folder type.

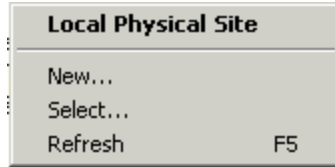
Example 1

When you right-click the **[ServerName] > Transfer Management > Monitoring > Log** folder, Gateway displays the following context menu.



Example 2

When you right-click the **[ServerName] > Partner Management > Sites > Local Site** folder, Gateway displays the following context menu.



In each example, Gateway displays a context menu that:

- Gives the name of the folder you right-clicked
- Lists the commands you can perform on the folder contents

When you select any of the commands listed in a context menu, Gateway either:

- Directly executes the command
- Opens a dialog window from which you execute the command by entering necessary information

For details on how to complete command dialog windows for the various Gateway objects and elements, refer to the related topics in this online help.

Display pane

The display pane displays the contents of the folder you select in the [Directory tree structure](#). Use the commands of the [View](#) menu to select the format in which the content is displayed.

User actions in the display pane

- **Click** any displayed element in the display pane to select that element. You can then use the tools of the [Toolbar](#) to perform actions on the element.
- **Double-click** any displayed object to open the editing window related to that object. The editing window varies according to the element type. In some cases the editing windows are read-only. For details concerning the contents of the editing windows for the various Gateway objects and elements, refer to the relevant topics in this online help.
- **Right-click** any element in the display pane to view a context menu of the possible actions on that element. The contents of the context menu varies according to the element type. For details of the possible actions you can perform on Gateway objects and elements, refer to the relevant topics in this online help.
- Press **Ctrl+F** to open the *Find* window. Use this window to search for a string in the display pane.

Workspace

When you open an editing window in the [display pane](#), and then click the window reduction icon in the upper right-hand corner of the editing window, Gateway reduces the window to a labeled button that is visible in the Workspace.

Display of the Workspace is optional. You can choose to display or hide the Workspace by using the **Workspace** command of the [Window](#) menu.

To restore an editing window to its working dimensions in the display pane, do one of the following:

- Click the icon that represents the editing window you want to reopen in the Workspace. Then select **Restore** in the context menu that is displayed.
- Double-click the icon that represents the editing window you want to reopen in the Workspace.

To close a single reduced editing window, the Workspace, click the icon that represents the window and then select the **Close** command from the displayed context menu.

To close all reduced editing windows as well as all other open editing windows, select the [Window](#) menu

Status bar

The Status bar provides information about the selection you make in the [Directory tree structure](#).

Display of the Status Bar is optional. You can choose to display or hide the Status bar by using the **Status bar** command of the [Window](#) menu.

The displayed information depends on the type of folder you select and on the state of the elements in the folder.

Axway Gateway: User Interfaces

Online commands

Overview

The Command Line Interface is a computer interface in which you enter commands in the form of a string of text characters.

You can manage virtually all Axway Gateway functions via the set of commands provided for the Command Line Interface.

You can use these commands:

- In interactive mode from the system prompt of Windows or UNIX systems
- Via batch mode procedures

You can use online commands to:

- Start and stop Gateway
- Manage the following Gateway objects:
 - Sites
 - Applications
 - Models
 - Diffusion Lists

- CGates and CGateGroups
- Connections
- Connected Users
- Transfer Requests
- Statistics
- Log files
- Decision Rules
- Rule Tables
- Proxies
- Virtual File Directories
- Network profiles
- Security profiles
- Certificates
- Keys

Before using the Command Line Interface

Before running online commands, set the Gateway environment variables.

Gateway provides a batch file that you can use to set all Gateway variables for your supporting OS:

- In UNIX environments, execute the **profile** batch file located in the <Gateway installation directory>/run_time/etc directory.
- In Windows environments, execute the **profile.bat** batch file located in the <Gateway installation directory>\run_time\etc directory.

In Windows environments, when you select **Start > Programs > Axway Software > Axway [ProjectName] > Gateway > Command Prompt** to open the command console, the environment variables are automatically taken into account.

Online commands and utilities

Commands

For a complete list of Gateway online commands, refer to [Catalog of online commands](#).

To locate a command or subcommand from an alphabetical index, refer to [Index of online commands](#).

To locate a command or subcommand from a list of command tasks, refer to [Index of tasks](#).

Utilities

In addition to commands, Gateway provides the following utilities that you can run from the Command Line Interface.

Utility	Description	Refer to
trncc	Builds character conversion tables used by Gateway code conversion modules	Code conversion and file formatting
trnxcode	Converts the contents of a file into a defined data code	Code conversion and file formatting
kfmcp	Maintains and optimizes the Mailbox file via the following actions: • Copies Mailbox records from input to output files • Sets or resets the primary and secondary allocation space for the Mailbox file	Maintaining the Mailbox (kfmcp command)
pelencpass	Encrypt a password using the AES 256 algorithm (for example, for PassPort AM/PM, SecureRelay, SWIFTNet HA Database). Generate an AES 256 encryption key (for example, for exporting/importing Gateway objects).	See <i>Security guide</i> > Secure Password Management for more information (requires login)

Syntax and conventions

Syntax

The syntax you use to enter commands and utilities varies according to the command or utility concerned. The following table summarizes syntax use.

Command or utility	Syntax
peladm pelbase pelctl pelmon peldsp secadm secdsp vfdadm vfddsp	command subcommand Parameter list
pelpur	command [subcommand] Parameter list
peltrans	command Parameter list
pelencpass	command Parameter list
trncc (utility)	trncc <infile>
trnxcode (utility)	trnxcode <codein.codeout> <infile> <outfile>
kfmcp (utility)	kfmcp [infile] outfile

Conventions for entering parameters

A parameters list can be empty, or it can contain one or several items.

Each parameter is identified by a keyword preceded by the minus ("-") character. Typically, a parameter is followed by a value. Certain optional parameters are restricted to the keyword only.

This section explains the syntactic conventions used throughout this documentation for command descriptions and examples.

The same syntax is used for the parameters list and values.

- Mandatory parameters are enclosed in curly brackets: { }
- Optional parameters are enclosed in square brackets: []
- Parameter abbreviations appear between parentheses: -param (-p)

Parameters without values

Certain parameters take no values. You use these parameters to enable a particular functionality. Parameters without values are always optional.

Example:

```
[-trace_mode (-tm)]
```

This parameter, abbreviated to **-tm**, enables protocol trace mode for the specified Site.

Typical command example:

To start a Site with trace mode enabled, enter the following command:

```
pelctl start_site -alias SITEA -trace_mode
```

Alternatively, use the abbreviated parameter notation:

```
pelctl start_site -a SITEA -tm
```

Parameters with explicit labels

Certain parameters are followed by an explicit label that is a numeric or alphanumeric string. This type of parameter can be mandatory or optional. The label is always mandatory when the parameter keyword is mentioned.

Example:

```
[-name (-n)] Application name
```

Typical command example:

To delete an Application, enter the following command:

```
peladm delete_appli -name APPLI
```

Alternatively, use the abbreviated parameter notation:

```
peladm delete_appli -n APPLI
```

Parameters with enumerated values

Enumerated values are a predefined set of values available for a parameter.

A parameter with enumerated values can be mandatory or optional. Any default values are indicated in parentheses.

Note: Curly brackets enclosed in curly or square brackets indicate that a value is mandatory when the corresponding parameter is explicitly mentioned.

Example:

```
[-protocol (-pr) { PEL | ODETTE | PHSE | PHSD | FTP | SMTP | POP3 | SFTP  
}]
```

Typical command example:

To select Remote Sites defined with the PEL protocol, enter the following command:

```
peldsp select_site -pr PEL
```

```
IMTPLX01
```


UXTPLX01

This command selects two Remote Sites.

Parameters with a range of numeric values

A parameter with a range of numeric values can be mandatory or optional. The upper and lower limits of the range are separated by dots.

Some of these parameters have default values that are enclosed in parentheses.

Example:

```
[-retry_count_max (-rc_max) { 0..(10)..999 }]
```

This command specifies the maximum number of retries allowed on a transfer. The optional parameter takes a numeric value from 0 to 999, where the default = 10.

Typical command example:

To update a Site, enter the following command (using the abbreviated parameter notation):

```
peladm update_site -a SITE -rc_max 5
```

Return values and error reporting

When you enter a command in the Command Line Interface, Gateway can return a message that indicates either the execution result or that an error has occurred. If a response is expected, the command displays the Gateway response in the system output.

A command exit code is provided in an environment variable. This command exit code can have one of the following values:

- **0** - command executed successfully
- **1** - execution error occurred
- **2** - syntax error occurred

This environment variable is platform-dependent. Refer to the platform-specific documentation for more information.

The error messages can be one of two types:

- Execution errors that occurred during command processing
- Errors resulting from command syntax errors

Execution errors

Execution errors are displayed in the format:

```
<command_name> <subcommand> : <error_message>
```

where:

<command_name> indicates the command procedure where the error occurred (for example: **peladm**)

<subcommand> indicates the name of the subcommand (for example: **create_site**)

<error_message> is the corresponding error text. All possible messages are listed in the file **itperr.msg**.

Example:

peladm update_site: Record not found.

Syntax errors

If you make an error in the use of syntax, a message is displayed in the following format:

<Error: description of error>

<Usage: command ? to display help>

where:

Error: description of the error

Usage: displays the syntax to use to display help for the command in question

Example:

If you enter the command:

peladm create_user uuuuuu

the following syntax error message is displayed:

Error: Invalid parameter uuuuuu

Usage: peladm create_user ? to obtain help on command parameters.

The following table contains the possible error messages that you may receive on entering a command.

Error type	Text format of displayed message
Syntax error messages	• Error: Missing value for parameter %s • Error: Invalid parameter %s • Error: Parameter %s too long (char), size must be %d • Error: Parameter %s is mandatory • Error: Parameter %s requires a numeric value • Error: Parameter %s must be greater than or equal to %d • Error: Parameter %s must be less than or equal to %d • Error: Incorrect value for parameter %s • Error: Invalid subcommand %s
Warning messages	• Warning: Parameter %s must be greater than or equal to %d • Warning: Parameter %s must be less than or equal to %d • Warning: Incorrect value for parameter %s
Help messages	• Usage: [command name] ? to obtain basic help, ?? for detailed help • Usage: [command name] ? to obtain help on command parameters • Usage: [command name]

Displaying online help

Online help is available for all Gateway online commands.

From the command prompt you can display three help levels:

- [Lists of the subcommands associated with a specific command](#)
- [Lists of the parameters of a specified subcommand](#)
- [Lists of all details of a command or a subcommand](#)

Displaying a list of subcommands associated with a specific command

To display a list of the subcommands comprising a specific command, from the command prompt enter the command name followed by a space and a single question mark (with or without quotation marks).

Syntax:

```
command ?
```

or

```
command "?"
```

Example:

```
peladm ?
```

This command returns a list of all subcommands comprising the `peladm` command:

```
peladm  
  
    create_site  
    update_site  
    delete_site  
    create_loc_site  
    update_loc_site  
    delete_loc_site  
    create_lsite  
    update_lsite  
    delete_lsite  
    create_appli  
    update_appli  
    delete_appli  
    create_list  
    update_list  
    delete_list  
    create_model  
    update_model  
    delete_model  
    create_user  
    update_user  
    delete_user  
    create_profile
```

```
update_profile
delete_profile
update_trans
create_cgate
update_cgate
delete_cgate
create_cgategroup
update_cgategroup
delete_cgategroup
update_ruletable
create_decisionrule
update_decisionrule
delete_decisionrule
create_proxy
update_proxy
delete_proxy
create_tradepart
update_tradepart
delete_tradepart
```

Displaying a list of the parameters of a specific subcommand

To display a list of the parameters comprising a specific subcommand, from the command prompt enter the command and subcommand, followed by a space and a single question mark (with or without quotation marks).

Syntax:

```
command subcommand ?
```

or

```
command subcommand "?"
```

Example:

```
peladm create_user ?
```

This command returns a list of all parameters comprising the `create_user` subcommand.

```
peladm    create_user

{-username (-n)}

[-profile (-pr)]

[-group (-gr) group_name]

[-password (-pswd)]

[-comments (-cts)]

[-expired (-exd)]

[-maxretry (-mxr)]

[-administrator (-admin) {Y | (N)}

[-allxfer (-ax) {Y | (N)}

[disabled (-dd) {Y | (N)}
```

Displaying all details of a command and each available subcommand

To display details of all the parameters of a command and each available subcommand, enter the name of the command.

To display the complete parameter details of only a subcommand, enter the name of the command, the name of the subcommand and double question marks.

Syntax:

```
command ??
```

or

```
command subcommand ??
```

Example:

```
pelctl ??
```

This command returns the details:

```
pelctl

    start_site

        {-alias (-a)}

        [-init_mode (-im)]

        [-resp_mode (-rm)]

        [-commpath_used (-cu) { MAIN | ALT1 | ALT2 | ALT3 }]

        [-trace_mode (-tm)]
```

```
        [-cdi_mode (-cdm)]

stop_site

        {-alias (-a)}

        [-init_mode (-im)]

        [-resp_mode (-rm)]

        [-trace_mode (-tm)]

        [-cdi_mode (-cdm)]

control_trans

        {-local_ident (-i)}

        {-action (-a) { C | S | R }}

        [-retry_delay (-rd)]

archive_log

suspend_log

resume_log

archive_stat

suspend_stat

resume_stat

load_net_security

load_tcp_org_port

trace_process

        {-process_name (-pname)}

        {-trace_level (-l) { 0..(0)..4 }}

        [-all (-a) to trace process and entity]

trace_entity

        {-entity_name (-ename)}

        {-trace_level (-l) { 0..(0)..4 }}

kill_connection

        {-connection_id (-ci)}

disconnect_user

        {-connected_user_id (-cui)}

detect_swift_recovery
```

Displaying details of a command without subcommands

The `peltrans`, `pelpur`, and `pelmig` commands have no associated subcommands. You obtain the same help display with either of the following command formats:

`command ?`

and

`command ??`

Axway Gateway: User Interfaces

Index of online commands

The following table lists all Gateway online commands in alphabetical order. For each command, the table lists the subcommands comprising the main command. Click the command or subcommand to follow a hyperlink to the description in this document.

Command	Subcommand
kfmcp	no subcommands

Command	Subcommand
peladm	peladm check model

Command	Subcommand
	<u>peladm create appli</u>

Command	Subcommand
	<code>peladm create cgate</code>

Command	Subcommand
	<code>peladm create cgategroup</code>

Command	Subcommand
	<u>peladm create decisionrule</u>

Command	Subcommand
	<code>peladm create list</code>

Command	Subcommand
	<u>peladm create loc site</u>

Command	Subcommand
	<u>peladm create model</u>

Command	Subcommand
	peladm create_profile see Security Guide > Working with Profiles - command line (requires login) peladm create proplist peladm create proxy peladm create purge model peladm create ruletable peladm create site peladm create tradepart peladm create_user. See Security Guide > Working with users (command line) peladm create xmsctor peladm delete appli peladm delete cgate peladm delete cgategroup peladm delete decisionrule peladm delete list peladm delete loc site peladm delete model peladm delete_profile see Security Guide > Working with Profiles - command line (requires login) peladm delete proplist peladm delete proxy peladm delete purge model peladm delete ruletable peladm delete site

Command	Subcommand
	peladm delete tradepart
	peladm delete_user See Security Guide > Working with users (command line)
	peladm delete xmsctor
	peladm update appli
	peladm update cgate
	peladm update cgategroup
	peladm update decisionrule
	peladm update list
	peladm update loc site
	peladm update model
	peladm update_profile see Security Guide > Working with Profiles - command line (requires login)
	peladm update proplist
	peladm update proxy
	peladm update purge model
	peladm update ruletable
	peladm update site
	peladm update tradepart
	peladm update trans
	peladm update_user See Security Guide > Working with users (command line)
	peladm update xmsctor

Command	Subcommand
<u>pelbase</u>	<u>pelbase export appli</u>
	<u>pelbase export cgate</u>
	<u>pelbase export cgategroup</u>
	<u>pelbase export decisionrule</u>
	<u>pelbase export list</u>
	<u>pelbase export model</u>
	<u>pelbase export purge model</u>
	<u>pelbase export profile</u>
	<u>pelbase export proplist</u>
	<u>pelbase export proxy</u>
	<u>pelbase export ruletable</u>
	<u>pelbase export sgate</u>
	<u>pelbase export sgategroup</u>
	<u>pelbase export site</u>
	<u>pelbase export tradepart</u>
	<u>pelbase export user</u>
	<u>pelbase import</u>

Command	Subcommand
<u>pelctl</u>	<u>pelctl acquire swqueue</u>

Command	Subcommand
	<code>pelctl archive log</code>

Command	Subcommand
	<code>pelctl archive stat</code>

Command	Subcommand
	pelctl check asp status
	pelctl close input channel
	pelctl create input channel
	pelctl close output channel
	pelctl control trans
	pelctl create output channel
	pelctl delete input channel
	pelctl delete output channel
	pelctl disconnect user
	pelctl detect swift recovery
	pelctl kill connection
	pelctl open input channel
	pelctl open output channel
	pelctl release swqueue
	pelctl reload asp profiles
	pelctl reset connected user
	pelctl resume log
	pelctl resume stat
	pelctl start site
	pelctl stop site
	pelctl suspend log
	pelctl suspend stat
	pelctl trace entity
	pelctl trace process

Command	Subcommand
peldconf	<i>deprecated</i>

Command	Subcommand
peldsp	peldsp display appli

Command	Subcommand
	<u>peldsp display cgate</u>

Command	Subcommand
	<u>peldsp display cgategroup</u>

Command	Subcommand
	<u>peldsp display connected user</u>

Command	Subcommand
	<u>peldsp display connection</u>

Command	Subcommand
	<u>peldsp display decisionrule</u>

Command	Subcommand
	<u>peldsp display list</u>

Command	Subcommand
	<u>peldsp display loc site</u>

Command	Subcommand
	<u>peldsp display model</u>

Command	Subcommand
	<code>peldsp display_profile</code> see Security Guide > Working with Profiles - command line (requires login)
	<code>peldsp display proplist</code>
	<code>peldsp display proxy</code>
	<code>peldsp display purge model</code>
	<code>peldsp display ruletable</code>
	<code>peldsp display site</code>
	<code>peldsp display tradepart</code>
	<code>peldsp display trans</code>
	<code>peldsp display_user</code> See Security Guide > Working with users (command line)
	<code>peldsp display xmsctor</code>
	<code>peldsp select appli</code>
	<code>peldsp select cgate</code>
	<code>peldsp select cgategroup</code>
	<code>peldsp select connected user</code>
	<code>peldsp select connection</code>
	<code>peldsp select decisionrule</code>
	<code>peldsp select list</code>
	<code>peldsp select loc site</code>
	<code>peldsp select model</code>
	<code>peldsp select_profile</code> see Security Guide > Working with Profiles - command line (requires login)
	<code>peldsp select proplist</code>
	<code>peldsp select proxy</code>

Command	Subcommand
	peldsp select purge model
	peldsp select ruletable
	peldsp select site
	peldsp select tradeport
	peldsp select trans
	peldsp select_user See Security Guide > Working with users (command line)
	peldsp select xmsctor
	peldsp status connected user
	peldsp status connection
	peldsp status loc site
	peldsp status site
	peldsp status swoutputsession
	peldsp status trans
pelencpass	pelencpass encrypt_pass (See Security guide)
pelinst	no subcommands
pelmig	no subcommands
pelmon	pelmon change pass phrase
	pelmon check base
	pelmon init base
	pelmon rebuild vfd
	pelmon recover base
	pelmon reset
	pelmon restart

Command	Subcommand
<u>pelpur</u>	no subcommands or <u>xferPurge</u> <u>clean swa db</u>
<u>peltrans</u>	no subcommands
<u>secadm</u>	See Security guide > <u>Overview of SECS online commands</u> (requires login)
secbase	See Security guide > <u>search for secbase</u> (requires login)
<u>secdsp</u>	See Security guide > <u>Overview of SECS online commands</u> (requires login)
<u>vfdadm</u>	<u>vfdadm create dir</u> <u>vfdadm delete dir</u> <u>vfdadm move dir</u> <u>vfdadm update dir</u>
<u>vfdbase</u>	<u>vfdbase export</u> <u>vfdbase import</u>
<u>vfd dsp</u>	<u>vfd dsp display dir</u> <u>vfd dsp display file</u> <u>vfd dsp list dir</u>

Axway Gateway: User Interfaces

Index of tasks

The following table lists common tasks you can execute using online commands. Tasks are listed in alphabetical order. For each task, the table gives the associated command and subcommand. Click the command or subcommand to follow a hyperlink to the command description.

Task		Command/Subcommand
Archive	• Log files	pelctl archive log
	• Statistics files	pelctl archive stat
Change encryption passphrase file		pelmon change pass phrase
Check Gateway Mailbox integrity		pelmon check base
Control Transfer Request		pelctl control trans

Task		Command/Subcommand
Create	• Application object	peladm create appli
	• CGate object	peladm create cgate
	• CGateGroup object	peladm create cgategroup
	• Decision Rule object	peladm create decisionrule
	• Distribution List object	peladm create list
	• Local Site object	peladm create loc site
	• Model object	peladm create model
	• Network Security Profile	secadm create netprof
	• Profile object	peladm create profile
	• Property List	peladm create proplist
	• Proxy	peladm create proxy
	• Purge Model	peladm create purge model
	• Rule Table object	peladm create ruletable
	• Site object	peladm create site
	• Security Profile	secadm create sprof
	• SSH Security Profile	secadm create sshprof
	• Trading Partner (EDIINT)	peladm create tradepart
	• User object	peladm create user
	• Virtual File Directory	vfdadm create dir
	• Axway Messaging connector	peladm create xms connector

Task	Command/Subcommand
Delete	• Application object peladm delete appli
	• Certificate secadm delete cert
	• CGate object peladm delete cgate
	• CGateGroup object peladm delete cgategroup
	• Decision Rule object peladm delete decisionrule
	• Diffusion List object peladm delete list
	• Key (security) secadm delete key
	• Local Site object peladm delete loc site
	• Model object peladm delete model
	• Network Security Profile secadm delete netprof
	• Profile object peladm delete profile
	• Property List peladm delete proplist
	• Proxy peladm delete proxy
	• Purge Model peladm delete purge model
	• Rule Table peladm delete ruletable
	• Security Profile secadm delete sprof
	• Site object peladm delete site
	• SSH Security Profile secadm delete sshprof
	• User object peladm delete user
	• Trading Partner (EDIINT) peladm delete tradepart
	• Virtual File Directory vfdadm delete dir

Task		Command/Subcommand
	Axway Messaging connector	<code>peladm delete xmsctor</code>
Detect	SWIFTnet system recovery	<code>pelctl detect swift recovery</code>
Disconnect User		<code>pelctl disconnect user</code>

Task		Command/Subcommand
Display	• Application object	peldsp display appli
	• Certificate	secdsp display cert
	• CGate object	peldsp display cgate
	• CGateGroup object	peldsp display cgategroup
	• Connected users	peldsp display connected user
	• Connections	peldsp display connection
	• Decision Rule object	peldsp display decisionrule
	• Diffusion List object	peldsp display list
	• List of specified VFD contents	vfddsp list dir
	• Key (security)	secdsp display key
	• Key status	<code>secdsp status_key</code> (See Security guide)
	• Local Site object	peldsp display loc site
	• Model object	peldsp display model
	• Network Profile	secdsp display netprof
	• Network profile status	<code>secdsp status_netprof</code> (See Security guide)
	• Profile object	peldsp display profile
	• Proxy	peldsp display proxy
	• Purge Model	peldsp display purge model
	• Rule Table object	peldsp display ruletable
	• Site object	peldsp display site

Task	Command/Subcommand
The initial and dynamic states of a Connected User	<code>peldsp status connected user</code>
The initial and dynamic states of a Connection	<code>peldsp status connection</code>
Parameters of a Local Site	<code>peldsp status loc site</code>
The initial and dynamic states of a Remote Site	<code>peldsp status site</code>
Selected list of Transfer Request with a subset of information for each selected Request	<code>peldsp status trans</code>
Encrypt password (e.g. for PassPort AM/PM, SecureRelay, SWIFTNet HA Database)	<code>pelencpass encrypt_pass</code> (See Security guide)
Security Profile	<code>secdsp display sprof</code>
Security Profile status	<code>secdsp_status_sprof</code> (See Security guide)
SSH Security Profile	<code>secdsp display sshprof</code>

Task		Command/Subcommand
	• SSH Security Profile status	<code>secdsp status_sshprof</code> (See Security guide)
	• Trading Partner (EDIINT)	<code>peldsp display tradepart</code>
	• Transfer Request	<code>peldsp display trans</code>
	• User object	<code>peldsp display user</code>
	• Virtual File Directory details	<code>vfddsp display dir</code>
	• VFD logical file details	<code>vfddsp display file</code>

Task		Command/Subcommand
Export	• Application object	pelbase export appli
	• CGate object	pelbase export cgate
	• CGateGroup object	pelbase export cgategroup
	• Decision Rule object	pelbase export decisionrule
	• Diffusion List object	pelbase export list
	• Model object	pelbase export model
	• Network Security Profile object	secbase export_netprof (See Security guide)
	• Profile object	pelbase export profile
	• Property List object	pelbase export proplist
	• Proxy object	pelbase export proxy
	• Rule Table object	pelbase export ruletable
	• Security certificate	secadm export_cert (See Security guide)
	• SGate object	pelbase export sgate
	• SGateGroup object	pelbase export sgategroup
	• Site object	pelbase export site
	• TLS Security Profile object	secbase export_sprof (See Security guide)
	• SSH Security Profile object	secbase export_sshprof (See Security guide)
	• Trading Partner object	pelbase export tradepart
	• User object	pelbase export user

Task		Command/Subcommand
	• Virtual File Directory	vfdbase export
	• Generate an export encryption key	pelencpass generate_key (See Security guide)
Import	• Key (security)	secadm import key
	• Network Security Profile	secbase import (See Security guide)
	• Object	pelbase import
	• Security certificate	secadm import cert
	• Security Profile	secadm import sprof
	• SSH Security Profile	secadm import sshprof
	• Virtual File Directory	vfdbase import
Initialize the Gateway Mailbox		pelmon init base
Kill a connection		pelctl kill connection
Upgrade the Mailbox file between different versions of Gateway		pelmig
Move Virtual File Directory		vfdadm move dir
Obtain trace data	• For Gateway processing	pelctl trace process
	• For a Gateway entity (sub-task of a process)	pelctl trace entity
Purge the Mailbox of Transfer Requests		pelpur or pelpur xfer peladm create purge model
Purge the transfer entries from HA database. (Used by SWIFTNet High Availability)		pelpur clean swha db
Recover the Gateway Mailbox		pelmon recover base

Task		Command/Subcommand
Repair the Virtual File Directory		pelmon rebuild vfd
Reset connected user		pelctl reset connected user
Reset the status prompted by the gatestatus command to the default value		pelmon reset
Restart Gateway		pelmon restart
Resume	• Log file recording	pelctl resume log
	• Statistics file recording	pelctl resume stat
Retrieve status	• Certificate	secdsp status cert
	• Key (security)	secdsp status key
	• Network Profile	secdsp status netprof
	• Security Profile	secdsp status sprof
	• SSH Security Profile	secdsp status sshprof

Task	Command/Subcommand
Select	• Application object peldsp select appli
	• Certificate secdsp select cert
	• CGate object peldsp select cgate
	• CGateGroup object peldsp select cgate group
	• Connected user peldsp select connected user
	• Connection peldsp select connection
	• Diffusion List object peldsp select list
	• Key (security) secdsp select key
	• Local Site object peldsp select loc site
	• Model object peldsp select model
	• Network Profile secdsp select netprof
	• Profile object peldsp select profile
	• Proxy peldsp select proxy
	• Purge Model peldsp select purge model
	• Rule Table object peldsp select ruletable
	• Security Profile secdsp select sprof
	• Site object peldsp select site
	• SSH Security Profile secdsp select sshprof
	• Trading Partner (EDIINT) peldsp select tradepart
	• Transfer Request peldsp select trans
	• User object peldsp select user

Task		Command/Subcommand
Start	• A Site	pelctl start site
	• Gateway	gatestart
Stop	• A Site	pelctl stop site
	• Gateway	gatestop
Submit a Transfer Request, a message Request, or a Diffusion List Request		peltrans
Suspend	• Log file recording	pelctl suspend log
	• Statistics file recording	pelctl suspend stat
Test Gateway status		gatestatus
Trace	• A server entity	pelctl trace entity
	• A server process	pelctl trace process

Task		Command/Subcommand
Update	• Application object	peladm update appli
	• Certificate	secadm update cert
	• CGate object	peladm update cgate
	• CGateGroup object	peladm update cgategroup
	• Decision Rule object	peladm update decisionrule
	• Diffusion List object	peladm update list
	• Key (security)	secadm update key
	• Local Site object	peladm update loc site
	• Model object	peladm update model
	• Network Security Profile	secadm update netprof
	• Profile	peladm update profile
	• Proxy	peladm update proxy
	• Purge Model	peladm update purge model
	• Rule Table object	peladm update ruletable
	• Security profile	secadm update sprof
	• SSH Security profile	secadm update sshprof
	• Site object	peladm update site
	• Trading Partner (EDIINT)	peladm update tradepart
	• Transfer Request	peladm update trans
	• User object	peladm update user
	• Virtual File Directory	vfdadm update dir

Catalog of online commands

This topic lists all Axway Gateway online commands and related subcommands.

For each command, this topic provides:

- A brief description of the use of the command
- The command syntax
- A list of subcommands
- Links to topics that provide more detailed information

Online commands

• peladm	• pelimport	• secadm
• pelbase	• pelinst	• secdsp
• pelctl	• pelmig	• vfdadm
• peldconf	• pelmon	• vfddsp
• peldsp	• pelpur	
• pelexport	• peltrans	

Locating the appropriate command

To locate a command from a list of command task descriptions, refer to [Index of tasks](#).

For an overview of online command use, syntax and help sources, refer to [About online commands](#).

Note If you follow links to the Security guide on the Axway documentation portal, you will be required to be logged in.

peladm

Purpose

Use the `peladm` command to create, modify, and delete the following objects in the Gateway database:

- Sites
- Applications
- Models

- Diffusion Lists
- CGates
- CGateGroups
- Users and Profiles
- Decision Rules
- Rule Table
- Proxies

The `peladm` subcommands identify the object and the type of action to perform. For example, `peladm delete_site` identifies the object as a Remote Site and specifies deletion as the action to perform.

Each subcommand is associated with a set of parameters that define object properties.

Syntax

`peladm [subcommand] [parameters]`

Subcommands

This table lists the subcommands of the `peladm` command.

Subcommand	Use to...	Additional information
<code>peladm create_site</code>	create Remote Sites	Working with Remote Sites (command line)
<code>peladm update_site</code>	update Remote Sites	
<code>peladm delete_site</code>	delete Remote Sites	
<code>peladm create_loc_site</code>	create Local Sites	Working with Local Sites (command line)
<code>peladm update_loc_site</code>	update Local Sites	
<code>peladm delete_loc_site</code>	delete Local Sites	

Subcommand	Use to...	Additional information
<code>peladm create_appli</code>	create Applications	
<code>peladm update_appli</code>	update Applications	Working with Applications (command line)
<code>peladm delete_appli</code>	delete Applications	
<code>peladm create_list</code>	create Diffusion List	
<code>peladm update_list</code>	update Diffusion List	Working with Diffusion Lists (command line)
<code>peladm delete_list</code>	delete Diffusion List	
<code>peladm create_model</code>	create Models	
<code>peladm update_model</code>	update Models	Working with Models (command line)
<code>peladm delete_model</code>	delete Models	
<code>peladm create_user</code>	create Users	See the <i>Security Guide</i> > Working with Users (command line) for further information.
<code>peladm update_user</code>	update Users	
<code>peladm delete_user</code>	delete Users	
<code>peladm create_profile</code>	create Profiles	See the <i>Security Guide</i> > Working with Profiles (command line) for further information.
<code>peladm update_profile</code>	update Profiles	
<code>peladm delete_profile</code>	delete Profiles	

Subcommand	Use to...	Additional information
<code>peladm create_purge_model</code>	create Purge Model	
<code>peladm update_purge_model</code>	update Purge Model	Working with Purge Models (command line)
<code>peladm delete_purge_model</code>	delete Purge Model	
<code>peladm update_trans</code>	update Transfers	Working with Transfers (command line)
<code>peladm create_cgate</code>	create CGates	
<code>peladm update_cgate</code>	update CGates	Working with CGates (command line)
<code>peladm delete_cgate</code>	delete CGates	
<code>peladm create_cgategroup</code>	create CGateGroups	
<code>peladm update_cgategroup</code>	update CGateGroups	Working with CGateGroups (command line)
<code>peladm delete_cgategroup</code>	delete CGateGroups	
<code>peladm create_ruletable</code>	create Rule Tables	
<code>peladm update_ruletable</code>	set or modify the properties of the Rule Table	Working with Rule Tables (command line)
<code>peladm delete_ruletable</code>	delete Rule Tables	

Subcommand	Use to...	Additional information
<code>peladm create_decisionrule</code>	create Decision Rules	
<code>peladm update_decisionrule</code>	update Decision Rules	Working with Decision Rules (command line)
<code>peladm delete_decisionrule</code>	delete Decision Rules	
<code>peladm create_proplist</code>	create Axway Messaging connector Properties List	Working with the Axway Messaging connector (command line)
<code>peladm update_proplist</code>	update Axway Messaging connector Properties List	
<code>peladm delete_proplist</code>	delete Axway Messaging connector Properties List	
<code>peladm create_proxy</code>	create Proxies	Working with Proxies (command line)
<code>peladm update_proxy</code>	update Proxies	
<code>peladm delete_proxy</code>	delete Proxies	
<code>peladm create_tradeport</code>	create Trading Partners	Working with Trading Partner objects (command line)
<code>peladm update_tradeport</code>	update Trading Partners	
<code>peladm delete_tradeport</code>	delete Trading Partners	

pelbase

Purpose

Use the `pelbase` command to export Gateway objects from the database to an ASCII file or import objects from an ASCII file to the database.

You can export records in different formats depending on your requirements, so that you can adapt the records to your operating system (for example, you can complete the fields with environment variables).

Syntax

pelbase [subcommand] [parameters]

Subcommands

This table lists the subcommands of the **pelbase** command.

Subcommand	Use to...	Additional information
<code>pelbase export_site</code>	export a Remote Site	Importing and exporting objects
<code>pelbase export_appli</code>	export an Application	
<code>pelbase export_model</code>	export a Model	
<code>pelbase export_purge_model</code>	export a Purge Model	
<code>pelbase export_list</code>	export a Diffusion List	
<code>pelbase export_user</code>	export a User	
<code>pelbase export_profile</code>	export a Profile	
<code>pelbase export_cgate</code>	export a CGate	
<code>pelbase export_cgategroup</code>	export a CGateGroup	
<code>pelbase export_sgate</code>	export an SGate	
<code>pelbase export_sgategroup</code>	export an SGateGroup	
<code>pelbase export_proxy</code>	export a Proxy	
<code>pelbase export_decisionrule</code>	export a Decision Rule	
<code>pelbase export_ruletable</code>	export a Rule Table	
<code>pelbase import</code>	import an object	

pelctl

Purpose

Use the `pelctl` command to:

- Start and stop the activity on a Site
- Start, stop and cancel actions relating to Transfer states
- Suspend or resume Log file recording and archiving
- Suspend or resume Stat file recording and archiving
- Manage tracing processes

The related subcommands identify the object you want to select. The subcommands are associated with a set of selection parameters.

Syntax

`pelctl` [subcommand] [parameters]

Subcommands

This table lists the subcommands of the `pelctl` command.

Subcommand	Use to...	Additional information
<code>pelctl start_site</code>	Start a Site	Working with Remote Sites (command line)
<code>pelctl stop_site</code>	Stop a Site	
<code>pelctl control_trans</code>	Control a Transfer	Working with Transfers (command line)
<code>pelctl archive_log</code>	Archive the Log	pelctl archive_log
<code>pelctl suspend_log</code>	Suspend Log recording	pelctl suspend_log
<code>pelctl resume_log</code>	Resume Log recording	pelctl resume_log
<code>pelctl archive_stat</code>	Archive Statistics file	pelctl archive_stat

Subcommand	Use to...	Additional information
<code>pelctl suspend_stat</code>	Suspend Statistics recording	pelctl suspend_stat
<code>pelctl resume_stat</code>	Resume Statistics recording	pelctl resume_stat
<code>pelctl load_net_security</code>	<i>This command is present for compatibility purposes only and should no longer be used. It has been replaced by Network Profile.</i> Reload the content of the <code>tablex25.adr</code> and <code>tableip.adr</code> files.	
<code>pelctl load_tcp_org_port</code>	<i>This command is present for compatibility purposes only and should no longer be used.</i> Reload the content of the file <code>tcpoport.ini</code> after it has been modified. The file <code>tcpoport.ini</code> is used to assign the origin TCP port number for outgoing connections.	
<code>pelctl trace_process</code>	Obtain trace data for Gateway processing	pelctl trace_process
<code>pelctl trace_entity</code>	Obtain trace data for a Gateway entity (sub-task of a process)	pelctl trace_entity
<code>pelctl trace_rotate</code>	Trigger the trace rotation	Managing rotation and archiving of trace files
<code>pelctl kill_connection</code>	Kill a Connection	Monitoring Connections (command line)

Subcommand	Use to...	Additional information
<code>pelctl disconnect_user</code>	Disconnect a User	
<code>pelctl reset_connected_user</code>	Resets user quota to zero. An administrator can use this command to reassign access to users who have exceeded their quotas.	Monitoring Connected Users (command line)
<code>pelctl detect_swift_recovery</code>	Initiates a manual system recovery detection procedure	SWIFTNet system recovery
<code>pelctl acquire_swqueue</code>	Acquire a SWIFTNet queue.	Configuring the Axway Gateway/SWIFTNet environment
<code>pelctl release_swqueue</code>	Release a SWIFTNet queue.	
<code>pelctl open_output_channel</code>	Open a SWIFTNet output channel.	SWIFTNet queues and output channels
<code>pelctl close_output_channel</code>	Close a SWIFTNet output channel.	
<code>pelctl reload_asp_profiles</code>	Reload SWIFTNet ASP profiles.	SWIFTNet ASP check
<code>pelctl check_asp_status</code>	Check SWIFTNet ASP profiles status.	
<code>pelctl open_input_channel</code>	Open a SWIFTNet input channel.	SWIFTNet input channels
<code>pelctl close_input_channel</code>	Close a SWIFTNet input channel.	
<code>pelctl delete_input_channel</code>	Delete a SWIFTNet input channel.	SWIFTNet queues and output channels
<code>pelctl create_output_channel</code>	Create a SWIFTNet output channel.	
<code>pelctl delete_output_channel</code>	Delete a SWIFTNet output channel.	

Subcommand	Use to...	Additional information
<code>pelctl reload_xsr_ipfilter_file</code>	Reload the blacklist (black list) file dynamically, without restarting Gateway.	

peldconf

Purpose

This command is now deprecated.

The `peldconf` command was previously used to dynamically modify the Gateway configuration.

peldsp

Purpose

Use the `peldsp` command to select the following objects defined in the administration database and display their parameters:

- Sites
- Applications
- Models
- Diffusion Lists
- Users and Profiles
- CGates and CGateGroups
- Connections and Connected Users
- Diffusion List and Transfer Requests
- Decision Rules
- Rule Tables
- Properties lists
- Proxies
- Axway Messaging connectors

Use subcommands with `peldsp` to identify the type of object you want to select. The subcommands are associated with selection parameters.

Syntax

peldsp [subcommand] [parameters]

Subcommands

This table lists the subcommands of the **peldsp** command.

Subcommand	Use to...	Additional information
peldsp select_site	display a selected list of Remote Sites	
peldsp display_site	display the attributes of a Remote Site	Working with Remote Sites (command line)
peldsp status_site	display the initial and dynamic states of a Remote Site	
peldsp select_loc_site	display a selected list of Local Sites	
peldsp display_loc_site	display the parameters of a Local Site	Working with Local Sites (command line)
peldsp status_loc_site	display the initial and dynamic states of a Local Site	
peldsp select_appli	display a selected list of Applications	Working with Applications (command line)
peldsp display_appli	display the parameters of an Application	
peldsp select_model	display a selected list of Models	Working with Models (command line)
peldsp display_model	display the parameters of a Model	
peldsp select_list	display a selected list of Diffusion Lists	Working with Diffusion Lists (command line)
peldsp display_list	display the parameters of a Diffusion List	

Subcommand	Use to...	Additional information
<code>peldsp select_user</code>	display a selected list of Users	See the <i>Security Guide</i> > Working with users (command line) for further information.
<code>peldsp display_user</code>	display the parameters of Users	
<code>peldsp select_profile</code>	display a selected list of Profiles	See the <i>Security Guide</i> > Working with Profiles (command line) for further information.
<code>peldsp display_profile</code>	display the parameters of Profiles	
<code>peldsp select_trans</code>	display a selected list of Transfer Requests	Working with Transfers (command line)
<code>peldsp display_trans</code>	display the parameters of a Transfer Request	
<code>peldsp status_trans</code>	display a selected list of Transfer Request with a subset of information for each selected Request	
<code>peldsp select_cgate</code>	display a selected list of CGates	Working with CGates (command line)
<code>peldsp display_cgate</code>	display the parameters of CGates	
<code>peldsp select_cgategroup</code>	display a selected list of CGateGroups	Working with CGateGroups (command line)
<code>peldsp display_cgategroup</code>	display the parameters of CGateGroups	
<code>peldsp select_connection</code>	display a selected list of Connections	Monitoring Connections (command line)
<code>peldsp display_connection</code>	display the parameters of a Connection	
<code>peldsp status_connection</code>	display the initial and dynamic states of a Connection	

Subcommand	Use to...	Additional information
<code>peldsp select_connected_user</code>	display a selected list of Connected Users	
<code>peldsp display_connected_user</code>	display the parameters of a Connected User	Monitoring Connected Users (command line)
<code>peldsp status_connected_user</code>	display the initial and dynamic states of a Connected User	
<code>peldsp select_decisionrule</code>	display a list of Decision Rules	Working with Decision Rules (command line)
<code>peldsp display_decisionrule</code>	display Decision Rule parameters	
<code>peldsp select_ruletable</code>	display a list of Rule Tables	Working with Rule Tables (command line)
<code>peldsp display_ruletable</code>	display Rule Table parameters	
<code>peldsp status_swoutputsession</code>	display the status of a SWIFTNet output session	SWIFTNet queues and output channels

pelinst

Purpose

Use the `pelinst` command to install or remove a Windows service.

Syntax

```
pelinst [parameters]
```

To install a Windows service:

```
pelinst <service name> <base directory> <account> <password>  
<description>
```

To remove a Windows service:

```
pelinst <service name> REMOVE
```

Subcommands

The **pelinst** command has no subcommands.

pelexport

Purpose

Use the **pelexport** command to export a configuration `.dat` file, to be reimported later using the **pelimport** command. This is relevant in the context of a migration, from 32-bit AIX to 64-bit AIX for example.

Syntax

pelexport [parameters]

To export a .dat file

```
pelimport -infile <nameOfFileToExport> -io <prefixOfConfigurationFile>
```

The default value of `<prefixOfConfigurationFile>` is `xfer`.

Example: this imports the exported file `exportedxferbfile.dat` into `xferb.dat`:

```
pelimport -infile xferbaix32.dat -io xferb
```

Subcommands

The **pelexport** command has no subcommands.

pelimport

Purpose

Use the **pelimport** command to import a configuration `.dat` into a configuration, from a file previously exported with the **pelexport** command. This is relevant in the context of a migration, from 32-bit AIX to 64-bit AIX for example.

Syntax

pelimport [parameters]

To import a .dat file

```
pelimport -infile <nameOfFileToImport> -io  
<prefixOfConfigurationFile>
```

The default value of <prefixOfConfigurationFile> is `xfer`.

Example: this imports the exported file `exportedxferbfile.dat` into `xferb.dat`:

```
pelimport -infile exportedxferbfile.dat -io xferb
```

Subcommands

The `pelimport` command has no subcommands.

pelmig

Purpose

Use the [pelmig](#) command to upgrade the Mailbox file between different versions of Gateway, for example, when upgrading from the previous version of Gateway to the current version.

Syntax

```
pelmig [parameters]
```

Subcommands

The `pelmig` command has no subcommands.

pelmon

Purpose

Use the `pelmon` command to:

- Check Mailbox integrity
- Restart Gateway
- Initialize the Mailbox
- Recover the Mailbox (if [Mirror option](#) is activated)
- Change the encryption passphrase file
- Repair the Virtual File Directory
- Reset the status prompted by the [gatestatus](#) command

Syntax

pelmon [**subcommand**] [**parameters**]

Subcommands

This table lists the subcommands of the **pelmon** command.

Subcommand	Use to...	Additional information
pelmon restart	restart Gateway	Managing the Mailbox: pelmon restart
pelmon reset	reset the status prompted by the gatestatus command to the default value	Managing the Mailbox: pelmon reset
pelmon init_base	initialize Gateway Mailbox	Managing the Mailbox: pelmon init_base
pelmon check_base	check Gateway Mailbox integrity	Managing the Mailbox: pelmon check_base
pelmon recover_base	recover Gateway Mailbox	Managing the Mailbox: pelmon recover_base
pelmon rebuild_vfd	repair the Virtual File Directory	Managing the Mailbox: pelmon rebuild_vfd
pelmon change_pass_phrase	change the encryption passphrase file	Managing the Mailbox: pelmon change_pass_phrase

Deprecated subcommands

The following **pelmon** subcommands are now deprecated. You should not use these commands with this version of Gateway.

Subcommand	Previously used to...	Command replaced by...
<code>pelmon start</code>	start Gateway	the gatestart command
<code>pelmon stop</code>	stop Gateway	the gatestop command
<code>pelmon status</code>	test Gateway status	the gatestatus command

pelpur

Purpose

Use the [pelpur](#) command to purge the Mailbox of all Transfer Requests that match the specified criteria. The purged Transfers are stored in an archive file (unless you set the optional parameter `-no_arc`).

You can also use the `pelpur` command to purge the transfer entries that match the specified criteria from the HA database used by SWIFTNet High Availability. To purge a transfer, the SWIFTNet High Availability feature has to be activated and connectivity with the HA database must be functional.

Syntax

`pelpur` [subcommand] [parameters]

`pelpur` [parameters]

Subcommands

Subcommand	Use to...	Additional information
<code>pelpur xferPurge</code>	Purge the mailbox of all Transfer Requests that match the specified criteria. This subcommand has the same effect as only running <code>pelpur</code> [parameters].	Works with transfers (command line) – table with <code>pelpur xferPurge</code>
<code>pelpur clean_swha_db</code>	Purge the transfers from the HA database (used by SWIFTNet High Availability) that match the specified criteria.	Works with transfers (command line) – table with <code>pelpur clean_swha_db</code>

Note `pelpur [parameters]` is similar to `pelpur xferPurge [parameters]`. Both formats can be used.

peltrans

Purpose

Use the [peltrans](#) command to submit a Transfer Request, a message Request, or a Diffusion List Request.

The request is a Diffusion List Request if the `dest_alias` parameter is a Diffusion List.

Syntax

peltrans [parameters]

Subcommands

The `peltrans` command has no subcommands.

secadm

Purpose

Use the `secadm` command to manage Gateway security elements.

Syntax

secadm [subcommand] [parameters]

Subcommands

This table lists the subcommands of the `secadm` command.

Subcommand	Use to...	Additional information
<code>secadm import_cert</code>	import a certificate from a file into the local database	
<code>secadm export_cert</code>	export a certificate to a remote database	See the <i>Security Guide</i> > Managing certificates (command line) for further information.
<code>secadm update_cert</code>	modify a certificate	
<code>secadm delete_cert</code>	delete a certificate from the database	
<code>secadm import_key</code>	import a key into the local database	See the <i>Security Guide</i> > Managing keys (command line) for further information.
<code>secadm update_key</code>	modify a key	
<code>secadm delete_key</code>	delete a key from the database	
<code>secadm import_sprof</code>	import a TLS Profile from a file into the local database	See the <i>Security Guide</i> > Managing TLS Security profiles (command line) for further information.
<code>secadm delete_sprof</code>	delete a TLS Security Profile from the database	
<code>secadm import_sshprof</code>	import an SSH Security Profile from a text file into the local database	See the <i>Security Guide</i> > Managing SSH Security profiles (command line) for further information.
<code>secadm delete_sshprof</code>	delete an SSH Security Profile from the database	
<code>secadm create_netprof</code>	create a Network Profile	See the <i>Security Guide</i> > Managing Network Profiles (command line) for further information.
<code>secadm update_netprof</code>	modify a Network Profile	
<code>secadm delete_netprof</code>	delete a Network Profile	

secdsp

Purpose

Use the **secdsp** command to select security elements and to view their parameters.

Syntax

secdsp [subcommand] [parameters]

Subcommands

This table lists the subcommands of the **secdsp** command.

Subcommand	Use to...	Additional information
secdsp display_cert	display security certificate attributes	
secdsp select_cert	display a selected list of certificates	See the <i>Security Guide</i> > Managing certificates (command line) for further information.
secdsp status_cert	display the state of a certificate	
secdsp display_sprof	display security profile attributes	
secdsp select_sprof	display a selected list of security profile attributes	See the <i>Security Guide</i> > Managing TLS Security Profiles (command line) for further information.
secdsp status_sprof	display the state of a security profile	

Subcommand	Use to...	Additional information
<code>secdsp display_sshprof</code>	display SSH profile attributes	
<code>secdsp select_sshprof</code>	display a selected list of SSH profile attributes	See the Security Guide > Managing SSH Security Profiles (command line) for further information.
<code>secdsp status_sshprof</code>	display the status if an SSH profile	
<code>secdsp display_netprof</code>	display Network Profile attributes	
<code>secdsp select_netprof</code>	display a selected set of Network Profile attributes	See the Security Guide > Managing Network Profiles (command line) for further information.
<code>secdsp status_netprof</code>	display the status of a Network Profile	
<code>secdsp display_key</code>	display security key attributes	
<code>secdsp select_key</code>	display a selected list of security key attributes	See the Security Guide > Managing keys (command line) for further information.
<code>secdsp status_key</code>	display the status of a security key	

vfdadm

Purpose

Use the `vfdadm` command to manage the Virtual File Directory.

Syntax

vfdadm [subcommand] [parameters]

Subcommands

This table lists the subcommands of the **vfdadm** command.

Subcommand	Use to...	Additional information
vfdadm create_dir	Create a new Virtual File Directory and set its attributes	
vfdadm update_dir	Update the attributes (including the optional mount point attributes) of an existing Virtual File Directory	
vfdadm move_dir	- Move a VFD directory (or mounted directory), from its current location in the VFD to a new location - Rename a VFD directory - Move and rename a VFD directory	Working with VFDs (command line)
vfdadm delete_dir	Delete a Virtual File Directory	

vfd dsp

Purpose

Use the **vfd dsp** command to select and display Virtual File Directory attributes.

Syntax

vfd dsp [subcommand] [parameters]

Subcommands

This table lists the subcommands of the **vfd dsp** command.

Subcommand	Use to...	Additional information
<code>vfddsp display_dir</code>	display the details, or attributes, of a VFD logical directory	
<code>vfddsp display_file</code>	display the details of a VFD logical file identified by its VFD parent directory absolute path (or its alias) and its name	Working with VFDs (command line)
<code>vfddsp list_dir</code>	list the contents of a specified Virtual File Directory	

Axway Gateway: User Interfaces

Online commands in Perl

GIK Perl overview

Axway Gateway provides two GIK Perl Command module extensions for Perl Version 5.24.0. These modules enable you to run Gateway command lines from a Perl script:

- **GIKCMDPL**: module for *local* command calls
- **GIKCCMDPL**: module for *remote* command calls

Syntax

Use the following command-line syntax:

```
GIKCMDPL::cmd('command line')
```

or

```
GIKCCMDPL::cmd('command line')
```

Using Perl modules

The Perl modules use the `cmd(...)` command. To execute a command, perform the following steps (detailed below):

1. Invoke the GIKCMDPL or GIKCCMDPL module.
2. Open the session.
3. Execute the command.
4. Close the session.

The Perl modules comprise the following routines.

Routine	Use
<code>Int openSession(char *, char *, char *) ;</code>	To open a session
<code>Int closeSession() ;</code>	To close a session
<code>Int cmd(char *) ;</code>	To run a command line
<code>Void printErrMsg() ;</code>	To display the error output
<code>Int getNumVar(char *)</code>	To retrieve numeric symbolic variables
<code>Char *getStringVar(char *) ;</code>	To retrieve alphanumeric symbolic variables
<code>Char **getResult() ;</code>	To retrieve the results array

Invoking the Perl module

To invoke the GIKCMDPL or GIKCCMDPL module, both a Perl environment and the module must be installed on the same platform as Axway Gateway.

Use either of the following call syntaxes in a Perl script:

```
#importing the GIKCMDPL module
```

```
use GIKCMDPL ;
```

or

```
#importing the GIKCCMDPL module
```

```
use GIKCCMDPL ;
```

Opening and closing a session

To initialize the Perl environment, call the **openSession** routine. If security is enabled, you must define the **openSession** command input parameters.

The **openSession** routine requires three parameters, including the user name and password. The first parameter is reserved for future use.

```
openSession('future use', 'user', 'password')
```

If security is enabled:

```
$rc = openSession(' ', 'user', 'password') ;
```

If security is disabled:

```
$rc = openSession(' ', ' ', ' ') ;
```

Before exiting the Perl script, call the **closeSession** routine to close the session and release memory space.

Note: You do not need to open a session for the **pelldconf** (*deprecated*) or **pelmon** commands.

Executing commands

The GIKCMDPL module is based on *local* online commands.

The GIKCCMDPL module is based on *remote* online commands.

The following table lists the commands available for each module.

Command	Available for:	
	GIKCMDPL	GIKCCMDPL
peladm	Yes	Yes
peldsp	Yes	Yes
pelbase	Yes	Yes
pelctl	Yes	Yes
pelpur	Yes	Yes
peltrans	Yes	Yes
pelmon	Yes	No
pelmig	No	No
pelldconf (<i>deprecated</i>)	Yes	Yes
vfdadm	Yes	Yes
vfd dsp	Yes	Yes
secadm	Yes	Yes
sec dsp	Yes	Yes

Syntax

The call syntax of the **cmd** function is:

```
Int cmd('command line')
```

The command line passed as an argument is parsed and processed. The function returns 0 if the command line executed successfully. Otherwise, the return code is negative.

If the return code is negative, use the void **printErrMsg()** function to display the corresponding error message.

peltrans commands

The **peltrans** commands return the Transfer identifier if processing was successful, or a negative value if processing failed.

peldsp commands

The **peldsp** commands include the **-st** option, which is used to set internal symbolic variables. For example:

```
$rc = GIKCMDPL::cmd('peldsp select_site -st');
```

If you use a **select** sub-command with the **-st** option, you can then use the **getResult()** function to retrieve the symbolic variables. This function returns an array containing the information requested during the command.

For example:

```
#results array display

@result=GIKCMDPL::getResult();

for $i ( 0 .. $#result ) {

print "\t $result[$i], \n";

}
```

The following example shows a **display** sub-command that uses the **-st** option.

```
$rc = GIKCMDPL::cmd('peldsp display_site -a LOOPPHSE -st');
```

If you use a **display** sub-command with the **-st** option, you can access the symbolic variables via the **getStringVar('var')** and **getNumVar('var')** functions, as described in the following section.

Accessing internal symbolic variables

Use the command **peldsp display_xxx -st** to create symbolic variables. All internal variables conform to the following syntax:

```
peldsp display_object ... -st
```

This command makes the internal variables available:

```
object.attribute
```

For example: **site.alias**.

There are two types of internal variable: numeric and alphanumeric. To access both types of variable, the call functions are as follows:

```
getNumVar('variable name') - for numeric variables
```

```
getStringVar('variable name') - for alphanumeric variables
```

If errors occur, use the **printErrMsg()** function. For example, if a numeric value returns a negative value or if a variable is not set for an alphanumeric variable, and so on.

```
#retrieving internal variables
```

```
GIKCMDPL::cmd(" peldsp display_site -a LOOPPHSE -st ");
```

```
$alias = GIKCMDPL::getStringVar(" site.alias ");
```

Axway Gateway: Configuration

Configuration

[Overview](#)

[Platform support](#)

[Network clients](#)

Overview

After installing a new version of Gateway, you need to configure it.

Although default parameters are always provided, you should verify and set the parameters for each component depending on your operational requirements.

The Gateway configuration menu enables you to make modifications to configuration parameters while Gateway is running.

Many of the modifications are taken into account immediately without having to shut down Gateway. This feature improves Gateway availability by reducing the need for shutdowns. However, after changing certain configuration parameters, you need to shut down Gateway for them to be taken into account. In this case a warning message is displayed.

Platform support

Linux and Windows support the following Gateway features:

- PEL
- PeSIT HS D and E
- OFTP (Odette)
- FTP
- SFTP
- HTTP
- SMTP
- POP3
- TCP/IP
- RAS *
- SNA LU 6.2
- TCP Server

- Axway MFT Navigator (formerly IUI) Server
- SSL/TLS
- EDIINT AS1
- EDIINT AS2
- EDIINT AS3
- RosettaNet
- SWIFTNet
- JMS
- X.400 (X.420)

* Windows platforms only

For specific platform versions, refer to the *Axway Supported Platforms* document.

Network clients

This version of Gateway supports two types of network client:

- *TCP/IP clients* that communicate via TCP/IP protocol
- *Axway MFT Navigator clients* that deploy Java over Internet (or intranet)

Configure the required server for your type of network client(s):

- [Navigator server](#): for TCP/IP clients
- [Axway MFT Navigator server](#): for Axway MFT Navigator clients

[Configuration: Procedure](#): How to access the configuration menu and configure Gateway.

[Configuration: Gateway parameters](#): The **Gateway** parameters of the configuration menu.

[Configuration: Customizing Trading Partner parameters](#): Advanced configuration parameters for certain trading file management situations.

[Configuration: Connectivity parameters](#): The **Connectivity** parameters of the configuration menu.

[Configuration: Secure Relay Router Agent parameters](#)

[Configuration: Secure Relay advanced options](#)

[Configuring protocols](#)

[Configuring connectors](#): The **Axway Connectivity** parameters of the configuration menu.

Axway Gateway: Configuration

Configuration: procedure

About configuration

You configure Gateway using the configuration menu of the GUI.

Some changes are taken into account immediately. For other changes, you need to stop and then restart Gateway.

Configuration modifications are tracked in the standard Gateway log.

Configuring Gateway

1. In the left pane of the GUI main window, right-click the Gateway server that you want to configure and select **Configure** from the context menu.

The *Configuration* window is displayed.

2. In the left pane of the *Configuration* window, select the item you want to configure.
Click  to expand the nodes.

The parameters are displayed in the right pane.

3. Fill-in or modify the required configuration information.

Refer to:

- [Configuration: Gateway parameters](#)
- [Configuration: Connectivity parameters](#)
- [Configuring protocols](#)
- [Configuring connectors](#)

4. For **Connectivity** parameters only:

After completing the parameter fields, click **Activate** to enable the protocol or connector in Gateway.

If you no longer want to use a protocol or connector, click **Deactivate** to disable it.

5. Click **OK** to save the changes.

The *Configuration* window closes.

Alternatively, click **Apply** to make these changes and continue configuring Gateway.

6. If Gateway displays a warning message, stop and then restart Gateway.
(This depends on the parameters you modified.)

At any time during configuration:

- Click **Help** for contextual help information.
- Click **Cancel** to abandon all changes that you have entered since you clicked **Apply**.

Configuring Gateway parameters

The **Gateway** parameters of the configuration menu condition the Gateway supervision services. For example: product and Local Site information, database options, user access control and log levels.

[Gateway product and local Site information](#)

[Advanced parameters](#)

[Trace levels](#)

[Log levels](#)

[Route transfer states](#)

[Gateway database information](#)

[Automatic archiving of Mailbox records, log and statistics files](#)

[Database protection options](#)

[User access control](#)

[Trace file archiving parameters](#)

Gateway parameters

Use the following parameters to set up Gateway product and Local Site information.

Parameter	Description
Local site alias	Enter the default Local Site alias. Maximum: 20 alphanumeric characters with no spaces. You can use the same value as the Local site name if desired.
Local site name	Enter the default Local Site name. Maximum: 20 alphanumeric characters with no spaces. You can use the same value as the Local site alias if desired.

Parameter	Description
License Key 1 - 4	The license keys that control your rights to use Gateway. A license key is linked to the hostname of your machine, or to the Key label below. You can enter up to four license keys. Only one key is mandatory. The key is validated every time Gateway is started. If the key is invalid, a diagnostic message is logged and the activation process aborted.
Key label	<i>Optional</i> The key label that controls your rights to use Gateway. A key label is not linked to the hostname of your machine. If you have a Key label, enter it in this field.

Advanced parameters and Trace levels

Normally, you do not need to modify any of the Advanced parameters.

You may need to use Advanced parameters if, for example, you need to [Customize Trading Partner parameters](#) for certain trading file management situations.

To modify Advanced parameters or Trace levels:

1. Click **Advanced options....**
2. In the *Parameter* column, select the required item.
3. In the *Value* column, type the new value.
4. Click OK to save the new settings.

Examples of Advanced parameters

Parameter	Description
<code>cancel_transfer_on_max_site_retry_reached</code>	Cancels the transfer when the maximum number of connection retries to a given Remote physical Site has been reached. Set value to 1 to activate this parameter, set to 0 to deactivate this parameter.

Parameter	Description
<code>job_max</code>	Specifies the maximum number of simultaneous activated scripts (for routed commands). This parameter is only taken into account by Gateway if the <code>started_command_user_control</code> parameter is set to yes. Use this parameter in conjunction with the <code>started_command_user_control</code> parameter.
<code>gc_debugger_cmd</code>	Configuration of the debugger command to launch after the 4th missed heartbeat for one vital process, to enable root cause analysis of non-responding processes. If the <code>gc_debugger_cmd</code> parameter is not set, no debugger is launched. There are four place-holders that can be used inside the definition of the debug command: • <code><PID></code> - it will be replaced with the process ID of the faulty process • <code><PWD></code> - the current working directory • <code><PNAME></code> - the name of the faulty process as it is known by Gatecontroller • <code><DATETIME></code> - the current timestamp For example, on Linux, the debugger command can be: <code>[gatecontroller]gc_debugger_cmd="strace -t -p <PID> -o <PWD>/<PNAME>_<DATETIME>.straceout"</code> Note: the debugger command can be used to launch any command and should be used with care, as the command will be ran with Gateway's user privileges
<code>gc_proc_max_retries</code>	Specifies the maximum number of restarts for certain non-vital Gateway processes that are controlled by the GateController if these processes stop abnormally. For example, <code>ipelapid</code> (the Gateway GUI daemon) or <code>xfbapid</code>.
<code>gc_stop_mon_maxretries</code>	Specifies the maximum number of retries to send a stop request to the supervisor process. You may need to modify this value if for example there were many requests to the supervisor just before the stop request. In this case the supervisor takes longer to handle the stop request as it first has to handle all other requests received earlier. If the supervisor has not handled the stop request within the number of retries defined by this parameter, GateController stops Gateway in a more abrupt manner.

Parameter	Description
<code>[monitor] ack_rcmax</code>	Specifies the maximum number of retries Gateway will attempt for an outgoing RECEIPT or EERP ack transfer. For manually initiated ACK transfers, if the <code>rc_max</code> parameter is specified for the transfer the maximum number of retries is established by the value of the <code>rc_max</code> parameter specified in the <code>peltrans</code> command. Example: <pre>peltrans -type RECEIPT -rtx 123 -rc_max 3</pre> If no value is set for the <code>[monitor] ack_rcmax</code> parameter, and <code>rc_max</code> is not specified in the <code>peltrans</code> command, the default fallback value is 5 (not configurable).
<code>gc_gw_max_loop_restart</code>	Specifies the maximum number of GateController restarts allowed. Use this parameter in conjunction with the <code>gc_gw_max_loop_restart_period</code> parameter. Example: If the GateController restarts every X seconds and you do not want more than Y restarts in X*Y seconds, you can modify the values of these parameters as follows: <pre>gc_gw_max_loop_restart_period = X gc_gw_max_loop_restart = Y</pre> The GateController will then be deactivated after the Yth following restart.
<code>gc_gw_max_loop_restart_period</code>	Specifies the period between two GateController restarts. Use this parameter in conjunction with the <code>gc_gw_max_loop_restart</code> parameter. Refer to the example provided in the <code>gc_gw_max_loop_restart</code> parameter section.
<code>started_command_user_control</code>	Set this parameter to yes if you want to control the maximum number of simultaneous activated scripts (for routed commands). Default = no. Use this parameter in conjunction with the <code>job_max</code> parameter that specifies the maximum number of simultaneous activated scripts.
<code>stop_site_on_max_retry_reached</code>	Deactivates the Remote physical Site when the maximum number of connection retries to this Site has been reached. Set value to 1 to activate this parameter, set to 0 to deactivate this parameter.

Parameter	Description
<code>ftp_vfd_fifo</code>	This parameter determines the FIFO behavior for the RETR, STOR, MDTM and SIZE FTP commands when they are used in combination with file name duplication (if the VFD allows it). Possible values: 0 = use the current LIFO behavior using the file modification time 1 = use the old Interpel FIFO behavior using the file modification time You must restart Gateway after modifying this parameter.
<code>vsite_age_max</code>	When using PassPort PM, Gateway saves the results of queries in specific Site objects called <i>Virtual Sites</i>. These Virtual Sites save PassPort from being queried all the time by Gateway. Virtual Sites have a tunable lifetime set by the <code>vsite_age_max</code> parameter. Setting a long lifetime reduces the number of queries to PassPort, resulting in shorter processing times. However, during that time any change made in the related PassPort entry is not refreshed. On the other hand, having a short <code>vsite_age_max</code> value allows quicker updates of PassPort modifications within Gateway, but with poorer performance. Possible values: 0 - 65 000 seconds. Default = 300.
<code>cancel_transfer_on_connection_refusal</code>	<i>PeSIT only</i> Cancel transfers on protocol connection phase failure. Set this parameter to 1 to activate this parameter (cancel transfers on connection refusal). Set this parameter to 0 to deactivate this parameter (do not cancel transfers on connection refusal).
<code>sid_limit</code>	<i>HTTP only</i> Use this parameter to limit the number of sessions a user can have on the server. Each time a user logs in, the value of this parameter is compared with the number of sessions registered for that user. If the number of registered sessions is lower than the value of this parameter, the user is allowed to login. Otherwise the login is refused. This parameter is only taken into account if you have set the HTTP server identification Method to <i>Cookie session id</i> or <i>CGI session id</i> in the configuration menu.

Parameter	Description
<code>select_transfer_by_pathname</code>	Use this parameter to enable resuming interrupted transfers when working with monitored RFD (real file directory). This parameter only affects protocols that support resuming interrupted file transfers, like FTP. Possible values: • 0 (default) = resuming interrupted transfers disabled • 1 = allow resuming interrupted transfers
<code>select_with_regex</code>	Use this parameter to enable regex searching with select commands. Possible values: • 0 (default - means no) • 1 regex characters: • * match zero to any number of characters (any printable character) • ? match exactly one character (any printable character) • [marks the start of a character group •] marks the end of a character group • – inside a character group, defines a range of characters. example: <code>a [b-d] c</code> will match <code>abc</code>, <code>acc</code> and <code>adc</code>. • ^ or ! used inside a character group, negates the entire group of characters. Example: <code>a [!abc] c</code> will not match <code>aac</code>, <code>abc</code> or <code>acc</code>, but will match <code>adc</code>, <code>aec</code>, <code>afc</code>, <code>agc</code>, etc., • \ escape character. Search for special characters. Example: <code>*</code> if you want to find a * (star) in the text. The search is case sensitive.

Examples of Trace levels

The higher the specified trace level, the more information is dumped to the trace file.

Parameter	Description
<code>x420.trace_level</code>	Trace level for X.420. Possible values: 0 – 4 (default = 1).
<code>ft_sftp.trace</code>	Trace level for SFTP generic code. Possible values: 0 – 4 (default = 0).
<code>ft_sftp.trace.tcp</code>	Trace level for low-level TCP code related to SFTP. Possible values: 0 – 4 (default = 0).
<code>ft_sftp.trace.ssh</code>	Trace level for SSH code related to SFTP. Possible values: 0 – 4 (default = 0).

Dynamic traces in Gateway

You can now activate the traces of Gateway processes without restarting the product. Trace settings now appear as dynamic parameters in the unified configuration and changing their value is done either:

- using the `peluconf` utility
Set `-s xsr log_level N`, where $0 \leq N \leq 3$. If `N` is numerical but other than 0, 1, 2 or 3, an error message is displayed.
- in Gateway Navigator (**Log levels > Secure Relay**)

The only difference is for Gatecontroller, whose trace level needs to be set using the new `pelmon` subcommand `traceset`. For Gatecontroller, the updates made with the command `pelmon trace_set` are only active until restart. There is no link between trace set in Navigator and trace set from the command line. The Gatecontroller trace parameter set in Navigator requires restart.

The list of trace level parameters is available in the advanced parameter configuration Gateway Navigator or using the `pelmon get` command, filtering by trace.

Some processes (`ipelsock`) may take some time to be applied requiring up to 120s to acknowledge the configuration change.

To verify that the trace level was set, look in the `log.properties` file in the `xsr` directory (`install_dir/run_time/xsr`). The line should appear as: `log4j.rootLogger=ERROR, XSR_ROLLINGFILE`.

Limitation

For APIs used by Gateway to interoperate with other products (Sentinel API, Integrator API, X.400, X.420, Messaging), the dynamic trace level is not supported.

Log levels

You can set the level of detail for log files for various components independently in Gateway.

To set the log level:

1. Click the **Log levels...** button.
2. Set the log level for the required component:
 - CGate
 - Minimum: errors only (For example: "No CGate matching the connection")
 - Short: name of selected CGate only
 - Full: name of CGate and its details (settings)
 - SecureRelay
 - Minimum: error messages (around 90), warnings (6 in total), 2 informational messages ("SecureRelay router agent is now available again" and initial listen point establishment)
 - Short: Dynamic listening ports for FTP data channel
 - Full: Incoming connections (with source and destination address)
 - FTP control session
 - Minimum & Short: control connection init/disconnect/error, start/end sending file, user authentication (220 welcome message, user/pass -- gtw as server only)
 - Full: (all from Minimum/Short), extra server commands
 - FTP data session
 - Minimum, Short & Full: data connection init/disconnect/error
 - AS3 control session (same as FTP control sessions)
 - AS3 data session (same as FTP data sessions)
 - MGet/MPut
 - Minimum: errors only
 - Short: info on whether it was an MPut or MGet, and the transfer ID for the master transfer
 - Full: details of which directory was listed and which files were requested or sent
 - GateController (for process monitoring)
 - Minimum & Short: gcontroller errors only
 - Full: details on each Gateway process heartbeat and gcontroller listening port
 - AS2 (Same as HTTP)
 - Security server (for SSL/TLS and SSH)
 - TSD (TCP Server Daemon)
 - Minimum & Short: TSD server listen ports for all active protocols
 - Full: (all from Minimum/Short), incoming connections for each active protocol

- Trade (EDIINT Trading partner)
 - Minimum: errors only
 - Short: (all from Minimum), originator/destination information
 - Full: (all from Short), encode/decode start/stop
- SFTP
 - Minimum, Short & Full: control connection init/disconnect/error, start/end sending file
- SWIFTNet
 - Minimum: transfer start/end, SWIFT sites started/stopped, SWIFT processes start/stop
 - Short: (all from Minimum), output channels and queues acquisition/release
 - Full: (all from Short), system recovery
- JMS
 - Minimum: errors, transfer stated/ended/failed
 - Short: (all from Minimum), warnings: queue read/write went down
 - Full: (all from Short), queue read/write opened/closed
- HTTP
 - Minimum: errors only, connection init/disconnect
 - Short: (all from Minimum), transfer start/end, HTTP primitive summary related to transfers
Medium - (all from Short), HTTP primitive summary for all pages
 - Full: (all from Medium), HTTP primitive summary for all files (including images on Gateway web page)
- RosettaNet HTTP (Same as HTTP)
- X.400 (X.400 support will be discontinued)

For each component, you can choose from the following options:

- **Minimum**
- **Short**
- **Medium** (only for AS2, HTTP, RosettaNet HTTP and X.400)
- **Long** (only for X.400)
- **Full**

X.400 log settings: You can set log levels for the various elements of the X.400 sub-system. These log messages do not appear in the Gateway log file. They are written to separate log files. Set the **Log size** (in Bytes), **Log full action** and **Log level** for each of the:

- MTA
- MTC
- MS
- UA X.420

where:

Log size represents the maximum size of the log entries specified in bytes.

In **Log level**, select what kind of information should be logged. The higher the log level is, the more information is logged:

- 0: Log Errors
- 1: Log level 0 + transport events
- 2: Log level 0-1 + transport PDUs
- 3: Log level 0-2 + internal messages
- 4: Log level 0-3 + configuration data

3. Click **OK** to confirm the new log level settings.

For more information about log files in Gateway, refer to [Log files](#).

Route transfer states

The Route transfer states configuration menu allows you to select the transfer states to trigger the routing mechanism and the Transfer Change State Decision Rule Table scanning in Gateway. The default values select the most important and common transitions and should be enough for most cases.

Note: Do not change the default values unless Axway Support or any specific Gateway documentation advises you to do this.

1. Click the **Route transfer states...** button.
2. Select the states to send to the router:

- Frozen (F)
- Delayed (D)
- To begin (B)
- To restart (R)
- Processing (P)
- Processing (V) (resp. side)
- Suspended (S)
- Interrupted (I)
- Cancelled (C)
- Ended (E)
- Created (T)
- To sign (G)
- Acknowledged (A)
- To ack (W)
- nack by USER (U)
- nack by XFB (X)

- Ended routing (1)
 - Canceled routing (2)
 - To pack (3)
 - Packing (4)
 - To unpack (5)
 - Unpacking (6)
3. Click **OK**.

Gateway > Database

Use the following parameters to set up Gateway database information.

Parameter	Description
Default directory for received files	Enter the default directory used to store received files. The directory that you specify here must exist. You can set this value in several Gateway objects. Gateway takes the value from object definitions in the following order of precedence: • Application • Remote Site • Local Site • Default directory defined here Click the browse button to locate and select the directory.
Mirror option	Gateway mirroring is a transparent software mirroring service that requires no specific hardware. When activated, it duplicates the Gateway Mailbox on a backup copy so that if the main Mailbox is damaged for any reason, it can be repaired from the mirroring Mailbox. Select one of the following options: • No mirroring: Mirroring is deactivated • Stop on error: Gateway stops if an error occurs during mirroring • Disable on error: Mirroring is disabled if an error occurs during mirroring • Ignore error: Mirroring continues and errors are ignored Refer to Managing the Mailbox (command line).

Parameter	Description
Mirror DB directory	Enter the directory path name where the backup copy of the Mailbox is stored. You should store the backup copy on a different disk drive from that used by the main Mailbox. The main Mailbox is located in the directory <Gateway installation directory>\run_time\data\. Click the browse button to locate and select the directory.
File IO ordering	Use this option to enable flushing in case of system crash. Select one of the following values: • Cache: No IO write ordering or flushing is performed (default). This option renders maximum performance • Order: The product forces flushing of write operations in order to preserve metadata consistencies in case of a system crash • Flush: The product forces flushing of write operations after each change in the data files
Purge transferred files after routing	Check this option if you want Gateway to automatically delete files during the Mailbox purge which were deposited to the Gateway and then routed to another machine.
Statistics option	Check this option to record statistics on completed transfers.
File name display option	<i>FTP and SFTP only</i> Set the default way in which VFD file names are displayed. Select one of the following values: • Gateway specific = including sXferIdent (default) • Standard = file name only Refer to VFD file name display option.

Parameter	Description
File name duplication policy	<i>FTP, HTTP and SFTP only</i> <i>This option is only available if you select Standard in File name display option.</i> Set the default way in which duplicate file names are handled. Select one of the following values: • Duplicate = duplicate names allowed (default) • Forbid = duplicate names not allowed • Override = if a duplicate file name is encountered, the new file replaces the old file with the same name Refer to VFD file name duplication policy.

Gateway > Database > Archive

Use the following parameters to set up automatic archiving of Mailbox records, log and statistics files on a periodic basis.

Parameter	Description
Mailbox cleanup:	
Log and statistics archive:	
Active	Select this option to enable automatic archiving of the log file and statistics file. Then enter one of the following: • Start date and Period, or • Threshold size If you specify both the Start date/Period and the Threshold size, the Start date/Period is ignored and the Threshold size is used. Refer to Automating maintenance operations.

Parameter	Description
Start date	Start date and time in the format: MM:DD hh:mm:ss, where: • MM = month (01 - 12) • DD = day in the month (01 - 31) • hh = hour in the day (00 - 23) • mm = minutes in the hour (00 - 59) • ss = seconds in the minute (00 - 59)
Period	The period in the format: MM:DD hh:mm:ss where: • MM = number of months • DD = number of days • hh = number of hours • mm = number of minutes • ss = number of seconds
Threshold size	Threshold size (in KB)

Gateway > Database > Protection

You can configure database protection to encrypt Gateway database files using a 128-bit AES key. Database protection guarantees the confidentiality of security elements.

If activated, all Network Profile objects, Security Profile objects, Private keys and certificates (plus keys) are stored in separate encrypted files. This database protection option also applies to administration objects (Sites, Applications, and so on) and Transfer Request objects.

You can enable the payload integrity check to assure that payload haven't changed between the moment is has been received by Gateway and further actions: routing to Integrator or routing to a third party. If activated, Gateway generates the signature for the incoming transfer's payload, using Gateway private key and validates the signature for the routed transfers. The signature for the routed transfers can be generated either by Gateway or by Integrator. The protocols for which the signature is computed in Gateway, for incoming transfers are: SWIFTNet, PeSIT and JMS.

Use the following parameters to set up database protection options.

Parameter	Description
Database protection option	To encrypt the Gateway administration files and/or the Mailbox, select one of the following values: • None: No database file encryption • Database files only: Encrypt administration files only • Database files and transfer mailbox: Encrypt administration files and the Mailbox When you use TLS and/or SSH with the Security server, you must set the encryption option to <i>Database files only</i> or <i>Database files and transfer mailbox</i>. This prevents eavesdropping and keeps critical information, such as RSA or DSA private keys, unreadable and unusable by unauthorized people. Important: After setting the required encryption level, you must perform the following steps: 1. Reload the profile to set the environment variables. 2. Stop Gateway. 3. Encrypt (or unencrypt) the database files using the pelmon change pass phrase command. 4. Restart Gateway. If you do not perform these steps, the encryption level is not changed.
Password derived key file	Enter the full path and file name where the derived key for encrypting your password is stored. Gateway uses this file (and the encrypted password file) to recover the password used to encrypt and protect your administration database files and the Mailbox. Click the browse button to locate and select the file.
Password encrypted file	Enter the full path and file name where the encrypted password is stored. Gateway uses this file (and the derived key file) to recover the password used to encrypt and protect your administration database files and the Mailbox. Click the browse button to locate and select the file.
Encrypt temporary received files	Select this option to encrypt received files.

Parameter	Description
Enable payload integrity check	Enable the payload signature generation for the incoming transfers and payload signature validation for the routed transfers.
RSA private key	Path to Gateway private key in PEM format
Password dk file	[Optional] path to the private key's password derived key file. The file must be generated with the <code>pelencpass</code> utility
Password enc file	[Optional] path to the private key's password file. The file must be generated with the <code>pelencpass</code> utility

Gateway > Connection control

Use the following parameters to set up user access control.

Parameter	Description
Administration control:	

Parameter	Description
User access control	This parameter controls user access to Gateway. Select one of the following values: • None: No access control. User credentials in the login request are ignored. • By object type: Users must provide their authentication (user ID and password) in their login request. Access control is based on object type. A user with authorized access to a type of object has the same access to all instances of that object type. Validation is carried out by Gateway or by customizable user exit routines. See Security guide > Access control using exits (requires login). • By object instance: Users must supply their authentication (user ID and password) in their login request. Access control is based on individual object instances. Validation can only be carried out by customizable user exit routines. • By PassPort AM: Access control is managed centrally for the Axway Platform platform by PassPort AM. See the Security Guide > Passport AM See Security Guide > Access Control Management for further information (requires login).
Default user name	<i>Optional</i> Enter the default user name. The name you enter here is assigned to files that Gateway receives.
Maximum number of simultaneous users	Enter the maximum number of concurrent users allowed in Gateway (1 - 200). When the upper threshold is reached, all subsequent login requests are denied.
Protocol control:	

Parameter	Description
Connection control method	Select one of the following connection control methods: • Monitor: Applies no identification via CGate or exit (implements identification via static Site object) • CGate: Applies default identification via CGates • Exit: Executes protocol connection exit • CGate & Exit: Applies CGate identification and then executes protocol connection exit • Exit & CGate: Executes exit and then applies CGate identification. Use this option to force the use of a particular CGate. • Exit & SGate: Executes exit and then applies SGate identification. Use this option to force the use of a particular SGate.
Set option per protocol...	Click this button to display the <i>Gateway Connection Control Method</i> window. Then, for each protocol you use, select one of the following connection control methods: • Use global: Applies the identification method that is specified in the global configuration parameter Connection control method. • Monitor: * • CGate: * • Exit: * • CGate & Exit: * • Exit & CGate: * • Exit & SGate: * * Refer to the descriptions for Connection control method (above).

Parameter	Description
Disable password pattern matching	<i>For CGate connection control methods</i> Select this option if, during user login, you want to disable the interpretation of wildcard characters in the login password (as defined in the CGate). In this case, during login, Gateway analyzes the password for plain text characters only. The field content is compared "as is" with the password. The only exception is when the password field contains a single *. Then it will match a null password as well as a single *. Do not select this option if you want the password field to contain any wildcard characters (as well as other printable characters). Wildcard characters: • * = matches any character, as many occurrences as possible • ? = matches any character, only once, at the specified position in the pattern • [] = specifies a set of allowed characters contained between the brackets. The user-specified password must match any one of the characters in this set, at the specified position in the pattern.
Hashed password	<i>Only if you have selected Disable password pattern matching.</i> Select this option if, for improved security, you want to store the user password in hashed format.
CGate blocking	Select this option to enable connection blocking at the CGate level. If you use this option, check the values of the Maximum login attempts before blocking and Blocking duration parameters.
Maximum login attempts before blocking	Enter the maximum number of login attempts before temporarily blocking the connection at CGate level.
Blocking duration	Enter the blocking duration (in seconds) before allowing a new connection attempt at CGate level.
Maximum number of initiated connections	Enter the maximum number of simultaneous initiated connections allowed. 1 - 999 (default: 100).

Parameter	Description
Maximum number of responded connections	Enter the maximum number of simultaneous responded connections allowed. 1 - 999 (default: 100).
User inactivity timeout (minutes)	Enter the user inactivity timeout for users connected in Gateway, representing the number of minutes after which the user is timed out and disconnected from Gateway. If set to 0 , the user inactivity timer will be infinite and the user will never be automatically logged out. The value can also be set using a line command: <pre>peluconf set -s monitor user_idle_timeout 10</pre> 0 – 65000 (default: 15).

Trace file archiving parameters

The following table provides details of the trace file archiving (collecting) parameters accessible through the peluconf command. You can find these parameters (or add them) in the collector configuration section.

Parameter	Description	Default value
max_files_to_move	Specifies the maximum number of files to be moved per iteration during a trace files collect operation. This should be fraction of the max_files_to_collect parameter.	10
max_files_to_collect	Specifies the maximum number of trace files to be considered for a collection per iteration during preparation of the archive operation.	100
collect_interval	Specifies the interval (in minutes) between two successive trace file collector runs.	10
collect_delay	Specifies the interval (in minutes) from the Gateway start until the trace collector is first activated.	0
archive_path	Specifies the directory where the archived traces are to be moved.	%p_home_dir%\run_time\tmp\archive

Configuring trading Partner parameters

This topic lists and describes the configuration parameters that you can use to customize Gateway behavior when managing certain trading file management situations.

For the procedure to follow, refer to [Configuration: Gateway parameters - Advanced parameters and Trace levels](#).

Parameter	Description
<code>trade_mic_generation_method</code>	Defines the EDIINT MIC calculation method when sending files: • 0: MIC is always calculated (default) • 1: MIC is only calculated if receipt is requested signed
<code>trade_allow_remote_duplicate</code>	<i>Only available for messages received for remote partners (To:).</i> Defines whether or not to allow duplicate messages: • 0: reject duplicate messages (default) • 1: skip search for duplicate messages
<code>trade_remove_duplicated_files</code>	Defines how to manage files containing messages that were already received: • 0: keep file • 1: remove file (default)
<code>trade_unknown_receipt_ended</code>	Defines how to handle the receipt of an unknown transfer: • 0: receipt canceled (default) • 1: receipt ended
<code>trade_recipient_check</code>	Defines whether to check the "Final-Recipient" field of the acknowledgement against the originator's name of the acknowledged transfer: • 0: ignore Final-Recipient field • 1: check Final-Recipient (default)

Configuration: connectivity parameters

[Activating/deactivating protocols and connectors](#)

[Navigator server](#)

[Secure Relay](#)

Activating/deactivating protocols and connectors

To enable a protocol or connector in Gateway, click **Activate**.

If you no longer want to use a protocol or connector, click **Deactivate** to disable it.

Configure Passport with SecureRelay

Starting from Gateway version 6.16.0, the Secure Relay Router Agent no longer connects to PassPort, so the settings for the PKI server (used when using Secure Relay with termination) are no longer set on the Router Agent, but rather on the Master Agent.

Use the following parameters to set up the Navigator server (GUI server) for the support of remote network clients over TCP/IP

- On Router Agent: set the PKI server type to `Generic`
- On Gateway: fill the following fields using the `peluconf` command:

```
[xsr]pki_server_type='None'  
[xsr]pki_server_host='<NO VALUE SET>'  
[xsr]pki_server_port='7101'  
[xsr]pki_server_use_tls='yes'  
[xsr]pki_server_login_user='<NO VALUE SET>'  
[xsr]pki_password_dkfile='%p_xsr_dir%\pkidkfile.dat'  
[xsr]pki_password_saltfile='%p_xsr_dir%\pkisaltfile.dat'  
[xsr]pki_password_datafile='%p_xsr_dir%\kipwd.dat'  
[xsr]pki_ca_cert_file='%p_home_dir%\extras\PassPort\PassportCA.crt'
```

These are the default settings; when using PassPort, the type is `PassPort`.

The password files should be generated using the `pelencpass` tool provided with Gateway.

Connectivity > Navigator server

1. Use the following parameters to set up the Navigator server (GUI server) for the support of remote network clients over TCP/IP.

Parameter	Description
Listening port (mandatory)	Enter the port number assigned to the Gateway Navigator server. The port number must be in decimal format and unique on the operating system. Default = 6390
Network interface (mandatory)	Enter the local IP address in dotted format, or the local host name as defined in the HOST file.
<i>The following fields are optional. You can activate the Navigator server without using the TLS option.</i>	
Use TLS	Select this option to use TLS/SSL and then complete the following parameters. TLS versions: the preferred TLS version is TLS 1.2, but Gateway Server also supports TLS 1.1 and TLS 1.0. There is no configurable option for choosing the TLS version. If an older client (Navigator or XIB) connects to Gateway using lower TLS versions, the server will fallback to the client's version (provided the connection is allowed - please see the "API version check" parameter in this table).
X509 certificate	Enter the name of file that contains the X.509 certificate to be used by the Navigator server. Typically, this is a file in PKCS 12 format.
Password dk file	Enter the name of the file containing the derived key for decrypting the password to the X.509 certificate. This file should be one of the outputs of the <code>pelencpass</code> command.
Password enc file	Enter the name of the file containing the encrypted password to the X.509 certificate. This file should be one of the outputs of the <code>pelencpass</code> command.
Server max	Maximum number of client connections. This includes connections from Gateway Navigator, but also from Integrator (which is using Gateway's C API to connect)

Parameter	Description
API version check	Checks whether client & server API versions match. • If <i>not active</i>, the server will allow the connection, regardless of the API versions on server and client. • If <i>active</i> and the client version is <i>newer</i> than the server one, connection is rejected. • If <i>active</i>, and if the client version is <i>older</i> than the server version, and if <code>User access control</code> is other than <i>None</i>: the user's right to connect depends on per-user settings. See the Security Guide for further information (requires login). • If <i>active</i>, and if the client version is <i>older</i> than the server version, and if <code>User access control</code> is equal to <i>None</i>: the connection is allowed automatically. (There can be no API version check without some form of User access control). Note: when using Passport AM for authentication, to allow older API version clients to connect (i.e. XIB): add <code>LOGIN_XIB</code> privilege to the user's role.

2. Click **Activate** to enable the Navigator server.

Connectivity > Secure Relay

1. Use the following parameters to set certificate, address and port information for the Master Agent and Router Agent components of Secure Relay.

Parameter	Description
Outgoing calls bypass Secure Relay for address mask	<i>For Gateway outgoing connections to local network entities.</i> Enter an alphanumeric mask number enabling local network calls to bypass Secure Relay. You can enter up to 32 masks, separated by a blank space or a semi-colon. You can use the following wild-card symbols: • * = integer between 0 and 255 • ? = integer between 0 and 9 Examples of legitimate mask numbers are: • 178.14.*.* • 199.0.2??.*

Parameter	Description
Master Agent	
Certificate file	Enter the name of the security certificate for the Secure Relay Master Agent. Click the browse button to help you locate a certificate on your machine. Note: No sample certificate is provided with Gateway. By default this parameter is empty and you need to explicitly provide a value.
Certificate password	Enter the password for access to the Secure Relay Master Agent certificate. Note: For additional security you can encrypt the password stored in the Gateway configuration file. Refer to the <i>Security guide</i> > Working with Secure Relay for further information (requires login).
CA Certificate file	Enter the name of the CA certificate file. Click the browse button to help you locate a certificate file on your machine. Note: No sample CA certificate is provided with Gateway. By default this parameter is empty and you need to explicitly provide a value.
Router Agents This pane lists the Router Agents that are currently configured. You can use up to eight Router Agents. To update the list of Router Agents, use the Modify button:	
Modify	Double-click a Router Agent in the list to modify the Router Agent parameters. Click Modify to configure a new Router Agent.
Activate	Select a Router Agent in the list. Click Activate to activate or deactivate this Router Agent.

- Click **Advanced...** to access [Secure Relay advanced options](#).
- Click **Activate** to enable Secure Relay Master Agent processes.

[Configuration: Procedure](#)

[Configuring protocols](#)

[Configuring connectors](#)

Configuration of Secure Relay

To preserve your data from hackers, the information previously available in the User guide, relative to the configuration of Secure Relay has now been moved to the *Gateway Security guide* > [Configuring Secure Relay](#) (requires a login to read online).

Axway Gateway: Configuration

About configuring protocols

Supported protocols and network interfaces

Gateway supports the following file transfer protocols and associated network interfaces:

File transfer protocol	Network interface
PEL	TCP/IP, SNA, PAD
PeSIT HS (D and E)	TCP/IP, SNA
OFTP (Odette)	TCP/IP
FTP	TCP/IP
HTTP	TCP/IP
SMTP	TCP/IP
POP3	TCP/IP
SFTP	TCP/IP
AS1	TCP/IP
AS2	TCP/IP
AS3	TCP/IP
RosettaNet	TCP/IP
X.400 (X.420)	TCP/IP

Network interfaces

A file transfer protocol stack needs at least one network transport interface. You can, however, associate a file transfer protocol with more than one network interface.

If you are using a UNIX platform, refer to [Network configuration for UNIX platforms](#).

To configure Gateway to communicate using a specific protocol and network interface, refer to [Configuring protocols](#).

Axway Gateway: Configuration

About configuring protocols

Supported protocols and network interfaces

Gateway supports the following file transfer protocols and associated network interfaces:

File transfer protocol	Network interface
PEL	TCP/IP, SNA, PAD
PeSIT HS (D and E)	TCP/IP, SNA
OFTP (Odette)	TCP/IP
FTP	TCP/IP
HTTP	TCP/IP
SMTP	TCP/IP
POP3	TCP/IP
SFTP	TCP/IP
AS1	TCP/IP
AS2	TCP/IP
AS3	TCP/IP
RosettaNet	TCP/IP
X.400 (X.420)	TCP/IP

Network interfaces

A file transfer protocol stack needs at least one network transport interface. You can, however, associate a file transfer protocol with more than one network interface.

If you are using a UNIX platform, refer to [Network configuration for UNIX platforms](#).

To configure Gateway to communicate using a specific protocol and network interface, refer to [Configuring protocols](#).

Axway Gateway: Configuration

Configuring protocols

This topic describes the **Internet protocols** and **Legacy protocols** parameters of the configuration menu. These are the parameters that you can set to enable Gateway to communicate using different protocols.

[Activating/deactivating protocols](#)

[Internet protocols](#)

[Legacy protocols](#)

[SecureRelay access policy](#)

Activating/deactivating protocols

In the configuration menu:

- To enable a protocol in Gateway, click **Activate**.
- If you no longer want to use a protocol, click **Deactivate** to disable it.

Connectivity > Internet protocols

To configure a file transfer protocol, select the desired protocol in the left (tree view) pane, then complete the parameter fields in the right pane.

Connectivity > Internet protocols > FTP

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use in server mode for this protocol. See [TCP/IP parameters](#).
3. Click **Options...** to access [FTP options](#).
4. (optional) Specify the **Listen port range in client active mode** in the **Min** and **Max** fields.

5. Click **Activate** to enable this protocol.
6. Click **OK** or **Apply**.

Connectivity > Internet protocols > HTTP

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use for this protocol. See [TCP/IP parameters](#).
3. Click **Options...** to access [HTTP options](#).
4. Click **Activate** to enable this protocol.
5. Click **OK** or **Apply**.

Connectivity > Internet protocols > SFTP

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use for this protocol. See [TCP/IP parameters](#).
3. Click **Advanced...** to access [TCP advanced options](#).
4. Click **Activate** to enable this protocol.
5. Click **OK** or **Apply**.

Connectivity > Internet protocols > Mail > SMTP

1. Use the following parameters to configure Gateway for sending electronic mail to a mail server using SMTP.

Parameter	Description
Add Gateway parameters to subject field	Select this option to add specific Gateway parameters to the <i>Subject</i> field as the default behavior.
Gateway subject delimiter	Specify the delimiter character to use for Gateway parameters in the subject field. For example: <code>//XFB</code>
Default SMTP site	Select the default SMTP Site to use.
Use S/MIME	Select this option to apply S/MIME security functions to SMTP transactions.
Use SecureRelay	Select this option to use Secure Relay with this protocol.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **Activate** to enable this protocol.
4. Click **OK** or **Apply**.

Connectivity > Internet protocols > Mail > POP3

1. Use the following parameters to configure Gateway for the retrieval of electronic mail from a mail server using POP3.

Parameter	Description
Gateway subject delimiter	Specify the delimiter character to use for Gateway parameters in the subject field. For example: //XFB
Use S/MIME	Select this option to apply S/MIME security functions to POP3 transactions.
Use SecureRelay	Select this option to use Secure Relay with this protocol.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **Activate** to enable this protocol.
4. Click **OK** or **Apply**.

Connectivity > Internet protocols > EDIInt > AS1 > SMTP

1. Use the following parameters to configure Gateway for sending electronic mail to a mail server using EDIINT AS1 SMTP.

Parameter	Description
Use SecureRelay	Select this option to use Secure Relay with this protocol.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **Activate** to enable this protocol.
4. Click **OK** or **Apply**.

Connectivity > Internet protocols > EDIInt > AS1 > POP3

1. Use the following parameters to configure Gateway for the retrieval of electronic mail from a mail server using EDIINT AS1 POP3.

Parameter	Description
Use SecureRelay	Select this option to use Secure Relay with this protocol.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **Activate** to enable this protocol.
4. Click **OK** or **Apply**.

Connectivity > Internet protocols > EDInt > AS2

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use for this protocol. See [TCP/IP parameters](#).
3. Click **Options...** to access [HTTP options](#).
4. Click **Activate** to enable this protocol.
5. Click **OK** or **Apply**.

Connectivity > Internet protocols > EDInt > AS3

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use in server mode for this protocol. See [TCP/IP parameters](#).
3. Click **Options...** to access [FTP options](#).
4. (optional) Specify the **Listen port range in client active mode** in the **Min** and **Max** fields.
5. Click **Activate** to enable this protocol.
6. Click **OK** or **Apply**.

Connectivity > Internet protocols > RosettaNet > HTTP

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use for this protocol. See [TCP/IP parameters](#).
3. Click **Options...** to access [HTTP options](#).
4. Click **Activate** to enable this protocol.
5. Click **OK** or **Apply**.

Connectivity > Internet protocols > RosettaNet > MAIL > SMTP

1. Use the following parameters to configure Gateway for sending electronic mail to a mail server using RosettaNet SMTP.

Parameter	Description
Use SecureRelay	Select this option to use Secure Relay with this protocol.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **Activate** to enable this protocol.
4. Click **OK** or **Apply**.

Connectivity > Internet protocols > RosettaNet > MAIL > POP3

1. Use the following parameters to configure Gateway for the retrieval of electronic mail from a mail server using RosettaNet POP3.

Parameter	Description
Use SecureRelay	Select this option to use Secure Relay with this protocol.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **Activate** to enable this protocol.
4. Click **OK** or **Apply**.

Connectivity > Legacy protocols

To configure a file transfer protocol, select the desired protocol in the left (tree view) pane, then complete the parameter fields in the right pane.

Connectivity > Legacy protocols > PeSIT E > TCP

1. Select the required [SecureRelay access policy](#).
2. Specify the ports to use for this protocol. See [TCP/IP parameters](#).
3. Click **Advanced...** to access [TCP advanced options](#).
4. Click **Activate** to enable this protocol.
5. Click **OK** or **Apply**.

Connectivity > Legacy protocols > PeSIT E > SNA

Refer to [Configuring protocols: SNA LU 6.2](#).

Connectivity > Legacy protocols > PeSIT D > TCP

Refer to [Connectivity > Legacy protocols > PeSIT E > TCP](#).

Connectivity > Legacy protocols > PeSIT D > SNA

Refer to [Configuring protocols: SNA LU 6.2](#).

Connectivity > Legacy protocols > OFTP (Odette) > TCP

Refer to [Connectivity > Legacy protocols > PeSIT E > TCP](#).

Connectivity > Legacy protocols > PEL > TCP

Refer to [Connectivity > Legacy protocols > PeSIT E > TCP](#).

Connectivity > Legacy protocols > PEL > SNA

Refer to [Configuring protocols: SNA LU 6.2](#).

SecureRelay access policy

For each protocol that you use for exchanges with your Internet partners, specify the type of access that is allowed for connections to Gateway.

Parameter	Description
SecureRelay access policy	Select the required access policy: • SecureRelay is not used: only direct access (via the Gateway port specified in the <i>Gateway port</i> column). • Through SecureRelay only: access through Secure Relay (via the Gateway port specified in the <i>SecureRelay port</i> column). • Through SecureRelay and directly: both direct access and access through Secure Relay. If you select this mode you can specify a supplementary access port number for incoming connections to the Secure Relay Router Agent.

Note: Before changing the **SecureRelay access policy** for a given protocol, first delete any ports that were previously defined for that protocol.

[Configuration: Procedure](#)

[About Axway Gateway protocols](#)

IPv6 support

Gateway is able to communicate over TCP IPv6 with remote peers. Gateway objects involved in configuring the transport phase of a transfer now accept IPv6 addresses. Partner definition remains unchanged with the only modification being in the network address field: an operator can configure an IPv6 address or a hostname that resolves into an IPv6 address. Network profiles and cgate/sgates filtering rules allow operators to define rules using the IPv6 address format. The same pattern

matching rules as with IPv4 addresses can be used, with an asterisk "*" replacing an entire group and a question mark "?" replacing a single character. The GUI provides default matching rules for all addresses. (These default matching patterns also act as a guideline on how filters can be defined).

Configuration has been extended to accept IPv6 interfaces for Gateway services (protocol configuration) with or without Secure Relay. The operator can configure the interface on the Router agent side on which the listening ports are to be established. This means that when combined with Secure Relay, Gateway continues to function only on IPv4 addresses and uses IPv6 only when connecting to the outside world via Secure Relay (any combination is supported).

Axway Gateway: Configuration

Configuring TCP/IP parameters

This topic describes how to configure TCP ports for a file transfer protocol.

In this version of Gateway you enter port information in the *Port specification in server mode* pane.

Former syntax: If you have upgraded from a previous version, Gateway attempts to convert all previous TCP port configuration information. In certain cases, this is not possible. If the port configuration information cannot be converted, Gateway displays the port configuration information in the [Port specification using former syntax](#) field.

[About TCP ports](#)

[Configuring TCP ports](#)

[Port specification using former syntax](#)

About TCP ports

TCP port numbers must be in decimal format and unique on the operating system. The TCP port associates an incoming TCP/IP connection request with the appropriate file transfer Application.

Default values:

- 6310 for PEL
- 6320 for PeSIT HS D
- 6330 for PeSIT HS E
- 6360 for OFTP
- 6370 for FTP
- 6380 for HTTP
- 6321 for SFTP
- 6382 for AS2
- 6373 for AS3
- 6386 for RosettaNet HTTP

Additional information for TCP port configuration with FTP (and AS3)

When using FTP (and AS3), you can specify the ports that are used for control sessions. Optionally, you can separately define the dynamic data sessions in order to control the range of ports available for a session. To do this, you need to define a port configuration block that lists the available ports for a specific type of session.

A port configuration block comprises:

- For the server - a control session port followed by the range of data connection ports that are used in passive mode
- For a client - the range of data connection ports that are used in active mode

Configuring TCP ports

When you configure a file transfer protocol in the configuration menu, the *Port specification in server mode* pane displays the current TCP port configuration.

To change the TCP port configuration:

1. Either:
 - Add a new port specification by clicking **New**, or
 - Modify an existing port specification by selecting the required row and clicking **Modify**Gateway opens the *SAP parameters* window.
2. Specify the TCP port(s) in the [SAP parameters](#) window.
3. Click OK.

Using TCP ports below number 1024

Introduction

Although it is strongly discouraged, it is possible to configure a product to bind explicitly to local TCP port numbers below 1024. This allows you, for example, to replace TCP services shipped with the Operating System by the equivalent service provided by the product. A typical usage of this feature could be to replace the O/S FTP server by configuring the product FTP listen port to value 21.

When doing so, you must ensure that the corresponding O/S TCP service is disabled prior to starting the product.

TCP ports below 1024 (i.e. in the range 1-1023) are called privileged ports. Applications that use such port numbers are required to run under the root account (or to be allowed to switch to this account) for the O/S to accept socket systems services that set up the local end-points.

Allowing an application to run either permanently or temporarily as root can be achieved in two different ways, as follows:

- Start the application under the root account.
- Make root the owner of the executable files(s) and set the `suid` attribute bit for the file(s).

Recommended method

For the product to operate with privileged local TCP ports, we recommend using the "`suid bit`" method over the "start under root account" method. This ensures that the product runs by default under the user account switching to root only for socket system services that require such a privilege. These system services include:

- `bind()` : Binds local address + port number pair to a socket.
- `listen()` : Establishes the queue of incoming calls on a bound socket allowing connections from clients to be accepted.
- `connect()` : Connect to a remote server.

Operating system dependencies

Some UNIX systems may allow the `listen()` service to succeed under a normal user account when the local end-point is bound to a privileged port, some others may require the application to run under the root account in the same situation. This difference may exist for the `connect()` system service. Since such a difference does not appear to be clearly documented, the product performs these system services under the root account when the local end-point is applied to a privileged TCP port.

Another difference applies to dynamically linkable libraries. Some UNIX systems may require all DLLs linked by the `suid` executable to be seen from the default library path.

Configuration

The executable files that may bind sockets to privileged local TCP ports include:

- `p_tsd` : Generic TCP server daemon. This executable is started when the product is configured for dynamic TCP servers.
- `p_ftpsock` : FTP protocol file transfer program.
- `p_httpsock` : HTTP protocol file transfer program.

These files are located in the sub-directory `bin` of the product installation home directory.

Depending on the actual configuration of privileged ports and/or the use of dynamic servers or not, only part of those files may have to be set as `suid` from root. You may not, for example, set the `suid` bit for file transfer programs that are not configured for privileged local ports.

To allow a program to start as root, the following attribute changes are to be performed under the root account (example for file `bin/p_ftpsock`):

1. Set owner root for the file: `chown root bin/p_ftpsock`.
2. Set the `suid` bit on exec: `chmod +s bin/p_ftpsock`.

Depending on your UNIX O/S visibility requirements for dynamically linkable libraries, all such libraries needed for the executable must be set as visible giving the default library patch. This can be achieved using logical links as follows (example for the file `obj/libtftools.so`):

- Create a logical link to `libtftools.a` in `/usr/lib`:

```
ln -s `pwd`/obj/libtftools.a /usr/lib/libtftools.so
```

Sample configuration scripts.

A sample shell script that performs the required operations for all files that may have to bind sockets to privileged local ports is supplied. The pathname is `run_time/etc/socksuid.sh`.

This shell script can only succeed when run under the root account. It does not check the actual product configuration but performs all operations for both FTP and HTTP protocols to be allowed to bind to local privileged ports. This script is provided as an example. System administrators may check their content prior to running it and optionally comment-out any command that is not needed depending on the actual product configuration.

For example, if only FTP is configured for privileged local ports, all lines involving files for other protocols should be commented-out. To allow the installation of patches and service packs, the script has rollback support. This option requires the name of the user that owns the product installation (the file owner prior to running the script). In this situation, the script must also be run under the root account.

Under the hood.

This section describes how the product interacts with the different uids attached to a `suid` process when local privileged ports have been configured. These brief explanations are intended for security analyzing.

The program detects that it has been started from a non-root user account and `set-uid` to root by the condition of having the effective `user_id` equal to zero and the real `user_id` different from zero.

Since, by default, all the resources it may have to create must be owned by the `<real user-id>`, this program sets the `<effective user-id>` to the value of the `<real user-id>` as soon as possible. This happens immediately after the main entry point. It has the advantage of preventing security problems since the program then runs with a non-root `<effective user_id>` value.

Due to the product having been configured to use privileged TCP port values, it may be necessary to switch the `<effective user-id>` to root (i.e. zero). For the attachment of such a port to be made possible, the following algorithm is used:

LISTEN to a privileged port

1. Try to switch `<effective user-id>` to root (zero).
2. Create a socket (using the `socket()` system service).
3. Build the address argument to use on "bind" using the privileged port value.

4. Bind the socket and listen to it (using the `bind()` system service, then the `listen()` system service in this order).
5. Revert the `<effective user-id>` to the `<real user-id>` value.

CONNECT from a privileged port (local port):

1. Try to switch `<effective user-id>` to root (zero).
2. Create a socket (using the `socket()` system service).
3. Build the address argument to use on "bind" using the privileged port value.
4. Bind the socket and connect to a remote (using the `bind()` system service, then the `connect()` system service in this order).
5. Revert the `<effective user-id>` to the `<real user-id>` value.

Care has been taken for the program to ensure that (5) will be executed when (1) succeeds.

Technically, the program uses the `setreuid()` system service in a way that ensures that either the `<saved user-id>` or the `<effective user-id>` is set to `<root user-id>` at any time, for the `<effective user-id>` to remain changeable to `<root user-id>` when required (and switched back to `<real user-id value>` as soon as it is no longer required).

As a result, when `suid-root` starts, the program is basically running by default with `<effective user-id>` set to the value of the `<real user-id>`, and will switch temporarily the `<effective user-id>` to root only when it is required. Such behavior minimizes security issues and ensures that all other resources are created and/or accessed under the `<real user-id>` account (i.e. `<effective user-id>` set to `<real user-id>`).

Port specification using former syntax

This section describes the TCP port configuration displayed in the **Port specification using former syntax** field.

This field is only displayed if:

- You configured TCP ports in a previous version of Gateway
- You have upgraded to this version of Gateway
- Gateway cannot convert the TCP port configuration information

TCP port configuration using former syntax

If a protocol listens on more than one port, the port numbers are specified as follows:

- single port: `nnnn`
- list of ports: `mmmm nnnn pppp`
- range of ports: `mmmm:nnnn`

A list of ports is separated by commas:

Addr1/Port1,Addr2/Port2, Addr3/Port3

where **Addr(n)** designates an IP address and **Port(n)** designates a port specification. If you do not specify the address, the value of the IP address is taken.

For example:

9950

9900 9920 9930

20000,1.72.10.5/21000:21002

TCP port configuration with FTP using former syntax

The format and rules listed below define one, or several, port configuration blocks **Ps(Ps1:Ps2)** (**Pc1:Pc2**)...

where:

- Ps = port used for server control sessions
- Ps(n) = range of ports used for data connections in server passive mode
- Pc(n)= range of ports used for data connections in client active mode

The following conventions are used when specifying control session ports in FTP:

- A space is used between the port configuration blocks, for example between **Ps(Ps1:Ps2)** and **(Pc1:Pc2)**
- No space is used between the server port and the port range listed in parentheses, **Ps(Ps1:Ps2)**
- No space is used for a range of ports for a session in parentheses **(Pc1:Pc2)**
- Client ports, which are used in the active mode, are not attached to a control session port

*Examples using former syntax***Example 1: Classic configuration**

This example displays a simple and effective FTP port configuration. Note the spaces between block units.

6370(8000:8010) 6470(8000:8010) (8000:8010)

- 6370 = server listening port for FTP control sessions
- 6470 = secondary listening port, for example for an SSL secured session
- (8000:8010) = this port range is used for data connections in server passive mode attached to a session established on port 6370 or 6470, as well as data connections in client active mode

Example 2

This example displays an alternate use of port configuration blocks.

6370(8000:8010) 6470(8011:8020) (8021:8030)

- 6370 = server listening port for FTP control sessions
- (8000:8010) = port range for data connections in server passive mode, attached to a session established on port 6370
- 6470 = secondary listening port, for example for an SSL secured session
- (8011:8020) = secondary port range for data connections in server passive mode, attached to a session established on port 6470
- (8021:8030) = range of ports for data connections in client active mode

Example 3

It is important to note in this example that there is a space between the listening port and the data connection port range. This space separates the two elements into two block entities, so that the range is a separate block and not the range for port 6370.

6370 (8021:8030)

- 6370 = server listening port for FTP control sessions. Note that when in passive mode, the server listens on any port for data sessions
- (8021:8030) = range of ports for data connections in client active mode

[TCP advanced options](#)

[Configuring protocols](#)

Axway Gateway: Configuration

Configuring protocols: SAP parameters

This topic lists the parameters that you can modify in the *SAP parameters* window. Gateway opens this window if you click the **New** or **Modify** button when configuring a protocol.

The choices available in this window depend on the value chosen in **SecureRelay access policy** for this protocol.

1. Complete the fields of the *SAP parameters* window as follows:

Parameter	Description
General	
Gateway listen port	Enter the TCP port number on which Gateway listens.
Gateway	Enter the Gateway network interface.

Parameter	Description
SecureRelay listen port	Enter the TCP port number on which Secure Relay listens.
SecureRelay	Enter the SecureRelay network interface.
Master side listen port	<i>Optional</i> Specify the port you want to open on XSR Master Agent side.
Listen port range in server passive mode	<i>Optional</i> Specify the range of ports (Min to Max) for data connections in server passive mode attached to a session established on this Gateway listen port.
Bypass data port range	<i>Optional</i> If checked (<i>true</i>) then the setting for the <i>Listen port range in server passive mode</i> (the general setting, see above) will be overwritten by the <i>per-XSR-RA</i> Incall Data Port Range setting (the specific setting). If not checked (<i>false</i> , default value) the above setting will be used.
Advanced	
Disable IP filtering in Secure Relay	Select this option to specifically disable the IP filtering on the current Sap, on all Secure Relay instances. This option is available only when general IP filtering is enabled in Configuration: Secure Relay advanced options .
Transport security in Secure Relay	Select this option if you require TLS termination in the DMZ.
Security profile	Select the required security profile.
Protocol login in Secure Relay	<i>FTP, HTTP only</i> Select this option if you require Login termination in the DMZ (protocol login).

2. Click **OK** to confirm the SAP parameters.

[Configuration: Connectivity parameters](#)

[Configuring protocols](#)

Configuring FTP options

For FTP and EDIINT AS3 protocols

This topic lists the FTP parameters that you can modify in the *FTP options* window. Gateway opens this window if you click the **Options...** button when configuring the FTP or EDIINT AS3 protocol.

1. Complete the fields of the *FTP options* window as follows:

Parameter	Description
Quota window size	<i>FTP / SFTP only</i> Enter the time (in minutes) for which the sliding quota window is active. Default: 0 (all defenses are disabled, irrespective of the quota values defined).
First failed connection interval	Enter the time (in seconds) between the first failed connection attempt and the second. The time that you enter in this field is doubled for each subsequent connection attempt. Minimum: 0. Default: 2.
Maximum login attempts	Enter the maximum number of times a User can try to log in. Minimum: 0. Default: 3.
Send allocation request (client mode)	<i>Client mode only</i> Select this option to send an Allocation Request to the server. If the server refuses the request, the Transfer is rejected.
Force final EOF on receipt	Select this option to force final End-of-File on file reception.

2. Click **Advanced...** to access [TCP advanced options](#).
3. Click **OK** to confirm the *FTP options*.

[Configuring protocols](#)

Axway Gateway: Configuration

Configuring protocols: HTTP options

This topic lists the HTTP parameters that you can modify in the *HTTP options* window. Gateway opens this window if you click the **Options...** button when you configure an HTTP-type protocol.

1. Complete the fields of the *HTTP options* window:

Parameter	Description
Directory listing	
Default template file	<i>Not for RosettaNet HTTP or EDIINT/AS2</i> Enter the full path of the HTML page used as a template to display directory content.
Server identification	
Method	Select the identification method accepted by the HTTP server: • Web • CGI session id (seen note below) • CGI user password (seen note below) • Cookie session id • Partner-Dependant identification • Cookie user password Note The server identification methods <i>CGI session id</i> and <i>CGI user password</i> are insecure by design and should not be used. These identification methods are deprecated, and are due to be removed in a future version of the product.
Server ID	Gateway displays this field if you select <i>Web</i> in the Method field. Enter the name that the server sends when Web authentication is required.
User found	Gateway displays this field if you select any identification method other than <i>Web</i> in the Method field. Use this field to retrieve the content of the <i>User</i> field in a CGI.
Password found	Gateway displays this field if you select any identification method other than <i>Web</i> in the Method field. Use this field to retrieve the content of the <i>Password</i> field in a CGI.

Parameter	Description
Session found	Gateway displays this field if you select <i>Cookie session id</i> or <i>CGI session id</i> the Method field. (In the Method field above, see note about deprecated parameter CGI session id) Use this field to retrieve the content of the <i>sessionId</i> field in a CGI.
Advanced connection behavior	
Do not keep alive	Select this option to close the connection at the end of each request.
Max packet size	Enter the maximum number of bytes sent in an outgoing packet of data.
Max list loop	<i>Not for RosettaNet HTTP or EDIINT/AS2</i> Enter the maximum number of consecutive LIST entries to return to the client.
Max CGI size	Enter the maximum number of bytes from which the server tries to retrieve information (such as user, password and so on). Internal services do not accept requests that exceed this value.

2. Click **Advanced...** to access [TCP advanced options](#).

3. Click **OK** to confirm the *HTTP options*.

Note: You can limit the number of simultaneous open sessions for a user. Refer to the `sid_limit` parameter in the Gateway [advanced parameters](#).

[Configuring protocols](#)

[About HTTP](#)

[Using HTTP in server mode](#)

[Using HTTP in client mode](#)

Axway Gateway: Configuration

Configuring protocols: TCP advanced options

Gateway opens the *TCP advanced options* window if you click the **Advanced...** button when configuring any protocol that uses TCP/IP.

1. Modify the TCP parameters as follows:

Parameter	Description
Maximum number of connections in initiator mode	Maximum number of simultaneous initiated connections allowed for the selected protocol. 1 - 999 . Default: 100 .
Maximum number of connections in responder mode	Maximum number of simultaneous responded connections allowed for the selected protocol. 1 - 999 . Default: 100 .
Use SSL/TLS	<i>Not for SFTP</i> Select this option to secure TCP/IP connections with SSL/TLS protocols.
Dynamic client processes (outgoing connections)	
Active	Select this option if you require a pool of dynamic processes for outgoing (client) connections.
No. processes min	Minimum number of processes to maintain in the pool.
No. processes max	Maximum number of processes allowed in the pool.
Balance ratio	You can distribute outgoing connections between different processes. If you set the value to 0, all incoming connections are directed to the first process until it is full. Subsequent processes are then directed to the second process and so on. Percentage: 0 - 100.
Dynamic server processes (incoming connections) <i>not for all protocols</i>	
Active	Select this option if you require a pool of dynamic processes for incoming (server) connections.
No. processes min	Minimum number of processes to maintain in the pool.
No. processes max	Maximum number of processes allowed in the pool.
Balance ratio	You can distribute incoming connections between different processes. If you set the value to 0, all incoming connections are directed to the first process until it is full. Subsequent processes are then directed to the second process and so on. Percentage: 0 - 100.

2. Click **OK** to confirm the *TCP advanced options*.

[Configuring protocols](#)

Axway Gateway: Configuration

Configuring X.400 O/R address

The *O/R Address* window is displayed if you click **Edit...** in:

- The configuration menu
- Remote Site
- Transfer Request

Message originators and recipients within X.400 are identified by their O/R address. The O/R address consists of a number of mandatory and optional address attributes. These attributes are organized hierarchically.

Specify the O/R address by modifying the address attributes in the *X.400 O/R Address* window. Only use [valid characters](#) in the O/R address, as listed in this topic.

Note: Not all fields are used for all objects. In this case they are grayed out.

General tab

Field	Description
Country (C)	Identifies the Country.
ADMD (A)	Administration Domain Name. Identifies an ADMD relative to the country specified by the Country Name. An ADMD provides a public messaging service within a single country. Several ADMDs may exist within a Country.
PRMD (P)	Private Domain Name. Identifies a PRMD relative to the ADMD or the Country. A PRMD is a messaging service within an organization. The ADMD operator must uniquely assign the Name of the Private Domain if the PRMD is connected to an ADMD service.
Organization (O)	Identifies an organization relative to the PRMD.

Field	Description
Org. Unit 1 (OU1)	Organization Unit Name 1-4. Can be used up to four times to identify divisions/departments within the organization.
Org. Unit 2 (OU2)	The names appear in order of importance. Always start with Org. Unit 1. You cannot enter an Org. Unit 2 unless you enter an Org. Unit 1 first, and so on.
Org. Unit 3 (OU3)	
Org. Unit 4 (OU4)	
Common name (CN)	Identifies a user, an application, or a distribution list. Caution: Common name is not supported in MTA version 84. You may encounter problems if your MTA is routed via a version 84 MTA.

Personal tab

Field	Description
Surname (S)	The Surname is a mandatory part of the Personal Name.
Given name (G)	The Given Name is an optional part of the Personal Name.
Initials (I)	The Initials are an optional part of the Personal Name.
Generation Qual (Q)	The Generation qualifier is an optional part of the Personal Name.

Domain defined tab

Field	Description
Type (DDA)	The Domain-defined attributes are specific to a messaging domain and are used to pass non-standard information.
Value	These attributes are used in addition to those that specify the user/application.

Additional tab

Field	Description
Free form name (FFN)	Free text.
Telephone number (TPH)	Telephone number of Originator/Recipient.

Valid characters in O/R addresses

Character description	Character
Uppercase alphabetic characters	A, B, C, ... Z
Lowercase alphabetic characters	a, b, c, ... z
Digits	0, 1, 2, ... 9
Space	
Apostrophe	'
Left and right parentheses	()
Plus sign	+
Comma and period	, .
Slash	/
Dash	-
Colon	:
Equal sign	=
Question mark	?

[About X.400](#)

[X.400 connector](#)

[Configuration: Connectors](#)

[New Remote Site: X.400 tab](#)[New Transfer Request: X.420 tab](#)

Axway Gateway: Configuration

Configuring protocols: X.400 advanced settings

This topic lists the X.400 advanced parameters that you can modify in the *Advanced settings* windows. Gateway opens one of these windows if you click any of the *Advanced settings* buttons when configuring the X.420 protocol.

[Advanced server ports settings](#)[Advanced OSITP settings](#)[Advanced MTA settings](#)[Advanced MTC settings](#)[Advanced MS settings](#)[Advanced UA settings](#)

Advanced server ports settings

1. Complete the fields of the *Advanced server ports settings* window as follows:

Parameter	Description
Sys starter port	This port is used by the X.400 system starter. Choose any free port.
OSI TP port	This port is used by X.400 internal processes. Choose any free port.
RFC 1006 port	This port is used to listen for incoming external IP connections from peer MTAs. Choose any free port.
X.420 UA port	This internal port is used by the Gateway X.420 facility to communicate with the X.400 subsystem. Choose any free port.
X.435 UA port	This internal port is used by the Gateway X.435 facility to communicate with the X.400 subsystem. Choose any free port.
Host name	This value must match a local IP address. Typical value: 127.0.0.1 or localhost. Maximum: 64 alphanumeric characters.

2. Click **OK** to confirm the settings.

Advanced OSITP settings

1. Complete the fields of the *Advanced OSITP settings* window as follows:

Parameter	Description
Max transport connections	Maximum number of transport connections that the transport module accepts. The memory required for transport connections is: Max transport connections * Buffer size. Memory is reserved for each new transport connection at connection time. If memory cannot be reserved, the connection is not established. Default value 32.
Max network connections	Maximum number of network connections the transport module can handle. Default value 32.
Max connection response time (ms)	Maximum time (in milliseconds) that the transport module waits for a connection response service primitive from the transport user to a previous issued connect indication service primitive. Default value 500 000 ms.
TS1 time (ms)	Time (in milliseconds) to wait for an answer to a transport Connect Request. If an answer is not received within this time, the network connection is disconnected. Default value 1 200 000 ms.
Buffer size (B)	Memory size, in bytes, that each transport connection may use for TPDU (Transport Protocol Data Unit) buffers. Default value 24 000 bytes.
Max TPDU size (B)	Maximum TPDU size (in bytes) that is proposed on an outgoing connect request. Default value 2048 bytes.

2. Click **OK** to confirm the settings.

Advanced MTA settings

1. Complete the fields of the *Advanced MTA settings* window as follows:

Parameter	Description
Delivery period (s)	Defines how frequently the MTA process should check the UA in-queues. If an incoming message is not read by the receiving UA, the MTA removes the message from the UA delivery queue and sends a non-delivery report message to the sender, if requested.
Poll period (s)	Frequency (in seconds) that the MTA should poll its in-queue. Messages from UAs and the MTC process are put in the in-queue. The poll period is used only when the queue is empty. If the queue is not empty, all messages are processed before polling is restarted.
Transfer time (s)	Maximum time (in seconds) that it may take for a message to be transmitted between two MTAs. If the time is exceeded, the transmission is considered to have failed and the connection is released.
Reload configuration at startup	Determines whether the configuration should always be reloaded at start-up.

- Click **OK** to confirm the settings.

Advanced MTC settings

- Complete the fields of the *Advanced MTC settings* window as follows:

Parameter	Description
Max connect attempts	Defines how many times the MTC process should try to connect to an MTA before giving up. If the connection fails, a non-delivery message is sent to the UA that had this MTA as destination for messages. If the parameter is set to 0 or 1, only one attempt is made. Default value: 3.
Max attempt timer (s)	When a connection attempt fails, this value specifies the time (in seconds) to wait before attempting to connect again. Default value: 30.
Max in connections	The maximum number of simultaneous incoming connections that are accepted by the MTC. If this value is reached, any new connection is rejected with a reason code indicating temporary congestion. Default value: 6.
Max out connections	The maximum number of simultaneous outgoing connections that are established by the MTC. Default value: 2. Refer to the Note above, under Max in connections .

Parameter	Description
Max message queued	The maximum number of messages that can be queued for the local MTA routing. If this level is reached, a new incoming connection is rejected with a code indicating temporary congestion. Default value: 512.
RTS window size	Specifies the number of checkpoints that can be transmitted without receiving an acknowledgment from the remote MTA. Checkpoints are used to synchronize a sending MTA with a receiving MTA. This value must be larger than 0. Default value: 3.
RTS checkpoint size	Specifies the maximum amount of data that can be transmitted between two checkpoints. Valid values are within the range of 0 to 100 KB. 0 means that checkpointing is disabled. Default value: 1.
Keep delivery reports time (h)	This parameter instructs the MTC process not to discard delivery reports when the maximum number of connection attempts has been reached. Typically, when the MTC process fails to connect to a remote MTA, that is, does not make a connection after the configured number of attempts: • All messages, notifications, and reports are removed from the message out-queue • A non-delivery report is sent for all messages and UA notifications, and • All reports are discarded If this option is used, all reports are kept in the message out-queue, and the MTC process keeps trying to connect to the remote MTA once every hour. The value of the hours argument specifies the number of hours during which the MTC process attempts to re-send delivery reports. The value must be a whole number greater than zero.

Parameter	Description
Keep notifications time (h)	This parameter instructs the MTC process not to remove any X.420 notifications from the message out-queue when the maximum number of connection attempts has been reached. Typically, when the MTC process fails to connect to a remote MTA, that is, does not make the connection after the configured number of attempts: • All messages, notifications, and reports are removed from the message out-queue • A non-delivery report is sent for all messages and notifications, and • All reports are discarded If this option is used, all X.420 notifications sent from the local UAs are kept in the message out-queue, and the MTC process keeps trying to connect to the remote MTA once every hour. The value of the argument specifies the number of hours during which the MTC process attempts to re-send notifications. The value must be a whole number greater than zero.
Min transfer speed (byte/s)	This parameter is used to specify the lowest network transfer speed that can be expected. The process to calculate maximum transfer times uses this value. A message transfer is aborted if the maximum transfer time is exceeded. Note: This option should be used only when there is a problem with transfer timeouts and the value should never be increased. Default value: 200 bytes/sec.
Use RTS recovery	Use this parameter to determine whether recovery should be carried out on broken connections or not. Note that recovery is rarely used. Default value: No recovery.
Dialog mode	
Use	Specifies whether messages may be sent in both directions on the same connection between two MTAs. Select one of: • Monolog: one direction (default) • Two way alternate: both directions (rarely used)

2. Click **OK** to confirm the settings.

Advanced MS settings

1. Complete the fields of the *Advanced MS client settings* window as follows:

Parameter	Description
Max message size (B)	Sets the maximum size on messages retrieved from the MS. If a message with a size larger than this value is found in the MS it is deleted from the Message Store (MS)
Message fetch limit	Maximum number of messages fetched in one MS poll. This functionality can be used to flow-control at peak hours. When the size is 0, no messages are retrieved. This can be useful when you want to test the MS connection without retrieving any messages.
Min transfer speed (b/s)	Specifies the lowest network transfer speed that can be expected. This value is used to calculate maximum transfer times. A message transfer is aborted if the maximum transfer time is exceeded.
Transport class	Transport class proposed at the establishment of a connection.
Time to keep sending notifications (h)	This parameter instructs the MS client not to remove any X.420 notifications from the message out-queue when the maximum number of connection attempts has been reached. Normally, when the MS client fails to connect to a remote MS, that is, does not make the connection after the configured number of attempts: • All messages and notifications are removed from the message out-queue. • An internal non-delivery report is sent for all messages and notifications. If this parameter is used, all X.420 notifications sent from the local UAs are kept in the message out-queue, and the MS client keeps trying to connect to the remote MS once every hour. The value of this field specifies the number of hours during which the MS client attempts to re-send notifications.

2. Click **OK** to confirm the settings.

Advanced UA settings

1. Complete the fields of the *Advanced X.420 UA settings* window as follows:

Parameter	Description
Allow anonymous addressing	If you select this option, the UA accepts messages coming from a sender with an OR-address that does not match any of the X.420 remote users.

Parameter	Description
Notifications expected within (m)	The time (in minutes) within which notifications are expected, if notifications have been requested. If this time is exceeded for a transfer for which notifications are requested, the original message is marked NACKED. Default value: 1380 minutes.

- Click **OK** to confirm the settings.

[Configuring connectors](#)

[Configuring Gateway for X.400](#)

[About X.400](#)

Axway Gateway: Configuration

Configuring SNA LU 6.2 (on Windows)

For Windows server platforms only.

One of the network interfaces supported by Gateway is SNA LU 6.2. Gateway accesses SNA LU 6.2 via the CPI-C interface.

[Restrictions](#)

[Defining SNA Resources](#)

[Configuring Gateway for SNA LU 6.2](#)

[Configuring Remote Sites for SNA LU 6.2](#)

[CPI-C symbolic destination](#)

[CPI-C Transaction](#)

[Troubleshooting](#)

[Error messages](#)

Restrictions

The current version implements only *basic conversation* functions, and does not support the following functions:

- Commit/rollback
- PIP (Program Initialization Parameters)
- Conversational security
- Control verbs

If required, you must define the LU-LU security and conversational services via the SNA configuration program.

Defining SNA Resources

Before setting up Gateway for SNA LU 6.2, ensure that Microsoft SNA server is installed and SNA resources are correctly configured. In particular, the following SNA resources must be defined:

- Node
- Link Service
- Connection
- Remote LU
- Local LU
- Mode
- [CPI-C Transaction](#)
- [CPI-C symbolic destination](#)

For more information on SNA Server configuration, refer to the following documentation:

- Microsoft SNA Server Installation Guide
- Microsoft SNA Server Administration Guide
- Microsoft SNA Server Managing Microsoft SNA Server
- Microsoft SNA Server Reference

Configuring Gateway for SNA LU 6.2

1. In the Gateway configuration menu, select **Connectivity > Legacy protocols**.
2. Select the required protocol:
 - PeSIT E
 - PeSIT D
 - PEL
3. Select **SNA**.
4. Use the following parameters to set up the SNA interface for use with the selected protocol. The SNA parameters listed here are only those that Gateway references directly. Parameters required for the normal operation of an SNA Server are configured separately during the installation of the SNA Server.

Parameter	Description
Incoming sessions	Specify the maximum number of concurrent incoming SNA LU 6.2 sessions for the protocol being configured. This value must be greater than or equal to the number of SNA LU 6.2 sessions configured in your SNA base.

Outgoing sessions	Specify the maximum number of concurrent outgoing SNA LU 6.2 sessions for the protocol being configured. This value must be equal to or greater than the number of SNA LU 6.2 sessions configured in your SNA base.
Local LU name	Specify the local LU to associate with the file transfer protocol being configured. This LU name must also exist in your SNA base.
Transaction program	Specify the CPI-C transaction name for the file transfer protocol being configured. Suggested default values: • PELCPICR for PEL • PHSECPICR for PeSIT HS E • PHSDCPICR for PeSIT HS D Refer to CPI-C Transaction .

5. Click **Activate** to enable the SNA interface for use with this protocol.

Configuring Remote Sites for SNA LU 6.2

Refer to [Site objects: Address configuration for Remote Sites](#).

CPI-C symbolic destination

Among the [SNA resources](#) mentioned above, only *CPI-C symbolic destination* is directly referenced by Gateway. Specify the CPI-C symbolic destination name in the **Destination** field in the Gateway Remote Site definition. It identifies the SNA resources to use to connect to the Site.

Below is a sample definition of a CPI-C symbolic destination for Microsoft SNA server:

- **Name:** PELIMVS
- **Comment:** MVS
- **Partner TP Name:** PELCPICR
- **Partner LU Name:** PELLU62A
- **Conversation Security:** None
- **Mode Name:** LPELLU62

The **Partner LU name** corresponds to an LU defined on the remote platform (for example, the ACB parameter in VTAM APPL macro on IBM MVS).

The **Partner TP name** designates the remote transaction to invoke. It must be known on the remote platform.

The **Mode Name** determines the session parameters to use. The same name must also be defined on the remote platform.

The **Local LU** used for the transaction is that defined during Gateway configuration.

CPI-C Transaction

For a transfer in Initiator mode, the remote platform determines which TPN to use.

For a transfer in Responder mode, the TPN used must be one of the following:

- PELCPICR: for PEL
- PHDCPICR: for PeSIT HS D
- PHECPICR: for PeSIT HS E

These TPNs are automatically defined in the Windows registry during Gateway configuration. They are "autostarted non_queued". Before starting an LU 6.2 transfer, ensure that `tpstart.exe` (a Microsoft SNA Server utility program) is started.

For more information on `tpstart.exe`, refer to the *Microsoft SNA Server CPI-C Programmer's Guide*.

Sample entries installed by the Gateway configuration procedure

PELCPICR

```
\HKEY_LOCAL_MACHINE\system\CurrentControlSet\Services\SnaBase\Parameters\
TPs\PELCPICR\Parameters\
SNAServiceType:REG_DWORD:0x06
PathName:REG_SZ:c:\winnt35\system32\cmd.exe
Parameters:REG_SZ:/C d:\xfb\bin\PELCPICR.cmd d:\pelint
AcceptNames:REG_SZ:PELCPICR
TimeOut:REG_DWORD:0xffffffff
ConversationSecurity:REG_SZ:NO
!--kadov_tag{}}-->AlreadyVerified:REG_SZ:YES
```

PHDCPICR

```
\HKEY_LOCAL_MACHINE\system\CurrentControlSet\Services\SnaBase\Parameters\
TPs\PHDCPICR\Parameters\
SNAServiceType:REG_DWORD:0x06
PathName:REG_SZ:c:\winnt35\system32\cmd.exe
Parameters:REG_SZ:/C d:\xfb\bin\PHDCPICR.cmd d:\pelint
AcceptNames:REG_SZ:PHDCPICR
TimeOut:REG_DWORD:0xffffffff
```

```
ConversationSecurity:REG_SZ:NO
```

```
AlreadyVerified:REG_SZ:YES
```

PHECPICR

```
\HKEY_LOCAL_MACHINE\system\CurrentControlSet\Services\SnaBase\Parameters\
```

```
TPs\PHECPICR\Parameters\
```

```
SNAServiceType:REG_DWORD:0x06
```

```
PathName:REG_SZ:c:\winnt35\system32\cmd.exe
```

```
Parameters:REG_SZ:/C d:\xfb\bin\PHECPICR.cmd d:\pelint
```

```
AcceptNames:REG_SZ:PHECPICR
```

```
TimeOut:REG_DWORD:0xffffffff/p>
```

```
ConversationSecurity:REG_SZ:NO
```

```
AlreadyVerified:REG_SZ:YES
```

Troubleshooting

The following checklist may be helpful when SNA connection problems occur:

- Is MS SNA running? If not, start it now.
- Is `tpstart.exe` running? If not, start it now.
- Is the physical link to the remote platform active? If not, start it now. If link activation fails, consult MS SNA documentation and check the corresponding parameters.
- Can you activate an LU-LU session with the remote platform? If LU-LU activation fails, consult MS SNA documentation and check the corresponding parameters.
- If you can activate the LU-LU session but conversation allocation fails, check that the symbolic destination used is defined in SNA. The partner LU, logmode and transaction program as defined in the symbolic destination must also be defined on the remote platform.
- If conversation allocation is accepted but a timeout occurs, check that the remote Gateway is started.
- If problems persist, further information is also available in Windows Event Log and MS SNA event log or audit files.

Error messages

When a CPI-C function call fails, an error message is written in the Gateway LOG file as shown below:

function=cmxxxx, rc=nnn, errcod=mmm

where:

- **cmxxxx** is the CPI-C function concerned
- **nnn** is the return code of that function
- **mmm** is the diagnostic specific to the SNA implementation on your platform

The following list is taken from CMC.H, and describes the **nnn** return codes in question.

nnn	Description
0	CM_OK
1	CM_ALLOCATE_FAILURE_NO_RETRY
2	CM_ALLOCATE_FAILURE_RETRY
3	CM_CONVERSATION_TYPE_MISMATCH
5	CM_PIP_NOT_SPECIFIED_CORRECTLY
6	CM_SECURITY_NOT_VALID
7	CM_SYNC_LVL_NOT_SUPPORTED_LU
8	CM_SYNC_LVL_NOT_SUPPORTED_PGM
9	CM_TPN_NOT_RECOGNIZED
10	CM_TP_NOT_AVAILABLE_NO_RETRY
11	CM_TP_NOT_AVAILABLE_RETRY
17	CM_DEALLOCATED_ABEND
18	CM_DEALLOCATED_NORMAL
19	CM_PARAMETER_ERROR
20	CM_PRODUCT_SPECIFIC_ERROR
21	CM_PROGRAM_ERROR_NO_TRUNC
22	CM_PROGRAM_ERROR_PURGING
23	CM_PROGRAM_ERROR_TRUNC
24	CM_PROGRAM_PARAMETER_CHECK
25	CM_PROGRAM_STATE_CHECK
26	CM_RESOURCE_FAILURE_NO_RETRY
27	CM_RESOURCE_FAILURE_RETRY
28	CM_UNSUCCESSFUL
30	CM_DEALLOCATED_ABEND_SVC
31	CM_DEALLOCATED_ABEND_TIMER
32	CM_SVC_ERROR_NO_TRUNC
33	CM_SVC_ERROR_PURGING
34	CM_SVC_ERROR_TRUNC
100	CM_TAKE_BACKOUT
130	CM_DEALLOCATED_ABEND_BO
131	CM_DEALLOCATED_ABEND_SVC_BO

132	CM_DEALLOCATED_ABEND_TIMER_BO
113	CM_RESOURCE_FAIL_NO_RETRY_BO
134	CM_RESOURCE_FAILURE_RETRY_BO
135	CM_DEALLOCATED_NORMAL_BO

Axway Gateway: Configuration

Using RAS

For Windows platforms only.

This topic describes how to use Microsoft Remote Access Service Networking (RAS) with Gateway.

[Enabling RAS](#)

[Customizing RASPPP.ini](#)

[How Gateway makes a connection using RAS](#)

[Common errors and troubleshooting](#)

Enabling RAS

One of the network interfaces supported by Gateway is the dial-up connection. This facility is implemented using Microsoft Remote Access Service Networking (RAS) on Windows platforms.

To enable this facility in Gateway:

1. Install MS dial-up networking service.
2. Define remote PPP servers in MS dial-up networking phone book.
3. Configure Gateway file transfer protocols with TCP/IP networking.
4. [Customize the Gateway RASPPP .ini file.](#)
5. Configure Gateway Remote Sites to use RAS connections.
Refer to [Site objects: Address configuration for Remote Sites.](#)

Configuring RAS

For details on how to install RAS and add RAS phone book entries, refer to the Microsoft RAS documentation.

For each remote PPP server, define an RAS networking phone book entry. This entry contains parameters such as phone line number, modem and server settings.

Customizing RASPPP.ini

Gateway uses a text file named `RASPPP.INI` (in the usual Windows INI file format) to bind itself to the RAS service. In the `RASPPP.INI` file, each remote PPP server is defined as an INI file section.

During Gateway installation, a default `RASPPP.INI` is copied to the `<Gateway installation directory>\run_time\etc\` directory. Customize this file depending on your RAS configuration before you run Gateway.

Sample RASPPP.INI file

```
;
; See <Gateway installation directory>\help\srvcfggen.hlp
; for more information
;
;=====
;
;
; [*links*] section for internal usage
;
; Max= maximum simultaneous RAS connections (decimal
; value, default is 4)
; Share= share RAS link with other applications (boolean:
; 0=NO, 1=YES, default is 0)
; KeepAlive= keep connection active on end of transfer
; (boolean: 0=NO, 1=YES, default is 0)
; Force= close all connections on program exit (boolean:
; 0=NO, 1=YES, default is 0)
;
;=====
;
[*links*]

Max=4
Share=1
KeepAlive=0
Force=0
;=====
;
; User section Samples:
;
; [Destination]
; LinkEntry= RAS networking phone book entry name
; (mandatory)
; Phonenumber= phone number (optional if Linkentry is
; specified)
; Server= server where user credentials are to be
; validated (optional)
; USERNAME= logon user name (optional for NT, mandatory
; for W9.x)
```

```

; PASSWORD= login user password (optional for NT,
mandatory for W9.x)
;
;
=====
=====

[PARIS]
linkentry=PARIS
Phoneno=
Server=
USERNAME=DUPONT
PASSWORD=BONJOUR

[LONDON]
linkentry=
Phoneno=
Server=
USERNAME=SMITH
PASSWORD=HELLO

```

Each section is defined with the following parameters:

Parameter	Meaning
[xxxxxxx]	INI file section header. It represents an RAS destination for Gateway.
Linkentry	<i>Mandatory</i> RAS phone book entry to use.
Phoneno	<i>Optional</i> Phone number to dial. If you do not specify this parameter, the parameter defined in RAS phone book entry is used. For purposes of consistency, it is recommended that you leave this parameter blank and use the RAS phone book entry definition.
Username	<i>Mandatory on Windows 9.x, optional on Windows NT</i> Login user name for the remote host. If not specified, the current login credentials are used.
Password	<i>Mandatory on Windows 9.x, optional on Windows NT</i> Login user password for the remote host. If not specified, the current login credentials are used.
Server	<i>Windows NT only</i> <i>Optional</i> Computer where login credentials are validated. If not specified, the dialed-up PPP server is used.

Apart from the destination sections(s), `RASPPP.INI` also has a `[*links*]` section, where parameters governing the program behavior are defined.

```
[*links*]
```

```
Max=4
```

```
Share=0
```

```
KeepAlive=0
```

```
Force=0
```

where:

- **[*links*]** is a reserved system section
- **Max** = Maximum number of simultaneous RAS connections.
A decimal value up to 64. Default = 4.
- **Share** = Share RAS connections with other applications. A Boolean value:
 - **0 = NO** (default): If **Share=0** and the required port (modem) is in use by another application, the connection request is refused.
 - **1 = YES**: If **Share=1**, the connection is considered valid and the file transfer is carried out over this pre-established link. You should only select this option if the external application and Gateway share the same PPP server.
- **KeepAlive** = Keep the RAS connection open after file transfer. A Boolean value:
 - **0 = NO** (default): If **KeepAlive=0**, the connection is closed when the file transfer is ended.
 - **1 = YES**: If **KeepAlive=1**, the connection remains open until the end of the program or until it is terminated by an external factor. A connection is closed if, and only if, it was opened by Gateway. So even if **KeepAlive=1**, connections opened by an external application are never closed on end of transfer.
- **Force** = Close all active connections on program exit. A Boolean value:
 - **0 = NO** (default): If **Force=0**, only connections that were opened by Gateway are closed.
 - **1 = YES**: If **Force=1**, all active connections are closed on program exit (including those established by a foreign application).

How Gateway makes a connection using RAS

When you initiate an outgoing TCP/IP transfer, the software checks the destination network address. If it begins with [xxxxxxx], an RAS destination is assumed. [xxxxxxx] is extracted and used to retrieve further information from `RASPPP.INI`. The Linkentry, Username and Password are then passed to RAS networking to dial up the remote PPP server. If the dial-up succeeds, or the connection was already established and is still active, subsequent TCP/IP transfers are performed on this connection. If it fails, the connection request is refused. When a transfer is ended, the connection reference count is updated. If no connection reference is active, the RAS connection is closed. (Assume **KeepAlive=0**).

This algorithm allows for multiple parallel TCP/IP transfers over an RAS connection. The RAS connection can also be established outside Gateway, as long as it designates the same PPP server as that expected by Gateway. In this case, it is also the responsibility of this foreign application to maintain and terminate the RAS connection. If the foreign application closes the RAS connection while the Gateway transfer is still in progress, the transfer is aborted.

Common errors and troubleshooting

RAS never dials, in spite of a network address starting with [xxxxxxx]

Check that `RASAPI32.DLL` is on the Windows system directory and is accessible. This is the system DLL that Gateway uses to access RAS services. Its absence disables RAS features in Gateway.

If `RASAPI32.DLL` is present, check that the `p_ras_cfg` environment variable points to a correct `RASPPP.INI` file (by default and unless otherwise modified, `p_ras_cfg = <Gateway installation directory>\run_time\etc\RASPPP.INI`). Then ensure that the requested [xxxxxxx] exists and is defined with appropriate parameters.

If **Shared=0**, check that the port is not in use by another application.

Additional diagnostic messages are also available in the Gateway log file and standard error output file: `<Gateway installation directory>\run_time\tmp\TCP_xxxx.out`.

RASDial / Cannot find phone book entry

Check that RAS networking is functioning correctly. Use the RAS networking dial-up dialog to start an RAS connection manually with the same RAS phone book entry as that defined in the `<linkentry=>` parameter.

RASDial / Hardware failure in port or attached device

The modem may be out of service or off line.

RASDial / There is no dial tone

The modem may not be attached to the network.

RASDial / Line is busy

The line called is busy or not responding.

RASDial / Your modem (or other connecting device) reported an error

The modem may be out of order. Turn the modem off and on again and then retry.

If there is dial-up tone but error persists, the remote PPP server may not be started.

RASDial / The port is already in use or is not configured for dial out**RASDial / The port is not available**

The port is either already used by another application process or is not configured for outgoing connections. Check that the port is configured for outgoing calls. Multiple applications may share the same port (line), so the first case may not be an error.

Axway Gateway: Configuration

Network configuration for UNIX platforms

This topic provides information about configuring protocols on UNIX platforms for the following network interfaces:

- [TCP/IP](#)
- [SNA LU 6.2](#)

TCP/IP

Gateway supports TCP/IP transport for the following protocols:

- FTP
- HTTP
- SFTP
- SMTP
- POP3
- EDIINT
- RosettaNet
- PeSIT
- OFTP (Odette)
- PEL

To use TCP/IP transport with a protocol, you must indicate:

- The hostname (or IP address)
- The port number to use to route incoming calls to that protocol

Assign a different port number to each Gateway protocol configured for TCP/IP. Make sure that no other application running on your system uses these port numbers.

It is recommended that you select port numbers within the range 5000...32760. Each value must be used once only on the system. If you are currently using an X11 graphic display on your system, or intend to do so, do not select port number 6000 since the X Server uses this port number to listen to X client connections.

For more information, refer to the *Axway Gateway Configuration Guide (UNIX)*.

SNA LU 6.2

Gateway support for LU 6.2 communications is not available on LINUX platforms.

A separate Gateway process handles each LU 6.2 session. Gateway starts processes that handle requester sessions. For server sessions, you must define transaction programs (TP) in the LU 6.2 sub-system. This sub-system is responsible for starting a Gateway LU 6.2 server process on each incoming connection.

For more information, refer to the *Axway Gateway Configuration Guide (UNIX)*.

[Configuring protocols](#)

Axway Gateway: Configuration

Configuring connectors

Set up the **Axway Connectivity** parameters to enable Gateway to communicate using different connectors.

Configuring connectors

To configure a connector, select the desired connector in the left (tree view) pane, then complete the parameter fields (if any) in the right pane.

Prerequisites for all connectors

- The required software (for example Integrator) must be installed on your system.
- All parameter values must correspond to those defined for that software (for example Integrator).

Activating/deactivating connectors

In the configuration menu:

- To enable a connector in Gateway, click **Activate**.
- If you no longer want to use a connector, click **Deactivate** to disable it.

Note: For PassPort there is no **Activate** or **Deactivate** button. After completing the configuration parameters, the connector is activated.

Connectivity > Axway Connectivity > Integrator

Use the following parameters to configure Gateway for use with the Integrator (formerly XIB) connector.

If the payload integrity check is enabled for transfers received from Integrator, Gateway makes a signature validation before routing the incoming payload. The signature is received from Integrator, that is why Integrator's public key is required for signature validation

Parameter	Description
Gateway to Integrator connection	
Address	Enter the IP address in dotted format, or the DNS host name of the Integrator connector.
Port	Enter the port number assigned to the Integrator connector. This value must be in decimal format and unique on the operating system.
Shared directory	Enter a file directory name. Gateway must have read-write access to this directory. Click the browse button to help you locate and select the directory.
RSA public key	Path to Integrator public key in PEM format
Gateway to Integrator - secured by TLS	
Tls active	Optionally secure the connection with Integrator using TLS
CA file	The path to the file containing the certificate authority that validates the certificate of Integrator
Certificate file	The path to the file containing the certificate Gateway can use to authenticate itself as TLS client to Integrator

Parameter	Description
Key file	The path to the private key of the client certificate of Gateway
Password file	The path to the file containing the enciphered password Gateway uses to access the private key of its client certificate. To generate this file, use the <code>pelencpass</code> utility.
Password key file	The path to the file containing the key used to decipher the password file containing the enciphered password Gateway uses to access the private key of its client certificate. To generate this file, use the <code>pelencpass</code> utility.

Click **Activate** to enable the connector.

Setting TLS parameters with the command line

The parameters related to securing the connection between Gateway and Integrator can also be configured using the `peluconf` utility, in the `amtrix` section. Please use the utility to inspect the section for more information:

```
peluconf get -s amtrix
```

Connectivity > Axway Connectivity > PassPort

Use the following parameters to configure Gateway for use with:

- PassPort PS (PKI Services)
- PassPort PM (Partner Management)
- PassPort AM (Access Management)

Parameter	Description
am_cache_privileges yes/no	Enables or disables the batch caching of user privileges. The default behavior of the PassPort AM connector is to cache each privilege individually, as it is checked. You can optimize this behavior with <code>am_cache_privileges yes</code> : at the first privilege check, the connector retrieves and caches <i>all</i> the privileges of <i>all</i> the users in <i>all</i> the domains that are assigned to the Gateway component. This cache is refreshed with the frequency configured by the <code>am_cache_duration</code> parameter. Once the cache expires, a new request is sent to the server to retrieve all privileges.
am_cache_duration nbMinutes	If <code>am_cache_privileges</code> is set to <code>yes</code> , controls the number of minutes the authentication status and/or privileges of a user are cached. <code>am_cache_duration 0</code> means the cache is disabled.
retry_min	Back off mechanism parameters, in seconds. <code>retry_min</code> and <code>retry_max</code> work together to determine the length of time Gateway waits between two attempts to reestablish the connection with PassportAM. The first attempt times out after <code>retry_min</code> seconds. Delays for subsequent attempts double each time, until the value reaches <code>retry_max</code> .
retry_max	
PassPort Version	Select the version of PassPort you are using with Gateway. Possible values: - PassPort 3.3 - PassPort 3.4 - PassPort 4.0 - Select this value for any version of PassPort from 4.0 onwards Depending on the PassPort version, different configuration options will be available.
Use access management	Select this option if you use PassPort's access management (AM) service with Gateway. This option is enabled if the selected PassPort version is PassPort v4.
Use PKI Services	Select this option if you use PassPort's PKI services (PS) with Gateway.
Use partner management	Select this option if you use PassPort's partner management service (PM) with Gateway.

Parameter	Description
PassPort Host	Enter the PassPort host name or IP address.
PassPort Ports	
PKI	Enter PassPort's port for PKI service.
HTTP	Enter PassPort's port for access management (AM) and partner management (PM) services.
Component Id	<i>Optional</i> Enter a name to identify Gateway in PassPort PM. If you leave this blank, the default Local Site alias is used. This field is enabled only if the selected PassPort version is PassPort v4.
Component login	Enter the user name for PassPort PM server login. This field is enabled only if the selected PassPort version is PassPort version 4.0. The corresponding password must be specified by using the <code>pelencpass</code> command executed locally on the Gateway server machine. See the <i>Security Guide > Password management</i> for further information.
Secure connection	
Use TLS	Select this option to use TLS/SSL and then complete the proceeding parameters.
PassPort Secure Ports	
PKI SSL	Enter PassPort's secure port for PKI services (PS).
HTTPS	Enter PassPort's secure port for access management (AM) and partner management (PM) services. This field is enabled only if the selected PassPort version is PassPort v4 and Use TLS option is selected.

Parameter	Description
PassPort CA location	Enter the location (absolute path) where the PassPort CA certificate is located. Click the browse button to help you locate and select the directory. This value is first set at installation time. The default value for this location is <code>\$p_home_dir/extras/PassPort/PassportCA.crt</code>. This field is enabled only if the selected PassPort version is PassPort v4 and Use TLS option is selected.
Client certificate location	Enter the location (absolute path) where Gateway's certificate for Passport AM is stored. Click the browse button to help you locate and select the directory. This certificate is required because the secured connection between Gateway and PassPort uses mutual authentication. If you don't have a certificate, a new one will be generated automatically the first time Gateway connects to Passport AM. This automatically generated certificate will be stored at the client certificate location you specified. The name of the automatically generated certificate is Gateway.p12. This value is first set at installation time. The default value for this location is <code>%p_home_dir%/extras/PassPort</code>. The password of this certificate must be specified by using the <code>pelencpass</code> command executed locally on the Gateway server machine. See the Security Guide for further information. This field is enabled only if the selected PassPort version is PassPort v4 and Use access management and Use TLS options are both selected.
PassPort PM v3 – this section is enabled only if the selected PassPort version is PassPort v3.3 or PassPort v3.4	
Home directory	Enter the path and name of the directory where PassPort PM is installed. The location of this directory can be local or remote. Click the browse button to help you locate and select the directory.
Parameters file	Enter the path and file name of the PassPort PM parameters file. Click the browse button to help you locate and select the file.

Note:

- After completing the PassPort configuration parameters, the connector is activated.

Configuring PassPort connector in command line

From Gateway 6.16 onwards, the `[passport]` parameter section is added to the `dconfsave.ini` file for unified PassPort configuration. This section is used by the GUI, and Gateway uses the parameters in this section by default when communicating with PassPort.

The old sections (`[am]` and `[tpm]`) have been preserved for backwards compatibility reasons. In case of legacy Gateway instances, the existing settings are also preserved. Although the new `[passport]` section is active by default, it is still possible to continue to use the old parameters by executing the following command:

```
peluconf set -s [passport] use_legacy_settings
```

The old parameters can be modified only via command line.

Connectivity > Axway Connectivity > Sentinel

Use the following parameters to configure Gateway for use with Sentinel.

Parameter	Description
Host name	Enter the IP address or the host name of the Sentinel server or Event Router. Maximum: 64 alphanumeric characters.
Port	Enter the TCP port number of the Sentinel server or Event Router. Maximum: 32 numeric characters.
Backup host	Enter the IP address or the host name of the backup Sentinel server. Maximum: 64 alphanumeric characters.
Backup port	Enter the TCP port number of the backup Sentinel server.
Transfer filter	Specify the level of information to send to Sentinel regarding Transfer state changes. Select one of the following values: • All: complete trace for all Transfer state changes • Summary: creation and end of Transfer only • None: no trace

Parameter	Description
Log filter	Specify the level of log information to send to Sentinel or the Event Router. Select one, or more, or none, of the following options: • Success • Information • Warning • Error • Failure Note: You can also set this parameter in the Remote Site, the Application and the Transfer Request objects.
sentinel_use_tls	Toggles the use of TLS for securing connection to Sentinel
sentinel_ca_file	Configures the certificate authority file of the sentinel server Gateway is configured to connect to

Click **Activate** to enable the connector.

Connectivity > Axway Connectivity > Messaging

You do not need to set any parameters for Axway Messaging (formerly XMS). Just click **Activate** to enable the connector.

Connectivity > Axway Connectivity > JMS

You do not need to set any parameters for JMS. Just click **Activate** to enable the connector.

Connectivity > Axway Connectivity > MFT Navigator

Use the following parameters to configure Gateway to use an Axway MFT Navigator (formerly IUI) server to support remote Internet or Intranet clients.

Parameter	Description
Listening port (mandatory)	Enter the TCP port number assigned to the MFT Navigator Server. The port number must be in decimal format and unique on the Operating System. Default = 6300.
Network interface (mandatory)	Enter the local IP address in dotted format, or the local host name as defined in the HOST file.
<i>The following fields are optional. You can activate the MFT Navigator server without using the TLS option.</i>	
Use TLS	Select this option to use TLS/SSL and then complete the following parameters.
X509 certificate	Enter the file name that contains the X.509 certificate to be used by the MFT Navigator server. Typically, this is a file in PKCS 12 format.
Password file	Enter the file name that contains the password used to access the X.509 certificate.

If you want to modify the cipher suites, click **Advanced...** to access *MFT Navigator advanced options*.

From the list, select one or more of the encryption algorithms supported by Gateway, then click **OK**.

Click **Activate** to enable the connector.

Connectivity > Axway Connectivity > SWIFTNet

Use the following parameters to configure Gateway for use with SWIFTNet.

General tab

Parameter	Description
Default RA installation pathname	Enter the default directory path name for SWIFTNet RA. Example: C:\SWIFTAlliance\RA Click the browse button to help you locate and select the directory.

Parameter	Description
Accept possible duplicate transfers	<i>FileAct only</i> Check this option to accept incoming SWIFTNet Possible Duplicate transfers.
Limit the number of initiated transfers	<i>FileAct only</i> Enable the limitation of the number of transfers in Initiator mode for FileAct only (delivery notifications excluded). The value of the maximum number of transfers is set on a SWIFTNet remote site. By default, this option is disabled.
Global maximum number of initiated transfers	The global value of the maximum number of transfers in Initiator mode, FileAct only. By default, this parameter is set to 0, meaning unlimited. The maximum value allowed is 999999. The number of initiated transfers is counted as the sum of FileAct transfers initiated for all remote sites.
Subscribe to all file events	<i>FileAct only</i> By default this option is enabled, allowing Gateway to subscribe to all file events. To improve performance, you can deactivate this option and disable the subscription to the ongoing (intermediary) events. Note that when this option is deactivated, the statistics for the number of bytes transferred is not updated. If this option is enabled, you must update the transfer status polling mechanism delay to a higher value (over 300) to avoid unwanted or unnecessary transfer state queries when intermediary file events are disabled, especially for large FileAct transfers.
Proof keeping	Check this option to activate SWIFTNet proof keeping.
Default proof directory	Specify the destination folder for SWIFTNet proof keeping. Either browse or manually set the folder. You can use Gateway environment variables: • %VAR% and \ separator for Windows • \$VAR and / separator for UNIX

Parameter	Description
Customize file info parser	Select whether or not to customize the SWIFT grammar rules for the Sw:FileInfo field. This is a global setting for all partners. Warning: In normal circumstances you should not override the default values provided in Gateway: • Separator: ; [semi-colon] • Equals: = [equals sign] • Quotes: " [quote mark] These default values correspond to the grammar for the Sw:FileInfo field as defined in the SWIFT documentation. In some specific cases, for example for backward compatibility with older applications that are not SWIFT compliant, you may need to modify one or more of these three special characters. You can override these global settings in each individual Trading Partner. This global setting enables you to easily change the values for all partners without having to modify each Trading Partner. If you modify the global values you must restart Gateway for the new values to be taken into account.
[Transfer status polling mechanism] Use the parameters Transfer status polling interval and Transfer status polling delay in combination with each other. By default the polling mechanism is disabled (Transfer status polling interval = 0).	
Transfer status polling interval	Specify the polling interval in seconds. Occasionally events from SWIFT concerning the transfer status (such as completed) can get "lost". This could result in an incorrect transfer status in Gateway. To ensure that the transfer status in Gateway is always up to date, Gateway can send a GetFileStatus primitive to SWIFT at regular intervals. This only happens if the Transfer status polling delay has been reached. After checking the received status, Gateway can update its transfer status to match the status at the SWIFT side. You can use this mechanism for incoming or outgoing FileAct messages. If you set the value to 0, this mechanism is disabled. If you use any other value, the polling mechanism is activated. Possible values: 0 – 7200. Default = 0.

Parameter	Description
Transfer status polling delay	Specify the polling delay (in seconds). The polling delay is the time passed from the last transfer update (request sent to or received from SWIFT, or a File Event) after which the transfer can be selected for a GetFileStatus operation. Possible values: 10 – 7200. Default = 60.
Number of compression/ decompression processes	Specify the number of helper processes to handle the compression/ decompression operations. Possible values: 1 – 100. Default = 1. All processes are started at Gateway startup. Therefore, if you modify this setting you must restart Gateway for the new value to be taken into account.

UserExit tab

Parameter	Description
Activate user exit	Check this option to use the user exit process to control incoming and outgoing Transfer Requests and to recover some transfer parameters.
Address	Enter the TCP IP address/host address for the internal user exit process listen interface. Default = 127.0.0.1
Port	Enter the listening port for internal user exit process connections. Default = 6666.
Timeout	Specify the incoming request maximum duration (timeout) in seconds.

Store and Forward tab

Parameter	Description
SnF check timer	Specify the time interval (in seconds) during which Gateway checks the status of the acquired SnF session in case no activity has been noticed on that queue in that time interval. The most suitable time interval for your environment depends on the SnF traffic volume and time constraints. When the timer expires, Gateway initiates an SnFStatusRequest and reports the returned status. Gateway uses this configuration parameter to monitor the status of acquired SnF sessions. For several reasons, in a time interval, it is possible that there is no activity (no pushed message request) on a given acquired queue. For example, if someone else reacquired the queue with the force option, thus the session became invalid; SWIFT has released the SnF session due to some internal technical issues; simply there is no message in our queue, and so on. Note that if a message request has been pushed on an SnF session that means the session is Active, and there is no reason to check the status of the acquired SnF session, the SnF check timer is reset for that SnF session (the one on which a message has been pushed). Possible values: 0 – 7200 (2 hours). Default = 120 (2 minutes).
Reacquire SnF session automatically	Select whether or not an SnF session should be reacquired automatically if it is lost. By default this option is enabled, so that the SnF session is reacquired when it is no longer valid. You can deactivate this option if required (for example if the same queue is used in more than one environment, or any other reason). For example, if the same queue is acquired on another system, the current SnF session will become invalid. Gateway detects this and reacquires the queue automatically.

ASP/RMA tab

Parameter	Description
ASP/RMA Check	Activates the ASP check and RMA filtering. Values: • ASP & RMA enabled • ASP enabled, RMA disabled • ASP & RMA disabled:(default)
ASP helpers	Number of processes started for handling the ASP check operations.
ASP Do not change default values unless absolutely necessary (for example, SWIFT changes file names).	
ASP Profiles Directory	Select the directory containing the ASP files Gateway loads at startup or reloads during runtime if requested
ASP Profile Schema	XML schema for ASP profiles.
ASP Profile Extension	File extension of the ASP profiles.
ASP Digest File	Name of the XML file from the ASP package containing the digests for all ASP profiles in the package.
ASP Digest Schema	XML schema for the digest file.
RMA	
RMA User Exit Library	If the RMA authorization check is enabled, enter or select the full path of the RMA user exit library, Gateway loads the library at startup.
RMA Configuration	If the RMA authorization check is enabled, enter or select the full path of the RMA configuration file that is passed to the RMA user exit library.
Stop unauthorized transfers in Trial Period	Specifies whether to stop unauthorized transfers in trial period. • Enable to stop. The behavior is the same as mandatory period. • Disable to allow unauthorized transfers in trial period. This is the default.

Click **Activate** to enable the connector.

HA (High Availability) tab

Parameter	Description
High Availability Exit	Activates and set the High Availability operation mode. Possible values: • HA operation disabled • Use correlation exit • Use external correlation
HA Correlation Exit	Do not change the default value unless absolutely necessary (for example the name of the HA library has changed). Enter the path to the High Availability Exit Library. Click the browse button to help you locate and select the file.
Database Type	Select the database type (MYSQL or ORACLE).
DB Data Source Name	Select the database DSN name.
DB Connection User	Select the database DSN user.
DB Connection Password	Not editable. The DB connection password must be encrypted using the <code>pelencpass</code> command, and the <code>swnet_ack_userexit_db_salt_file</code>, <code>swnet_ack_userexit_db_dk_file</code> and <code>swnet_ack_userexit_db_password_file</code> configuration parameters must be set accordingly. See <i>Security Guide</i> > Secure Password Storage for further information.

Note Parameters related to the HA Correlation Exit (*HA Correlation Exit*, *Database Type*, *DB Data Source Name*, *DB Connection User*, *DB Connection Password*) are only required if **High Availability Exit** is set to `Use correlation exit`.

Note The PeSIT Duplicate Checking mechanism uses the HA Correlation Exit, thus requiring that all parameters be set.

See the [SWIFTNet High Availability User Exit Configuration](#) and the [SWIFTNet High Availability with External Correlation on page 2180](#) sections for more details about configuring the SWIFTNet High Availability feature.

Connectivity > Axway Connectivity > X.400

Use the following parameters to configure Gateway for use with X.400.

Parameter	Description
Local MTA	
Name *	Enter a name (public name) for the local Message Transfer Agent (MTA).
TSAP *	Enter the TSAP (Transport Service Access Point). Maximum 20 characters. To specify a hexadecimal value, enter the value between apostrophes ('), for example, ' 7C, A2 '.
O/R address *	Specify the O/R address attributes that are common to all local users.
Edit	Click Edit... to configure the O/R address .
Clear	Click Clear to delete the displayed O/R address.
Advanced settings Use the following buttons to access advanced parameters.	
Server ports...	Refer to Advanced server ports settings .
OSITP...	Refer to Advanced OSITP settings .
MTA...	Refer to Advanced MTA settings .
MTC...	Refer to Advanced MTC settings .
MS...	Refer to Advanced MS settings .
UA...	Refer to Advanced UA settings .

(*) Mandatory parameter

Important: For X.400, make sure that you complete all configuration parameters and check that the entered information is coherent **before** activating the connector. Failing to observe this rule could result in unpredictable behavior.

Click **Activate** to enable the connector.

Axway Gateway: Configuration

Modifying conffile parameters

You can modify certain configuration parameters in the `conffile`. To make the modification permanent, you must modify the `pelsetup.def` file and then rebuild the configuration file.

[Modifying pelsetup.def](#)

[SWIFTNet parameters in the pelsetup.def file on page 237](#)

[Configure where log files are stored in the pelsetup.def file on page 238](#)

Modifying pelsetup.def

1. Go to the directory: `<Gateway installation directory>/run_time/etc.`
2. Edit the `pelsetup.def` file as required. (See the examples for SWIFTNet below.)
3. Rebuild the configuration file using the following command:
`peluconf standalone rebuild -f`
4. **Restart** Gateway so that the new values are taken into account.

Gateway's environment variables may contain either absolute or relative paths. You must run the profile before any operation related to your Gateway installation.

If you have issues after an update or upgrade (e.g: upgrade pack), you can use `peluconf standalone rebuild -f` to regenerate the configuration.

SWIFTNet parameters in the pelsetup.def file

The table below provides details of SWIFTNet parameters in the `pelsetup.def` file. You can find these parameters (or add them) in the SUP section of the file.

Parameter	Description	Exists by default?	Default value
<code>swift_job_retry_count_max</code>	This parameter controls the number of retries in the case of a connection error with SAG. The parameter value is taken into account by client and server processes. Example: If the connection to SAG is broken, the client/server process tries to reestablish the connections as many times as specified by this parameter.	no	10

Parameter	Description	Exists by default?	Default value
swift_job_restart_delay	This parameter stores the timeout (in seconds) between two successive connections attempts to SAG. The parameter value is taken into account by client and server processes. Example: If the connection to SAG is broken, the client/server process tries to reestablish the connections after the period of time given by this parameter.	no	30

Example of part of pelsetup.def file contents

```
[SUP]
trace=0
.....
swift_job_retry_count_max = 7
swift_job_restart_delay = 600
.....
```

Configure where log files are stored in the pelsetup.def file

You can determine where log files are stored.

To do that, set the `archived_logdir` parameter in the `[LOG]` section of `pelsetup.def` file. For example:

```
[LOG]
archived_logdir="%p_database%"
```

If the `archived_logdir` parameter is not set, the archived logs will be saved in the same folder where the `log.dat` file is located.

Application Service Profiles (ASP)

Introduction

An Application Service Profile (ASP) specifies the properties of a SWIFT service. The ASP contains a set of parameters decided by the Service Administrator when defining the service. The ASP is used by messaging interfaces and applications to correctly send and receive traffic for that service.

These parameters trigger the use of features within SWIFTNet or describe what a messaging interface must do with traffic sent or received.

The ASP contains the following parameters available for SWIFTNet services:

- RMA-related parameters:
 - Whether and how the RMA authorizations are exchanged (the use of RMA is defined per request type pattern)
 - Whether and how the RMA filtering is applied (the use of RMA is defined per request type pattern)
 - The date from which RMA will apply, if it applies
 - The request types for which RMA is required
 - Whether or not local/bootstrapped authorizations are allowed
 - The date from which the RMA protocol must be used to exchange authorizations
- End-to-end signing required and what format the signature can have
- Non-repudiation signing required
- Store-and-forward usage
- SWIFTNet Copy parameters
 - Copy usage (forbidden, optional, mandatory)
 - Copy mode (T-copy, Y-Copy, list of third parties)
- HeaderInfo usage (summary of the business data inserted within the request envelope)
 - Is it forbidden, optional or mandatory?
 - HeaderInfo schema

ASP package

The use of the ASP by the SWIFTNet InterAct and FileAct messaging interface is an optional feature. However you require ASP as soon as RMA filtering is enabled within the interface.

SWIFT provides an ASP package which is a ZIP file containing one file per service. SWIFT updates the ASP package periodically.

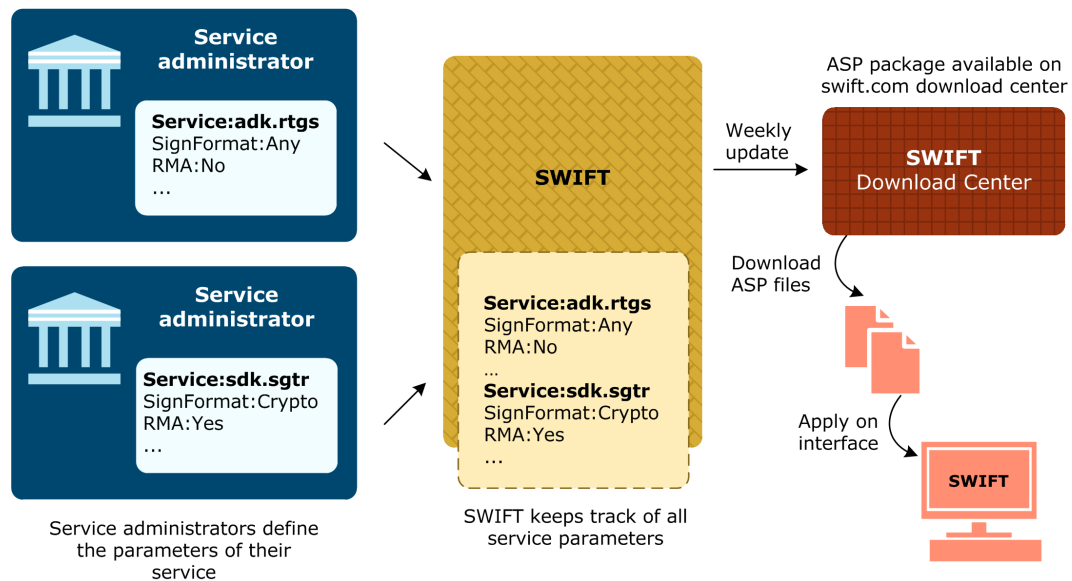
You can download the ASP package from the SWIFT Download Center at www.swift.com. Make sure you download the latest ASP package and import the ASP files to FEX RMA.

When emergency updates have to be done, a new file is made available for download with a description of the changes that have been implemented. The service administrator requesting these emergency updates will inform the members of the service(s) if the update requires the latest version to be used.

ASP lifecycle

The following figure shows the ASP lifecycle.

Application Service Profiles (ASPs)



More information

For more information about ASP management, refer to the SWIFTNet Application Service Profile documentation.

Relationship Management Application (RMA)

Introduction

Gateway RMA (Relationship Management Application) is an optional service of Gateway. It manages the exchange of SWIFTNet authorizations between partners. This first version of Gateway RMA concentrates on RMA filtering using Axway Gateway.

This version of Gateway RMA includes the following features:

Administration:

- Ability to import full distribution file in order to populate Gateway RMA with authorizations
- Ability to import Application Service Profiles (ASP), specifically for the RMA-related parameters
- Predefined roles used to manage access to the application are delivered

Filtering:

- An Axway Gateway user exit is provided. This enables the filtering of outbound and inbound SWIFTNet flows using the RMA authorizations stored in Gateway RMA.

Gateway services for Gateway RMA

Gateway RMA is used as part of Gateway in combination with the following products:

- Axway Gateway
- PassPort AM

Summary of Gateway RMA features

This section provides an overview of the features available in Gateway RMA. For a complete description, refer to the relevant topics in this document.

Housekeeping

The housekeeping scripts enable you to start, stop and check the current status of the RMA Server.

Access control

Gateway RMA uses PassPort AM for access control management. Predefined roles are delivered. When you start Gateway RMA for the first time, it automatically imports all required data into PassPort.

PassPort AM provides a Single Sign On (SSO) feature. Single sign-on enables users to log in to multiple Axway products with just one user name and password.

Checking and importing RMA distribution files

Gateway RMA allows you to import RMA distribution files. These are XML files that contain authorizations. Checking a distribution file consists of making sure that it is valid without actually importing it to the RMA Server.

Importing an ASP package

After you install and configure Gateway RMA, the ASP data tables are empty. You must download an ASP package from SWIFT and import it.

RMA filtering exit

Gateway provides an RMA filtering exit that is used in conjunction with Axway Gateway. Gateway uses the RMA filtering exit to make authorization requests to the Gateway RMA Server for transfers it receives or wants to send. Depending on the response of the RMA Server, the transfers will either proceed or be canceled.

RMA Server administration commands

The RMA command line enables you to act on a running RMA Server. The command line enables you to perform operations such as importing a distribution file or an Application Service Profile (ASP).

About RMA

Introduction

Relationship Management Application (RMA) is a service provided by SWIFT to help manage the business relationships between financial institutions. SWIFTNet 7.0 enables RMA capabilities for services based on InterAct and FileAct.

RMA consists of two main functions:

- Authorization: Customers exchange or create authorizations, indicating "who can send traffic to whom"
- Filtering: The customers' messaging interfaces apply these rules to control which traffic they can exchange with which correspondent

RMA uses a SWIFTNet InterAct Store and Forward service to exchange the permission data between financial institutions.

RMA usage is defined per service. The rules regarding its use are specified in the Application Service Profile (ASP), associated with each service.

Advantages of using RMA

RMA provides a SWIFT user with more control over the sending or receipt of messages in a many-to-many InterAct or FileAct service:

- It provides full control over the traffic received from a user's correspondents.
- It allows customers to control the traffic they receive from other users in a many-to-many environment, thus ensuring that traffic is only exchanged with those users with whom they have a business relationship.
- It can reduce operational risk – unwanted traffic is blocked at the sender's side when used on a bilateral basis
- It can enable the fine tuning of correspondent banking relationships – customers only receive the traffic reflecting the type of business that they conduct with their correspondent

RMA service parameters

As of SWIFTNet 7.0, service parameters, including those for RMA, are defined in the relevant Application Service Profile (ASP).

Service Administrators define the parameters of their service. Each Service Administrator must provide service participants with the information they need to correctly use the service in question, including RMA-related information.

SWIFT keeps track of all service parameter and makes them available to all participants in a downloadable ASP package.

RMA authorizations

Within the context of FIN and SWIFTNet services, RMA standardizes:

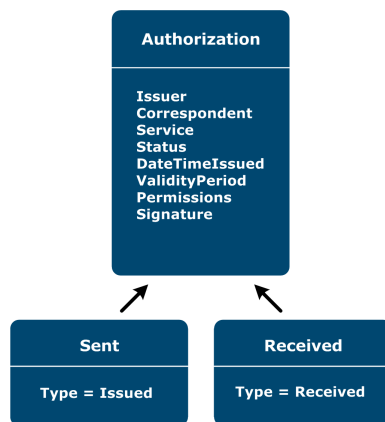
- How two partners agree to exchange documents (MTxxx for FIN, RequestType for InterAct/FileAct)
 - By exchanging authorization messages on a dedicated InterAct service
 - Or, for a transitory period, through a manual, local authorization process ("Bootstrap mode").

- The lifecycle of the authorizations
- How the authorizations can be exported and imported in a secure way, either for backup reasons, or to make authorizations available to specialized tools (for example, for filtering)
- How the authorizations are used to filter outgoing and incoming requests within the SWIFTNet messaging interfaces

Note: You do not have to renew an RMA authorization as long as the business relationship with your correspondent does not change.

RMA authorization data model

The following figure shows the data model of an RMA authorization.



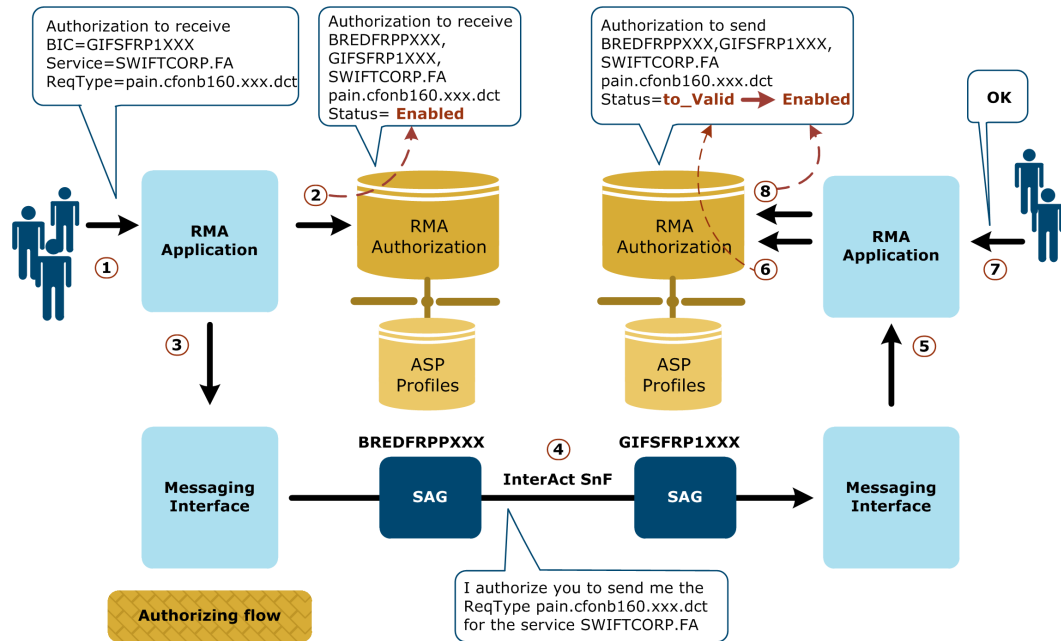
In this figure:

- The *issuer* is the sender of the authorization to receive, and therefore the receiver of the flow which is the object of the authorization. The issuer is identified by their BIC code
- The *correspondent* is the receiver of the authorization to send, and therefore the sender of the flow which is the object of the authorization. The correspondent is identified by their BIC code
- The *service* is the SWIFTNet service, object of the authorization
- The *status* is the position of the authorization within the RMA lifecycle
- The *dateTimeIssued* is the date of the beginning of the validity of the instruction contained within the authorization
- The *validityPeriod* is the period of application of the permissions contained within the authorization
- *Permissions* is a structure which depends on the service related to the authorization:
 - SWIFTNet service: The permission is a list of request type patterns and permission criteria which can have the value Exclude or Include
 - FIN service: The permission is a list of sub-structures (maximum one per category of FIN message). Each sub-structure is a list of FIN format patterns and permission criteria which can have the value Exclude or Include
- The *signature* allows the authentication of the authorization

RMA authorization lifecycle

When RMA is mandatory for a service, meaning when authorizations are exchanged, the initiator of the authorization exchange is always the receiver of the flow to authorize.

Authorizing a flow

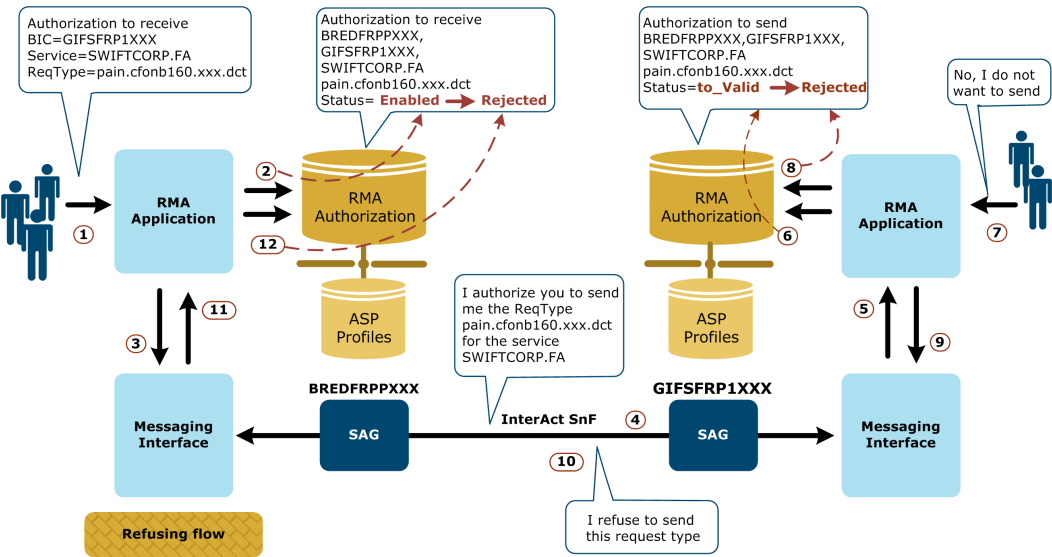


Steps in the authorization flow:

Step	Description
1	The receiver creates an authorization to receive. The receiver is the issuer of the authorization
2	The authorization is stored in the RMA data store with the status "Enabled"
3	The authorization is submitted to the messaging interface
4	The authorization is received by the messaging interface of the correspondent that is the sender of the flow to authorize
5	The authorization to send is transmitted to the RA application
6	The authorization to send is stored in the RMA data store with the status "To validate"
7	The user accepts the authorization to send

Step	Description
8	The authorization to send status is set to "Enabled". It will be taken into account at the filtering time

Refusing to send a flow



Steps in the authorization flow:

Step	Description
1	The receiver creates an authorization to receive. The receiver is the issuer of the authorization
2	The authorization is stored within the RMA data store with the status "Enabled"
3	The authorization is submitted to the messaging interface
4	The authorization is received by the messaging interface of the correspondent that is the sender of the flow to authorize
5	The authorization to send is transmitted to the RA application
6	The authorization to send is stored in the RMA data store with the status "To validate"
7	The user rejects the authorization to send

Step	Description
8	The authorization to send status is set to "Enabled". It will be taken into account at the filtering time
9	The rejection is submitted to the messaging interface
10, 11	The rejection is received by the messaging interface and transmitted to the RMA application of the issuer of the authorization
12	The status of the authorization is set to "Rejected" in the data store of the issuer of the authorization

Deleting an authorization

The receiver of an enabled authorization to send (sender of the authorized flow) deletes the authorization.

- The authorization status is set to "Deleted" within his data store
- An RMA Reject message is sent to the issuer of the authorization to receive
- The issuer of the authorization to receive (the receiver of the authorized flow) sees and processes the event as a rejection of authorization

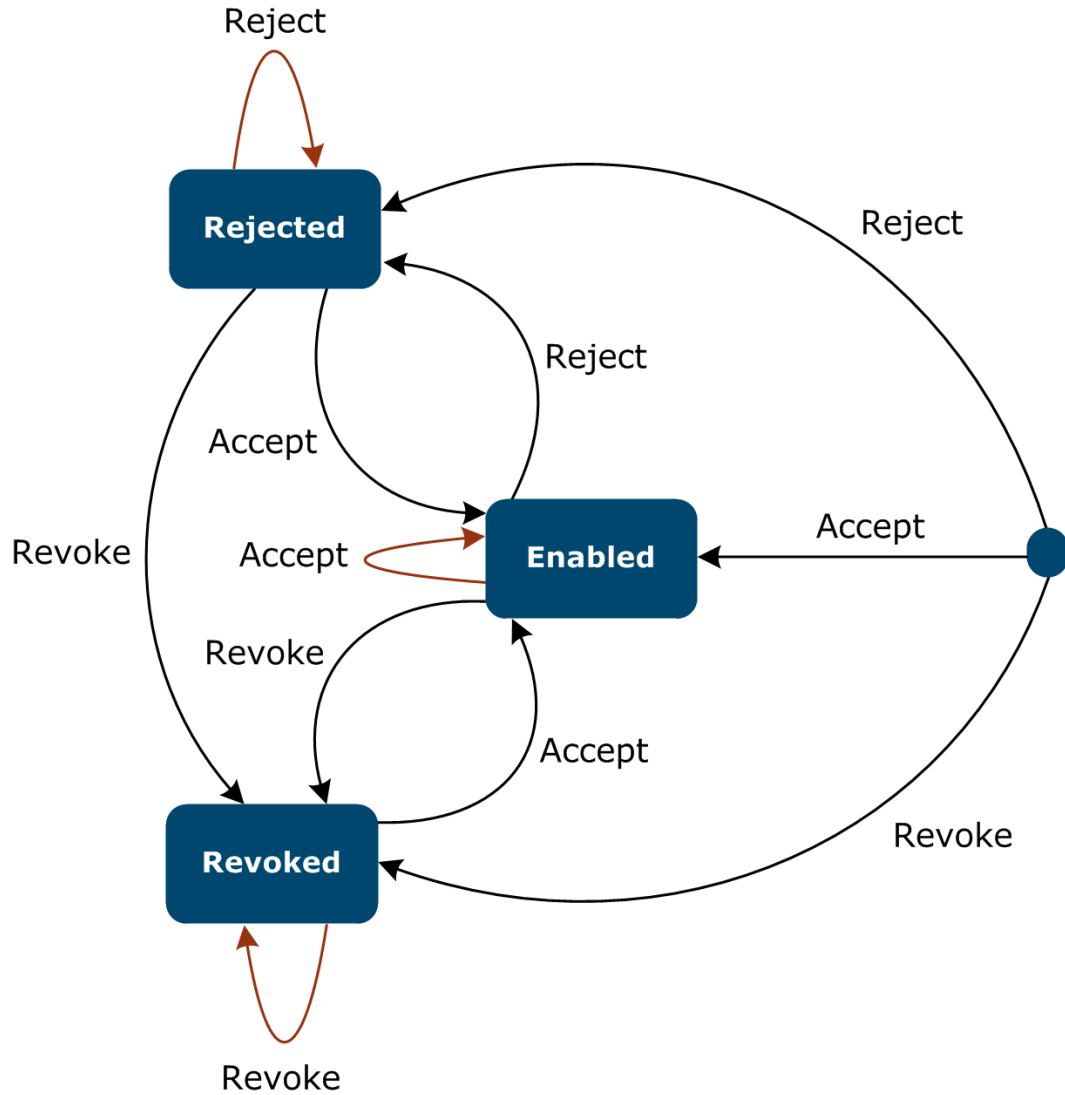
Revoking an authorization

The issuer of an enabled authorization to receive (receiver of the authorized flow) revokes the authorization.

- The status of the authorization to end is set to "Revoked" within his data store
- An RMA Revoke message is sent to the receiver of the authorization to send
- At the receiver site (the sender of the authorized flow), the status of the authorization to send is set to "Revoked".

Authorization to receive: state transitions

This diagram shows the state transitions relating to authorization to receive.



Authorization to send: state transitions

This diagram shows the state transitions relating to authorization to send. Transitions from "Revoked" to "Rejected" and vice versa are not shown.



RMA filtering

FEX provides an RMA filtering exit that is used in conjunction with Axway Gateway. The exit is used by Gateway to interface itself with the Gateway RMA Server to provide RMA capability.

RmaQuery and RmaAnswer

In some cases, Relationship Managers may wish to communicate with their peers within another organization to ask questions or to resolve problems with the authorizations exchanged.

The free-format RmaQuery message can be used to do this. The response to an RmaQuery message is an RmaAnswer message. The peer Relationship Manager sends the RmaAnswer (in response to an RmaQuery) back to the sender of the RmaQuery.

More information

For more information about RMA, refer to the following SWIFT documents:

- *RMA Service Description*
- *RMA Operations Guide*
- *Relationship Management Application Vendor Specifications*

Importing an ASP package

This topic explains how to import an Application Service Profile (ASP) package. You use the same procedure to update ASP profiles.

Introduction

After you install and configure FEX RMA, the ASP data tables are empty. You must import an ASP package, available from SWIFT.

SWIFT updates ASP package files regularly. To keep the ASP profiles up to date, download and import a new ASP package.

Downloading the latest ASP package

1. Download the latest ASP package from the SWIFT Download Center at www.swift.com.

The ASP package is contained in a ZIP archive.

2. Note the following information:
 - The path to the ZIP archive containing the ASP package
 - The corresponding security number (provided as an SHA-256 digest in Base64 format) that secures the ASP package

Importing (or updating) ASP profiles

Prerequisites

- You must be a user with the "ASP Manager" (as defined in PassPort AM) role to be able to use this command.
- Ensure that you have:
 - The path to the ZIP archive containing the ASP package
 - The digest that secures the ASP package

Import procedure

To import the ASP package, you use the RMA command line.

For more information about this command, and its parameters, refer to [RMA Server administration commands](#).

Enter the command:

```
rmaCmdLine.bat | rmaCmdLine.sh --importASP --user <user name> --password <password> --domain <domain> --file <ASP archive path> --key <ASP package digest> [--digestAlgorithm <digest algorithm>]
```

Make sure that you specify the full path name to the ZIP archive containing the ASP package. The ZIP archive must be accessible by the server.

The digest algorithm is optional: the RMA Server uses SHA-256 by default.

Examples

```
rmaCmdLine.sh -s -u myusername -p mypassword -d mydomain -f /full/path/name/to/asp/package/ASP.zip -k /QJL4lrueJcjYzRTJxTzNhVawYdj1HUq/zRlApVTTgA=
```

```
rmaCmdLine.sh --importASP --user myusername --password mypassword --domain mydomain --file /full/path/name/to/asp/package/ASP.zip --key /QJL4lrueJcjYzRTJxTzNhVawYdj1HUq/zRlApVTTgA=
```

The RMA Server extracts all files in the ASP package to a temporary folder and performs the following checks before storing the RMA service profiles in the database:

- Integrity check:
 - The entire package is checked against the specified digest
 - Each individual service profile file is checked against its digest such as provided within the dedicated file in the package
- Content check:
 - Each individual service profile file is checked against the XSD schema defined in the package
- Consistency check:
 - If an ASP profile has already been loaded then the version date of the corresponding profile file contained in the newly-received package must be equal or later than the version date of the current active profile. If the version check fails then the whole package is rejected.

As long as no error occurs during the validation of the ASP package, the RMA service profiles are stored in the database.

Housekeeping: starting and stopping the RMA Server

This topic explains how to use the RMA housekeeping scripts and how to purge log, audit and trace files if they become too large.

About RMA housekeeping scripts

All the RMA scripts are located in the `<FEX_RMA_INSTALL_DIR>/bin` directory.

These scripts include the housekeeping scripts that enable you to start, stop and check the current status of the RMA Server. The housekeeping scripts do not require any parameters.

Running the RMA housekeeping scripts

1. Go to the `<FEX_RMA_INSTALL_DIR>/bin` directory.
2. Run the script for your OS:

Task	Command	
	on Windows	on UNIX
Start RMA Server	<code>rmaStart.bat</code>	<code>rmaStart.sh</code>
Stop RMA Server	<code>rmaStop.bat</code>	<code>rmaStop.sh</code>
Check status of RMA Server	<code>rmaStatus.bat</code>	<code>rmaStatus.sh</code>

After starting the RMA Server you can check that it is fully booted by consulting the server `rma.log` file, located in the `<FEX_RMA_INSTALL_DIR>/logs` directory.

Alternative methods of starting the RMA Server on Windows

On Windows you can also use these methods to start the RMA Server:

- From the Windows **Start** menu, select **Start Gateway RMA**.
- Double-click the Start Gateway RMA icon on your desktop.

Installing (or removing) the RMA Server Windows service

To install the RMA Server Windows service, run the script

```
<InstallationDir>/RMAServer/service/installRMA.bat.
```

To remove the RMA Server Windows service, run the script

```
<InstallationDir>/RMAServer/service/removeRMA.bat.
```

Starting and stopping the RMA Server in Windows service mode

If, during installation, FEX RMA was installed as a Windows service, you can start and stop the RMA Server as follows.

Starting the RMA Server

1. From the Windows **Start** menu , select **Settings > Control Panel > Administrative Tools > Services**.

Windows displays the *Services* window.

2. Scroll down the Services list and right-click the FEX RMA service.
3. From the context menu, select **Start**.

Alternatively, run the script `<InstallationDir>/RMAServer/service/startRMA.bat`.

Stopping the RMA Server

1. From the Windows **Start** menu , select **Settings > Control Panel > Administrative Tools > Services**.

Windows displays the *Services* window.

2. Scroll down the Services list and right-click the FEX RMA service.
3. From the context menu, select **Stop**.

Alternatively, run the script `<InstallationDir>/RMAServer/service/stopRMA.bat`.

Purging log, audit and trace files

Log, audit and trace files can become quite large after time. You can delete the following files if you no longer need the information:

Type of file	File name	Location
RMA Server log	<code>rma.log</code>	<code><FEX_RMA_ INSTALL_ DIR>/logs</code>
RMA command line log	<code>rma_cmdline.log</code>	<code><FEX_RMA_ INSTALL_ DIR>/logs</code>
Audit	<code>audit.log</code>	<code><FEX_RMA_ INSTALL_ DIR>/logs</code>

Configuring access control for Gateway RMA

This topic explains how to use PassPort AM to manage access control for Gateway RMA.

Overview

Gateway RMA uses PassPort AM to store the following:

- User definition in PassPort AM repository or in an LDAP repository
- Role definition
- Link between users and roles

Importing required data into PassPort

Predefined roles for access management are delivered with Gateway RMA. When you start Gateway RMA for the first time, it automatically imports all required data into PassPort:

- *Resources* that can be protected by permissions
- Predefined *privileges*

- Predefined *roles*

Resources, privileges and roles

Resources

Basically, Gateway RMA has two resources: Authorization and ASP.

Resource	Description
Authorization	Represents the authorization that can be imported and used during filtering steps.
ASP	Related to the management of the ASP (Application Service Profile) package.

Privileges

Three privileges are predefined: Manage ASP, Read authorizations and Manage authorizations.

Privilege	Description
Manage ASP	Allows a user to import an ASP package into Gateway RMA.
Read authorizations	Allows a user to only read authorizations. This is the privilege which needs to be used for filtering purposes (for example).
Manage authorizations	Allows a user to create, update, read and delete authorizations.

Roles

Three roles are predefined: ASP manager, Authorization reader and Authorization manager.

Role	Description
ASP manager	All users assigned to this role have <i>Manage ASP</i> privileges. They can import ASP packages.
Authorization reader	All users assigned to this role have <i>Read authorizations</i> privileges.
Authorization manager	All users assigned to this role have <i>Manage authorizations</i> privileges. They can create, update, read and delete authorizations.

Note: For more information about resources, predefined privileges and predefined roles, refer to the Axway PassPort documentation.

Creating users in PassPort

Log in to the PassPort UI with an administrator user ID and create all the users you need for Gateway RMA.

You can then assign roles to the created users.

For more information about creating users and assigning roles, refer to the *Axway PassPort Administration Guide*.

Note: Instead of storing user definitions in PassPort, you can use an external LDAP directory. You have to connect the LDAP directory to PassPort. Refer to the *Axway PassPort Administration Guide* for more information.

Updating PassPort properties

1. Log in to the PassPort UI with an administrator user ID.
2. Browse to the System properties menu.
3. Set the property: `am.users.by.domain.query=true`

Checking and importing an RMA distribution file

This topic explains how to check and how to import an RMA distribution file.

Introduction

About distribution files

Distribution files are XML files that contain authorizations. They are useful for bulking import/export authorizations from/to the RMA Server. The file format is standardized so that distribution files can be processed by any RMA-compatible application.

Distribution file integrity and authenticity is ensured by a Local Authentication (LAU) block based on a hash function and a key (128 or 256 bit long).

Distribution file check

Checking a distribution file consists of making sure that it is valid without actually importing it to the RMA Server. This can be useful for large files: you do not risk trying to import the file and then discovering that the import failed because it is not valid.

Distribution file import

Importing consists of doing the same checks that are done in the check command above and then writing the authorizations contained in the file to the RMA database.

A distribution file can have one of the following file modes:

- Complete
- Partial (not supported in this version of FEX)

Additionally there are several import modes. Currently FEX only supports "restore" mode. In this mode all authorizations contained in the database are removed and the new ones are created from the import file.

Checking an RMA distribution file

To check a distribution file without importing it, enter the command:

```
rmaCmdLine.bat | rmaCmdLine.sh --check --user <user name> --password  
<password> --domain <domain> --file <distribution file path> --key  
<distribution file LAU>
```

Make sure that you specify the full path name to the distribution file. The distribution file to import must be accessible by the server.

The key should be either 128 bit (16 character) or 256 bit (32 character) long and in hexadecimal notation.

The outcome of the check command is either valid or invalid file.

Examples

Command with abbreviated parameters to check a file named `dist.xml`:

```
rmaCmdLine.sh -c -u myusername -p mypassword -d mydomain -f /full/path/name/to/file/dist.xml  
-k 3031323334353637
```

Command with full-format parameters, this time with a 256-bit key:

```
rmaCmdLine.sh --check --user myusername --password mypassword --domain mydomain --file  
/full/path/name/to/file/dist.xml --key 30313233343536373839414243444546
```

For more information about this command, and its parameters, refer to [RMA Server administration commands](#).

Importing an RMA distribution file

To import a distribution file, you use the same command as for checking an RMA distribution file, except that you use the action `--import` (or `-i`) instead of `--check` (or `-c`).

Enter the command:

```
rmaCmdLine.bat | rmaCmdLine.sh --import --user <user name> --password  
<password> --domain <domain> --file <distribution file path> --key  
<distribution file LAU>
```

As long as the file is valid, the RMA Server imports the distribution file after checking it.

Using the RMA filtering exit

Introduction

FEX provides an RMA filtering exit that is used in conjunction with Axway Gateway. The exit is used by Gateway to interface itself with the FEX RMA Server to provide RMA capability.

Important: To enable the RMA filtering user exit, make sure you select the option **Enabling RMA for Gateway** when you install the RMA Application.

Gateway uses the RMA filtering exit to make authorization requests to the RMA Server for transfers it receives or wants to send:

- When sending to a partner, RMA checking verifies whether there is a valid authorization to send a document to the partner
- When receiving from a partner, RMA checking verifies whether there is a valid authorization to receive a document from the partner

The user exit searches for a valid authorization in the Axway Financial Exchange authorizations data store.

Depending on the response of the RMA Server, the transfers will either proceed or be canceled. Transfers are not sent or received when valid authorizations are not found.

Prerequisites

Axway Gateway must be configured correctly for RMA (and ASP). Make sure that the path to the user exit library and the path to the RMA user exit configuration file are defined in Gateway. For details of how to configure Gateway, refer to the Axway Gateway online documentation.

Contents of the user exit configuration file

This section is provided for reference. The Axway Installer configures the exit when you install the RMA Application. The user exit configuration file contains the following parameters:

Parameter	Description
host	The machine name or IP address of the machine on which the server is running
port	The port on which the server is listening
username	The username that the command line will use to connect to the server
domain	The domain name to which the user belongs
password	The encrypted password corresponding to the user
timeout	The timeout (in seconds). The exit waits for this length of time before concluding that the server is not responding.
certificate_path	The full path containing the certificate required to authenticate connection with the RMA Server

Logging

Consult the Gateway log for log messages generated by the user exit.

RMA Server administration commands

This topic provides information about the RMA Server administration commands.

For more information about these commands, refer to the command help available in the command line interface (rmaCmdLine.bat --help or rmaCmdLine.sh --help).

All the RMA scripts are located in the `<FEX_RMA_INSTALL_DIR>/bin` directory. These scripts include the commands to administer the RMA Server.

Note: The command line must be installed together with the server on the same machine. The command line cannot be located on a different machine.

Introduction

The RMA command line is a client tool that enables you to act upon a running RMA Server. You use the command line to perform the following operations:

- Import a distribution file
- Check that a distribution file is valid before importing it
- Import an Application Service Profile (ASP)

Important: In order to perform actions on the server, you must have the required privilege. There is one privilege for each of the actions. For information about privileges, refer to [Configuring access control for FEX RMA](#).

Configuration file

The RMA command line configuration file `client.properties` is located in the `<FEX_RMA_INSTALL_DIR>/conf` directory. It contains parameters that enable the command line to locate the server, authenticate with it, and so on.

Parameter	Description
<code>server.host</code>	Name or IP address of the machine on which the server is running
<code>server.api.port</code>	Port on which the server is listening
<code>client.timeout.connection</code>	Number of milliseconds the client will wait when connecting before concluding that the server is not there
<code>client.timeout.receive</code>	Number of milliseconds the client will wait for a reply before considering that the server has taken too much time to perform an action
<code>passport.truststore.path</code>	Path to the trust store
<code>passport.truststore.type</code>	Type of trust store

Example of `client.properties` file

```
#####
## RMA Client configuration ##
#####
# Server host
```

```

server.host=localhost
# Webservice port number
server.api.port=9696
#####
# Client configuration #
#####
# timeout value in milliseconds for webservice connections to the
server
client.timeout.connection=60000
# timeout value in milliseconds for webservice response from the
server
client.timeout.receive=60000
#####
## Passport configuration ##
#####
passport.truststore.path=../fex-rma-server/conf/passport/passport_
truststore.jks
passport.truststore.type=JKS

```

Command syntax

The general syntax of an RMA command line:

OS	Command
Windows	<code>rmaCmdLine.bat --[action] --[parameter 1] -- [parameter 2] --[parameter n]</code>
UNIX	<code>rmaCmdLine.sh --[action] --[parameter 1] -- [parameter 2] --[parameter n]</code>

Note: You can specify parameters by their full name or their abbreviated name (short name). When using an abbreviated parameter name, use one dash before the parameter (instead of two dashes).

Examples:

```
rmaCmdLine.sh --check --user myname --password mypassword --domain mydomain --file
/full/path/name/to/file/dist.xml --key 30313233343536373839414243444546
```

```
rmaCmdLine.sh -c -u myname -p mypassword -d mydomain -f /full/path/name/to/file/dist.xml -k
3031323334353637
```

Command line actions

This table lists the RMA Server administration command line actions in their full and abbreviated forms.

Action	Command
Import ASP package	<pre>--importASP --user <user name> --password <password> --domain <domain> --file <ASP archive path> --key <ASP package digest> [-- digestAlgorithm <digest algorithm>]</pre> or <pre>-s -u <user name> -p <password> -d <domain> -f <ASP archive path> -k <ASP package digest> [-a <digest algorithm>]</pre>
Import a distribution file	<pre>--import --user <user name> --password <password> --domain <domain> --file <distribution file path> --key <distribution file LAU></pre> or <pre>-i -u <user name> -p <password> -d <domain> -f <distribution file path> -k <distribution file LAU></pre>
Check a distribution file without importing it	<pre>--check --user <user name> --password <password> --domain <domain> --file <distribution file path> --key <distribution file LAU></pre> or <pre>-c -u <user name> -p <password> -d <domain> -f <distribution file path> -k <distribution file LAU></pre>
Display help	<pre>--help</pre> or <pre>-h</pre>

Parameter list

Parameter long name	Parameter short name	Value(s)
--digestAlgorithm	-a	Digest algorithm Default: SHA-256
--domain	-d	Domain name

Parameter long name	Parameter short name	Value(s)
--file	-f	ASP archive path or distribution file path. Specify the full path name to the distribution file. The distribution file to import must be accessible by the server.
--help	-h	Display the command line help
--key	-k	ASP package digest or distribution file key. The ASP package digest is provided in SHA-256 in Base64 format. The distribution file key should be either 128 bit (16 character) or 256 bit (32 character) long and in hexadecimal notation.
--password	-p	The password corresponding to the user
--user	-u	User name

Command line errors

The command line returns an error code in the following cases:

- Authentication error – The user credentials have not been validated by PassPort AM through the RMA Server
- Privilege error – The user credentials are valid, but the user has no privilege to carry out this action
- Unable to perform operation – An error prevented the command line from executing the operation. See the on-screen explanation. For more details, consult the server log:
`logs/rma.log`.

Axway Gateway: Managing Transfers

Managing transfers

About transfers and transfer requests

Gateway processes file Transfers. A *file Transfer* is a set of processes that results in the exchange of files between two computers. In a transfer, one computer is the sender, the other is the receiver. The sender and receiver are linked together by a network. A file Transfer consists of sending:

- The file
- All related technical parameters associated with the file. For example: connected partners, local file storage format, and so on.

Transfer types

Gateway enables you to submit three types of Transfer:

- [Transfer Requests](#) – transfers a file between two computers
- [acknowledgment Message Requests](#) – returns an acknowledgment for a transfer
- [Application Message Requests](#) – enables you to send a limited-length text message via the PeSIT protocol

Transfer objects

Gateway provides a set of objects that enable you to create and define the characteristics of a Transfer. To create a Transfer you choose the objects useful for your Transfer and enter the object property values that define the processing that you require.

To execute a Transfer, you need:

- A *Transfer Request*
- A pair of *Site* objects

You may require additional objects to define additional transfer characteristics, depending on the nature of the Transfer and the context in which you want to execute it. For example, to handle any incoming connection request, you also need a *CGate* definition.

The following table lists the Gateway objects you can use to manage Transfers and summarizes the way you can use them.

Object	Description	Refer to
Application	To process files during transfer, Gateway requires information about the file characteristics. Application objects define: - Local file attributes, such as record format and length, local directory for received files and so on - Transferred file attributes, such as data code, record length, compression and so on You can enter values for certain Application object parameters directly in the Transfer Request. If you set the same parameters values in both the Transfer Request and the Application object, the Transfer Request parameter values override the Application parameter values.	About Applications
Connection Gate/ Connection Gate Group	Before Gateway can accept protocol connections from external entities, the access permissions and connection characteristics must be defined for the entity. The Connection Gate object enables you to attribute permission to connect to Gateway and exchange files with it. It also controls the connecting-client access to files stored in the Virtual File Directory.	About CGates and CGateGroups
Decision Rule	You can optionally configure Gateway to process the files it handles according to pre-set criteria. Decision Rules objects determine the processing for the current transfer based on a set of conditions that Gateway applies to the transfer. If all conditions specified in the Decision Rule match the parameters defined in the transfer, Gateway applies the specified execution type (for example, execute Perl script, redirect to Axway Integrator, and so on).	Managing Events on Gateway

Object	Description	Refer to
Diffusion List	Diffusion List objects enable you to transmit a Transfer Request to multiple destination Sites. You apply a Diffusion List object by first creating the Diffusion List object, and then selecting the Diffusion List object as a Transfer Request parameter.	About Diffusion Lists
Model	A Model is a template of a Transfer Request that groups a set of common parameters to apply to multiple Transfer Requests. Models accelerate and simplify the process of defining Transfer Requests. To use a Model, you: • Create the Model object • Select the Model object as a Transfer Request parameter You can enter values for certain Model object parameters directly in the Transfer Request. If you set the same parameters values in both the Transfer Request and the Model object, the Transfer Request parameters values override the Model parameter values.	About Models
Security	Security objects provide added security for user access and for TLS and SSH file transfers. Security objects include: • For user access security: ◦ User definitions and Profiles • For file transfers: ◦ Certificates and keys ◦ Network Profiles ◦ TLS and SSH Security Profiles	See Security guide for further information.

Object	Description	Refer to
Site	Every file exchange involves a sending and receiving partner. Site objects represent these partners. A Site object can represent a Local or Remote partner. Typically the Local Site object defines characteristics of the machine that hosts the Gateway. The Remote Site object defines the following parameters for the remote partner: • Protocol used for exchanges with this partner • Role: responder or initiator, file sender or receiver • Connection information (max number of connections to / from this Site, and so on) • Network type used for connections (TCP/IP, LU 6.2) • Network security information • Protocol-specific parameters Once you have created a Site you can select the Site as a transfer partner in a Transfer Request. If a Remote Site definition does not exist, when processing a Transfer Request, Gateway can create a dynamic Site.	About Site objects

Object	Description	Refer to
Transfer Request	To create a transfer with Gateway you create a Transfer Request. A Transfer Request defines the set of parameters required to submit a file transfer: • Name of file to transfer • Physical file attributes • Compression or encryption options • Behavior in case of transfer failure • Transfer scheduling, restarting and routing A Transfer Request can collect and reuse the parameters that you may have defined in other Gateway objects, such as the <i>Model</i>, the <i>Remote Site</i> and the <i>Application</i>. If you set values for the same parameters in both the Transfer Request and other objects, the Transfer Request parameter values override the other object parameter values.	Submitting Transfers

Groups

Most Gateway objects have a field called "group", that can be used as a namespace. Groups apply in a number of use cases:

- Mandatory: an application object used for an inbound transfer must be in the same group as its origin site object
- Mandatory: an application object used for an outbound transfer must be in the same group as its destination site object
- It is possible to give the same name to different applications, on condition they belong to different groups
- For ssh key objects, when performing authentication, a site can be used to represent a remote peer presenting distinct keys to authenticate with Gateway's SFTP server. In this case, all partners key can be grouped and the site having the protocol identifier equal to that peer login name would point to that group via the 'remote_public_key_group' attribute.

Axway Gateway: Managing Transfers

Submitting transfers

About transfers

You can use Gateway to create three types of transfer:

1. *File transfers*

The principle activity of Gateway is to handle the exchange of files between Sites. To execute a file transfer from Gateway, you create and submit a Transfer Request.

2. *Application message transfers*

When Gateway is configured to use the PeSIT protocol, you can send limited-length messages to a Remote Site. To do this, use the Application Message transfer function. This function enables you to send a short text message.

3. *Acknowledgement message transfers*

After receiving a File transfer from a Remote Site, you may need to return an acknowledgement message to the sending machine. To do this, create a message text and associate it with the local ID of the received Transfer Request. You can enter the acknowledgement text directly in the acknowledgement transfer, or alternatively, associate a file containing the message text. You can pre-program the date and time when the acknowledgement message is to be sent.

[Working with File Transfer Requests](#)

[Working with Application Messages](#)

[Working with Acknowledgement Messages](#)

[Working with Transfers \(command line\)](#)

[Transfers: Parameters List](#)

[Transfer icons](#)

[Transfer states in Gateway](#)

[Overview: File Transfers and Transfer Requests](#)

Axway Gateway: Managing Transfers

Transfer requests

Working with File Transfer Requests

[Submitting a Transfer Request](#)

[Managing records of Transfer Requests](#)

This topic describes how to use the GUI to create and manage Transfer Requests for the exchange of files between Sites.

Submitting a Transfer Request

Before you submit a Transfer Request, you need to create various objects:

- [Site objects](#), that are referenced by the Transfer object
- (Optionally) [Application](#), [Diffusion List](#) and [Model](#) objects, that enable you to submit Transfer Requests more easily

To submit a new Transfer Request:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click **Transfers**, and then select **New > Transfer Request** from the context menu.
Gateway opens the *New Transfer Request* window.
3. Complete the tabs of this window according to your transfer requirements.

The following topics describe the fields in each tab and their possible values:

- [General tab](#)
- [Attributes tab](#)
- [Details tab](#)
- [PeSIT tab](#) (only for PeSIT)
- [HTTP/FTP tab](#) (only for HTTP/FTP)
- [SMTP tab](#) (only for SMTP)
- [Trading Partner tab](#) (only for: S/MIME, EDIINT, RosettaNet)
- [SWIFTNet tab](#) (only for SWIFTNet)
- [JMS tab](#) (only for JMS)
- [OFTP tab](#) (only for OFTP)
- [X420 tab](#) (only for X.400/X.420)

For examples of sets of transfer parameter values, refer to [Transfer Examples](#).

4. After completing the tabs, click **OK**.

Gateway begins processing the Transfer Request, and returns a Transfer Request identifier: *local_ident*. This number serves as a local identifier for your Transfer Request. You can use the *local_ident* number in the GUI or in certain command lines when you need to identify the Transfer Request.

Managing records of Transfer Requests

Each time you submit a Transfer Request, Gateway assigns a local transfer ID number to the request and deposits a record of the Transfer Request to the Mailbox.

Use the Mailbox to monitor the status of transfers, gather information about successful and unsuccessful transfers and to re-initiate or abandon transfers according to your requirements.

To view Mailbox records of Transfer Requests:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click **Transfers**.

Gateway displays the Mailbox contents in the right pane.

[Working with Transfers \(command line\)](#)

[Transfers: Parameters List](#)

[Viewing and managing Mailbox contents](#)

[Transfer Examples](#)

Axway Gateway: Managing Transfers

New Transfer Request: General tab

Field	Description
General information	
Model	<i>Optional</i> To accelerate the creation of Transfer Requests, you can use a Model. The Model object groups a set of reusable parameters for submitting Transfer Requests. When setting parameters, note that the Transfer Request parameters override the Model parameters. Enter the name of the Model to use. Maximum: 25 alphanumeric characters. Alternatively, select an existing Model from the drop-down list.

Field	Description
Type	Select the type of transfer: • TRANS: Submit Transfer Request. • LOT <i>PEL protocol only</i>: After polling the Remote Site, Gateway creates the Transfer on the Local Site in <i>frozen</i> state. • POLL <i>PEL protocol only</i>: After polling the Remote Site, Gateway creates the Transfer on the Local Site in <i>To Begin</i> state. • LIST: Submit a Diffusion List type Transfer. • DIR <i>FTP, SFTP and HTTP Initiator mode only</i>: Gateway retrieves the contents of a given directory and saves the result in the file that you specify in the File component field. • MULTI <i>FTP and SFTP multiple transfers only</i>: Upload and download multiple files with a single request. This is an MGET or MPUT transfer (the distinction between the two is indicated by the Direction parameter). • INTERACT <i>SWIFTNet only</i>: Submit a SWIFTNet InterAct Transfer (equivalent to MSG Transfer type when using online commands). • (Empty): Gateway takes the Transfer type definition from the Application object first, and if undefined, from the Model object.
Mode	Select the Transfer connection mode: • Initiator • Responder • (Empty)
Direction	Select the Transfer direction: • Send • Receive • (Empty)

Field	Description
Application	<i>Optional</i> Select an existing Application. The Application defines file attributes to apply to the transferred file. Keep in mind that the Application that you select must belong to the same group as the Site that you specify in the Destination alias field. If you do not select an Application: • The Transfer takes the parameters that you explicitly specify in the Request • You must specify the Record format and the Record length (in the Attributes tab)
Ack option	<i>OFTP, PeSIT, SMTP and SWIFTNet only</i> Use this option to specify acknowledgment behavior for the Transfer. Select one of the following acknowledgment options: • Undefined (acknowledgment behavior undefined) • Not to ack (do not wait for acknowledgment) • To ack (wait for acknowledgment) For SMTP, refer to Configuring SMTP: Requesting an acknowledgment.
Purge option	To purge all files associated with the Transfer Request (via the pe1pur command), set this option to Yes. The value that you set here overrides the Purge option setting in the Model definition.
Permanent	This parameter controls the availability of files for download by remote clients when Gateway is acting in <i>server mode</i>. Usually (when Permanent = No), Gateway makes a specified file available only to the first client that submits a Transfer Request for the file. If you activate this option (Permanent = Yes), Gateway responds to each client's Transfer Requests by sending a copy of the requested file, keeping the original in the database available for subsequent client Transfer Requests. Select one of the following: • Yes (to set Transfer availability to permanent) • No • (Empty) The value that you set here overrides the setting in the Model definition.

Field	Description
Sites In the Sites section, complete either the first two fields or the last four fields. See the example below.	
Originator alias	Enter the Site alias of the source Site of the file transfer. Maximum: 31 alphanumeric characters.
Destination alias	Enter the Site alias of the file transfer destination. If you specify a Diffusion List in this field, Gateway transmits the Transfer to several destinations. Maximum: 31 alphanumeric characters.
Local agent alias	Enter the alias of the associated Local Site. Maximum: 31 alphanumeric characters.
Remote agent alias	Enter the alias of the associated Remote Site or Remote Site template. Maximum: 31 alphanumeric characters. Refer to About Site objects .
Originator	Enter the originator protocol identifier. Maximum: 31 alphanumeric characters. The name you enter in this field designates the source identifier of the Transfer.
Destination	Enter the destination protocol identifier. Maximum: 31 alphanumeric characters. The name you enter in this field designates the Transfer destination.
Site attributes – example PeSIT transfer from Gateway to an FTP client: • Local agent alias = GW • Remote agent alias = TFTP • Originator = Monitor (Local Site alias, originating transfer) • Destination = User (Remote Site alias, to receive transfer)	

New Transfer Request: Attributes tab

Use the **Attributes** tab to define the properties of the transferred file.

Many of the fields on this tab correspond to Application object parameters. If you select an Application to use for this Transfer, you do not need to complete the fields on this tab. If you select an Application **and** complete fields here, the Transfer parameters override the Application parameters.

Field	Description
Local attributes	
File component	For files to be sent, enter the local file name (without the file path) of the physical file to send. For files to be received, enter the physical name of the incoming file. Maximum: 127 alphanumeric characters.
Record format	Select the record format for the local file: • Variable text • Variable • Fixed text • Fixed • Stream • Undefined • (Empty)
Record length	Specify the record length. Integer: 0 - 32768.
Data code	Select the data code for the local file.
Transfer attributes	

Field	Description
Directory name	<i>FTP, SFTP and HTTP only</i> Enter the absolute path of the directory that contains the file to transfer. This location is the destination of the file in the case of outgoing Transfers, or the path for an incoming Transfer. If you specified: • Initiator mode on the General tab, specify a directory on the remote system • Responder mode on the General tab, click the Browse button to select a directory in the VFD • DIR in the Type field on the General tab, specify the directory whose contents you want to list. For FTP Transfers, you must add the file separator / at the end of the directory name. (The separator character is added automatically for HTTP Transfers.) Maximum: 127 alphanumeric characters.
Directory alias	<i>FTP, SFTP and HTTP only</i> This field is only available if you selected Responder in the Mode field on the General tab. Enter the alias of the VFD directory that contains the file to transfer. Maximum: 24 alphanumeric characters.
File name	The value that you enter in this field depends on the Transfer type. If you selected: • DIR in the Type field on the General tab, this field is valid only for Gateway-to-Gateway transfers and it is optional. Use this field to define a filter on the file names returned by the server. For example, if you set this value to *.txt, the server returns all .txt files in the directory that you specify in the Directory name field. If you do not fill in this field, Gateway does not set a default value. For DIR transfers to servers different from Gateway, this field should be left blank. • Anything else, enter a file name for the transferred file. This file name is transmitted to the Transfer recipient. Maximum: 127 alphanumeric characters.

Field	Description
File type	<i>PeSIT only</i> Enter the protocol file type. 1 - 65535. Reserved values: • 65535 = MSG • 65534 = EERP
Record format	Select the record format for the transferred file: • Fixed • Variable • Stream • Text (<i>OFTP only</i>) • Undefined • (Empty)
Record length	Specify the record length for the transferred file. Integer: 0 - 32768.
Data code	Select the data code for the transferred file.
Compression	This option allows you to specify the compression options for the transferred file. Select one of the following: • No • Horizontal • Vertical • Both • Zlib • (Empty)

Field	Description
Newline convention	<i>SFTP only</i> Set the end-of-line code, depending on the remote system type. This parameter is valid for text file transfers in both send and receive mode. Although this parameter is also available in the Remote Site definition, the Transfer Request parameter takes priority. Enter one of the following values: • U for UNIX • W for Windows • Blank for Undefined
Format	
Padding character (Hexa)	Specify the padding character code. You can enter two characters, from 00 to FF. Commonly-used value: 20.
Truncating allowed	If the Transfer uses the truncating feature, this option allows you to truncate a record without taking into account the padding character.
Multi transfer parameters	

Field	Description
Filter	<i>FTP and SFTP multiple transfers only</i> Specify the filtering criteria to select files to send or receive. You can filter according to: • Time: <pre>time = [last rounded] N [day[s] hour[s] minute [s]] [yyyyymmdd <=] time [<= yyyyymmddss] [yyyyymmddhhmmss <=] time [<= yyyyymmddhhmmss]</pre> For example, if it is currently 2007/06/13 17:15:28, then: <pre>last1day = 2007/06/12 17:15:28 rounded1day = 2007/06/13 00:00:00 last3days = 2007/06/10 17:15:28 rounded3days = 2007/06/11 00:00:00 last1hours = 2007/06/12 16:15:28 rounded1hours = 2007/06/12 17:00:00</pre> • File size: <pre>[min <] size [< max]</pre> Specify the file size in bytes. • File name: <pre>name = regexp (use StringMatch)</pre> Example: <pre>"20070528 <= time <= 20070609 ; 1 < size < 100000 ; name = *.dat"</pre>

Field	Description
Hash type	<i>FTP and SFTP multiple transfers only</i> Select the type of hashing to use. This parameter determines whether all files in a directory will be transferred a second time or only those files that have changed since the initial transfer. This is useful if a transfer has been interrupted or only updated files need to be sent. Possible values: • 0: No hash. No check of duplicate file transfers is made. All files are transferred again. • 1: (default) Gateway checks the following criteria to determine if a file has changed since the initial transfer: ◦ Alias of the partner ◦ Direction ◦ File size ◦ File date ◦ Directory name of the master ◦ File name (short for MGET, long for MPUT) If none of these criteria has changed, the file is considered as unchanged and is therefore not sent. • 2: <i>For future use</i> • 3: <i>For future use</i>
Model to use for child transfers	<i>FTP and SFTP multiple transfers only</i> Select the name of the Model to use to deposit <i>child</i> transfers. If this parameter is not filled-in, the Transfer Request for a <i>child</i> is created using the Application of the <i>master</i>. In all cases, a Transfer Request for a <i>child</i> is made with Type = TRANS and Mode, Direction, Originator alias and Destination alias derived from the <i>master</i>.

Axway Gateway: Managing Transfers

New Transfer Request: Details tab

Field	Description
User	

Field	Description
Param 1	Enter the user parameter 1. Maximum: 40 alphanumeric characters. Gateway uses the values you enter in this parameter locally. It does not transfer the values to your transfer partner. Use this parameter to locally store values that you can reuse via Decision Rules, Models and batch procedures.
Param 2	Enter the user parameter 2. Maximum: 40 alphanumeric characters. Gateway uses the values you enter in this parameter locally. It does not transfer the values to your transfer partner. Use this parameter to locally store values that you can reuse via Decision Rules, Models and batch procedures.

Field	Description
Message to send	Enter a user message. • FTP: User message to transmit at the connection phase. This message can be a list of special commands to send to the remote server using <code>SITE</code> or <code>QUOTE</code> FTP commands. • HTTP: User message to transmit at the connection phase. The user message enables you to add CGI parameters or non-standard HTTP headers to the Transfer Request. For CGI parameters you must also check the Send transfer attributes option on the HTTP tab in the Remote Site. Use the format: <code>xfer_param1=value1&xfer_param2=value2</code>. For non-standard HTTP headers, use the format: <code>[xs4hh]X-Id1=v1&X-Id2=v2[/xs4hh]</code>. • OFTP, PEL, PeSIT, EDIINT: User message to transmit at the connection phase. • SWIFTNet FileAct: [0 - 255]: Transfer description, supplied by the initiator. Free-text description of the transfer. [256 - 511]: Transfer information, supplied by the initiator. Structured information concerning the transfer. For more details, refer to the SWIFT documentation. • SWIFTNet InterAct: [0 - 511]: User message Maximum: • 8 alphanumeric characters (OFTP) • 256 alphanumeric characters for each part (SWIFTNet FileAct) • 512 alphanumeric characters (FTP, HTTP, PeSIT, EDIINT, SWIFTNet InterAct)
Retry and Timing	
Interval	Depending on the protocol, complete this field as follows: • PEL: specify the interval between two Remote Site polls (LOTS and POLL commands) • PeSIT HS: for cyclic selection requests only - in case of selection failure, specify the repetition interval in seconds

Field	Description
Maximum number of retries	Enter the maximum number of times Gateway tries to submit the Transfer Request. Integer.
Priority	Select the Transfer priority: • Normal • High • Low • (Empty)
Frozen state	This option allows you to create the Transfer in <i>frozen state</i>. Select one of the following: • Yes • No • (Empty) The value that you set here overrides the value that you set in the Model definition. After creating a Transfer in frozen state, it remains available for submission at a later date. To submit a Transfer that is in frozen state, select Restart a transfer. Alternatively, use the command <code>pelctl control_trans -a R</code> Note: it is not possible to create a Diffusion list transfer in frozen state.
Earliest start date	
Date	Click the check-box to activate the field. Then click the day, month or year units to select them and enter the earliest date to begin the transfer. Alternatively, click the down arrow to open a calendar window from which you can select the date.
Time	Click the check-box to activate the field. Then click the hours, minutes or seconds units to select them, and enter the earliest time to begin the transfer. Alternatively, click the unit and then click the up and down arrows to select the time.
Latest start date	

Field	Description
Date	Select the latest date to start the transfer. Use this field to define the validity period for the transfer. If the transfer has not taken place or is in progress when this period expires, it is considered timed out and is canceled. Click the check-box to activate the field. Then click the day, month or year units to select them and enter the latest valid date for the transfer. Alternatively, click the down arrow to open a calendar window from which you can select the date.
Time	Select the latest time to start the transfer. Click the check-box to activate the field. Then click the hours, minutes or seconds units to select them, and enter the latest valid time to begin the transfer. Alternatively, click the unit and then click the up and down arrows to select the time.
Routing/Reply	
Route from (Transfer ID)	Enter the local identifier of the Request from which the Transfer is routed.
Reply to (Transfer ID)	<i>acknowledgment Requests only</i> Enter the local identifier of the acknowledged Transfer Request.
Sentinel	
Transfer filter	Use this option to send a record of Transfer state changes to Sentinel. Select one of the following: • All: Sends all Transfer state changes to Sentinel • Summary: Sends a summary of Transfer state changes. Summary Transfer states are: canceled, ended, acked, created, to begin. • None: Sends no Transfer state changes • Undefined: Gateway searches for the definition in the Application, then the Remote Site. If the Sentinel Transfer Filter field is set to Undefined in both of these objects, Gateway uses the value set in the configuration menu.

New Transfer Request: PeSIT tab

Field	Description
Negative ack option	The Nack (negative acknowledgment) option indicates whether the Remote Site supports the negative EERP extension or not. Select: • Nack by user: negative acknowledgments are accepted • None: all negative acknowledgment requests are refused
File label	<i>Only applicable to PeSIT protocol PI 37</i> Enter the protocol label for the file. Maximum: 256 alphanumeric characters.
Generic reception	
Maximum number of requests	Specify the maximum number of Requests allowed. 0 - 32000. 1 = the number of Requests is unlimited.
Maximum number of files	Specify the maximum number of files allowed. 0 - 32000. 1 = the number of files is unlimited.
Send (Store-and-Forward mode only)	
User	Enter the sender User name. Maximum: 8 alphanumeric characters.
Application	Enter the sender Application name. Maximum: 8 alphanumeric characters.
Text	Enter free text. Maximum: 8 alphanumeric characters.
Receive (Store-and-Forward mode only)	
User	Enter the receiver User name. Maximum: 8 alphanumeric characters.
Application	Enter the receiver Application name. Maximum: 8 alphanumeric characters.
Text	Enter free text. Maximum: 8 alphanumeric characters.

New Transfer Request: HTTP/FTP tab

Field	Description
HTTP CGI parameters/ FTP user command	Depending on the protocol, complete this field as follows: FTP: Send special remote commands or unsupported FTP commands to the remote server before or after the Transfer. <code>SITE</code>, <code>QUOTE</code> and <code>SCRIPT</code> commands are supported. <code>SITE</code> and <code>QUOTE</code> commands indicate pre-Transfer processing. For post-Transfer processing, use <code>AE_QUOTE</code> and <code>AE_SITE</code> (ended state if errors occur) or <code>AC_QUOTE</code> and <code>AC_SITE</code> (canceled state if errors occur). Enter the command in the following format: <code>SITE <fct1> <param1>; SITE <fct2> <param2></code> <code>SCRIPT</code> commands reference a script that contains a set of FTP (<code>SITE</code> and <code>QUOTE</code>) commands: <code>SCRIPT <script_xfer></code> Refer to Calling a script. HTTP: Add CGI parameters to the Transfer Request in the following format: <code>xfer_param1=value1&xfer_param2=value2</code> SWIFTNet InterAct: Specify either: - Payload attributes in the following format: <code>attrib1=\"value1\"</code> - Compression information in the following format: <code>dt=\"GZIP-Base64\"</code>
FTP	
Unique file name option	FTP store unique option (<code>STOU</code> command). Use this option to assign a unique file name on the server. If a previous file with the same name exists on the server, a ".1" suffix is added to the file name. If a file name with this suffix already exists, the suffix is replaced with ".2" and so on until a unique name is found. The <code>STOU</code> command is supported in both real and virtual directories.

Field	Description
Append option	Use this option on the client side in sender mode. This option allows you to add data to the end of a received file. To enable the append option, select Yes . Otherwise, select No .
Directory listing type	<i>DIR Transfer types in Initiator/Receiver mode only.</i> Use this field to specify how to list the directory contents. Select one of the following: • Names (default): the <code>NLIST</code> command returns a basic list of directory contents. This option does not differentiate between files and directories, and provides no extra information such as the last modification date, the file size, and so on. • Long: the <code>LIST</code> command returns a detailed list of directory contents. For example, this option differentiates between files and directories. The LONG format depends on the server and is typically in a UNIX format.
HTTP	
Request method	Specify the method to use for HTTP Transfers: • Get or Post for incoming data • Post or Put for outgoing data

Axway Gateway: Managing Transfers

New Transfer Request: OFTP tab

Use the **OFTP** tab to define OFTP-specific properties for the Transfer Request.

Field	Description
Signed EERP	Is signed EERP required? Select Yes or No .
Virtual file description	Virtual file description. (free text)

Axway Gateway: Managing Transfers

New Transfer Request: SMTP tab

Field	Description
email subject	Enter the email subject. Maximum: 128 alphanumeric characters. This field is mandatory if Gateway must create the email. If you are sending a formatted email via a Gateway Transfer Request, do not complete this field manually. To transmit parameters between two Gateway monitors, use the MIME header Subject field.
Attach file	Select one of the following values: • Yes = the file is not formatted as an email: Gateway creates an email and attaches the file • No = the file is already formatted as an email: Gateway does not attach a file • Undefined = Gateway must open the file to determine how to proceed
Add Gateway parameters to subject	Set this option to Yes to add the following Gateway-specific parameters to the email subject: • destination (-dest) • origin (-org) • protocol file name (-file_name) • application (-appli) Use the following syntax: <pre>Subject: //XFB appli=appliName; DEST=destId; ORG=orgId FILENAME=fileName</pre>
From	This field is mandatory if you set the value of Attach file to Yes. If Attach file is set to No, the From field is already completed in the formatted email and you do not need to complete this field. Enter the return email address. Maximum: 128 alphanumeric characters. If you do not specify an originating address in this field, Gateway uses the default email address that you specify in the Local Site associated with the Remote Site.

Field	Description
To	This field is mandatory if you set the value of Attach file to Yes. If Attach file is set to No, the To field is already completed in the formatted email and you do not need to complete this field. Enter the main destination email address. There is no limitation on the number of characters you can enter in this field. If you do not specify a destination address, the default email address specified in the Remote Site is used. If the length of the field is not large enough to contain the corresponding list, it is completed by " * " and the value is stored in an overflow block. The overflow blocks are formatted as a character string. For example: To: (options) "description" <address@email>, ... \0 Cc: ... \0 Bcc: ... \0 \0
Cc	Enter the carbon copy destination addresses. There is no limitation on the number of characters that you can enter in this field.
Bcc	Enter the blind copy destination addresses. There is no limitation on the number of characters that you can enter in this field.

Axway Gateway: Managing Transfers

New Transfer Request: Trading Partner tab

Only use this tab for: S/MIME, EDIINT, RosettaNet or OFTP R2.0.

Field	Description
Destination partner	Enter the Trading Partner name that is the file destination.
Originator partner	Enter the Trading Partner name that is the file originator.

Field	Description
Format	Select the trading format: • SMIME • AS1 • AS2 • AS3 • RosettaNet • CMS (for OFTP R2.0)
Is signing	Select the signing option: • Yes • No • (empty) = use the value set in the destination Trading Partner
Is compressing	Select the compressing option: • Yes • No • (empty) = use the value set in the destination Trading Partner
Is encrypting	Select the encryption (ciphering) method: • Yes • No • (empty) = use the value set in the destination Trading Partner
Receipt request	Select the receipt request option: • Signed • Unsigned • None • Undefined = use the value set in the destination Trading Partner Note: Only the S/MIME protocol allows the sending of a signed receipt.
Synchronize receipt	<i>Only available for AS2 transfers if acknowledgment request is set to Signed or Unsigned.</i> Select this option if a synchronized receipt must be requested.

Field	Description
Receipt url/email	<i>Not Available for S/MIME protocol</i> Enter the receipt response target address. If you do not complete this field, the default value is the originator name.
Content-type	Enter the content type to use when sending an EDI or S/MIME file. For example: • application/EDIFACT • application/XML • text/plain

Axway Gateway: Managing Transfers

New Transfer Request: SWIFTNet tabs

Field	Description
Destination partner	Enter the Trading Partner alias that represents the file destination.
Delivery mode	Select the delivery mode for the file or message: • Undefined (default) • Real time = Immediate • Store and forward = Send and then store in the queue in the SWIFTNet domain (SnF)
Destination DN	Enter the name of the destination partner of the trading partner exchange. Maximum: 128 alphanumeric characters.
Compress	Select the compressing option: • Yes = Use the compression algorithm when sending a file or message to a partner • No = No compression • (empty) = Use the value set in the destination Trading Partner
Originator partner	Enter the Trading Partner alias that is the file originator.

Field	Description
Originator DN	Enter the name of the originating partner of the trading partner exchange. Maximum: 128 alphanumeric characters.
Service name	Enter the SWIFT business service name (as specified in the XML header). Maximum: 30 alphanumeric characters. If you do not complete this field, Gateway uses the value set in the destination Trading Partner.
Request type	Enter the type of request (as specified in the XML header). Maximum: 30 alphanumeric characters. If you do not complete this field, Gateway uses the value set in the destination Trading Partner.
Receipt sub-tab	
Receipt request	Select the receipt request option: • Signed = Request receipt • None = No receipt • Undefined = Use the value set in the destination Trading Partner
Receipt DN	<i>FileAct, Real Time mode only, if you have selected Request receipt</i> DN = Distinguished Name Enter the name of the receipt reception address. This is the address to which the receiving partner sends the receipt. If you do not complete this field, the default value is the originator name.
Receipt request type	<i>FileAct, Real Time mode only, if you have selected Request receipt</i> Type of acknowledgment request Maximum: 30 alphanumeric characters.
E2E (End To End) sub-tab	

Field	Description
Message id	The value of this parameter overwrites the message id created by Gateway in the Transfer Request. Enter the SWIFTNet message id of the new transfer. This should be a Universal Unique IDentifier (UUID). If the transfer is a Possible Duplicate, enter the message id of the initial transfer. Maximum: 40 alphanumeric characters.
Possible duplicate	Select this option to indicate to the responder that the transfer is a Possible Duplicate.
Header info	Reference to file on disk containing Header Info. This file must respect a specific structure.
Security sub-tab	
Sign	Select the signing option: • Yes • No • (empty) = Use the value set in the destination Trading Partner
Encrypt	<i>InterAct only</i> Select the encryption method: • Yes • No • (empty) = use the value set in the destination Trading Partner
Non repudiation	Select this option to request SWIFT to keep proof of sending (availability depends on the service used).
Sign DN	<i>InterAct only</i> If specified, name used for signing in Initiator mode. Otherwise the securityDN (supplied by SNL) is used. Maximum: 100 alphanumeric characters.
Encrypt DN	<i>InterAct only</i> If specified, name used for encryption in Initiator mode. Maximum: 100 alphanumeric characters.
Signature list	Reference to file on disk containing a Signature List. This file must respect a specific structure.

Field	Description
Other sub-tab	
Additional params	Reference to file on disk containing additional SWIFTNet structures (for example, Header Info and/or Signature List). This file must respect a specific structure .
InterAct sub-tab	
Send text message	Select this option to send a text message in InterAct. Enter the message in the text frame. Maximum: 512 alphanumeric characters.
Payload type	Select the payload format: • Undefined • Text • Embedded On the <i>Initiator</i> side, you can use this parameter to overwrite the Payload type parameter defined in the Trading Partner. On the <i>Responder</i> side, when using the Receipt request synchronized option you can respond with a custom payload type. Even if you saved the request payload as embedded in the Trading Partner, you can answer back with text mode.
SnF sub-tab [for Store and forward Delivery mode]	
Notification queue	Name of the SWIFTNet queue for receiver delivery notifications for the transfer.
Overdue time	
Date	SWIFTNet OverdueTime To set a date value for the OverdueTime, check this option and select the required date using the drop-down calendar. When the option is unchecked, no OverdueTime will be sent.

Field	Description
Time	SWIFTNet OverdueTime To set a time value for the OverdueTime, check this option and set the required hours and minutes (the seconds are truncated to '00' when sending). When the option is unchecked, but Date is checked, the OverdueTime sent is the date with time '00:00:00'. When both Time and Date are unchecked, no OverdueTime will be sent. Note: If you use the Time option, you must also set the Date option.
Copy services	
Request copy	SWIFTNet File or Message Copy service indicator. Select this option to request a copy of the file to be sent to a third party defined at the service level.
Request authorisation notification	SWIFTNet Authorization notification indicator. Select this option to receive Authorization system messages for files sent over Y-copy services.

Axway Gateway: Managing Transfers

New Transfer Request: JMS tab

Use the **JMS** tab to define JMS properties that are added to the JMS message during the transfer.

Field	Description
Transfer properties	
This section displays the list of properties that are added to the JMS message during the transfer. Modify the list according to your transfer requirements.	
Property	Property name
Value	Property value
New	Click New to add a property. Then enter the Property name and Property value.
Delete	Click Delete to remove a property.

Note: To add the same properties for use with many transfers, use the [Model object](#).

New Transfer Request: X.420 tab

Use the **X.420** tab to define X.420-specific properties for the Transfer Request.

Field	Description
Originator	Override the originator X.400 address. Click Edit... to configure the O/R address . Click Clear to delete the displayed O/R address.
Recipient	Override the recipient X.400 address. Click Edit... to configure the O/R address . Click Clear to delete the displayed O/R address.
Delivery notification request	Select the type of delivery notification request: • NDN = Non Delivery Notification (default) • DN_NDN = Delivery Notification and Non Delivery Notification
Notification request	Select the type of receipt request: • None (default) • NRN = Non Receipt Notification • RN_NRN = Receipt Notification and Non Receipt Notification
Interpersonal messaging Id	Define the IPM (Interpersonal Messaging) Identifier that uniquely identifies this IPM. Mandatory user relative identifier, which uniquely identifies the IPM.
Subject	Subject of the message.
Body part type	Select the body part type to be used: • ia5 (default) • g3facsimile • teletex • videotex • encrypted • bilaterallyDefined • externallyDefined

Field	Description
Importance	Specify the importance of the IPM: • low • normal (default) • high
Sensitivity	Specify the sensitivity of the IPM: • unspecified (default) • personal • private • companyConfidential
Content Id	Uniquely defines the message to be sent.
Message transfer system priority	Define the priority level for the message: • High • Normal • Low
Reply request	Select whether or not to request a reply: • N (default) • Y
Replied to IPM Id	Set the IPM identifier which identifies the message to which this message is the reply.

Axway Gateway: Configuration

Configuring X.400 O/R address

The *O/R Address* window is displayed if you click **Edit...** in:

- The configuration menu
- Remote Site
- Transfer Request

Message originators and recipients within X.400 are identified by their O/R address. The O/R address consists of a number of mandatory and optional address attributes. These attributes are organized hierarchically.

Specify the O/R address by modifying the address attributes in the *X.400 O/R Address* window. Only use [valid characters](#) in the O/R address, as listed in this topic.

Note: Not all fields are used for all objects. In this case they are grayed out.

General tab

Field	Description
Country (C)	Identifies the Country.
ADMD (A)	Administration Domain Name. Identifies an ADMD relative to the country specified by the Country Name. An ADMD provides a public messaging service within a single country. Several ADMDs may exist within a Country.
PRMD (P)	Private Domain Name. Identifies a PRMD relative to the ADMD or the Country. A PRMD is a messaging service within an organization. The ADMD operator must uniquely assign the Name of the Private Domain if the PRMD is connected to an ADMD service.
Organization (O)	Identifies an organization relative to the PRMD.
Org. Unit 1 (OU1)	Organization Unit Name 1-4. Can be used up to four times to identify divisions/departments within the organization.
Org. Unit 2 (OU2)	The names appear in order of importance. Always start with Org. Unit 1. You cannot enter an Org. Unit 2 unless you enter an Org. Unit 1 first, and so on.
Org. Unit 3 (OU3)	
Org. Unit 4 (OU4)	
Common name (CN)	Identifies a user, an application, or a distribution list. Caution: Common name is not supported in MTA version 84. You may encounter problems if your MTA is routed via a version 84 MTA.

Personal tab

Field	Description
Surname (S)	The Surname is a mandatory part of the Personal Name.
Given name (G)	The Given Name is an optional part of the Personal Name.
Initials (I)	The Initials are an optional part of the Personal Name.
Generation Qual (Q)	The Generation qualifier is an optional part of the Personal Name.

Domain defined tab

Field	Description
Type (DDA)	The Domain-defined attributes are specific to a messaging domain and are used to pass non-standard information.
Value	These attributes are used in addition to those that specify the user/application.

Additional tab

Field	Description
Free form name (FFN)	Free text.
Telephone number (TPH)	Telephone number of Originator/Recipient.

Valid characters in O/R addresses

Character description	Character
Uppercase alphabetic characters	A, B, C, ... Z
Lowercase alphabetic characters	a, b, c, ... z

Character description	Character
Digits	0, 1, 2, ... 9
Space	\ '
Apostrophe	'
Left and right parentheses	()
Plus sign	+
Comma and period	, .
Slash	/
Dash	-
Colon	:
Equal sign	=
Question mark	?

[About X.400](#)

[X.400 connector](#)

[Configuration: Connectors](#)

[New Remote Site: X.400 tab](#)

[New Transfer Request: X.420 tab](#)

Axway Gateway: Managing Transfers

Application messages

Working with Application Messages (PeSIT only)

Submitting Application Message Requests

To create and transfer an Application Message to a remote machine:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click **Transfers**, and then select **New > Application Message** from the context menu.
Gateway opens the *New application message* window.
3. Complete the tabs of this window.
The following topics describe the fields in each tab and their possible values:

- **General** tab
- **Options** tab

Use this tab to specify a date and time, or a range of dates and times, when the message can be sent. If you leave all of these fields empty, Gateway sends the message immediately.

4. After completing the tabs, click **OK** to validate and initiate the transfer.

If the parameters are correct, Gateway processes the transfer and deposits a transfer record to the Mailboxes of the local and remote machines.

[Working with Transfers \(command line\)](#)

[Transfers: Parameters List](#)

Axway Gateway: Managing Transfers

New Application Message: General tab

Field	Description
Originator Site alias	<i>Optional</i> If you do not enter an Originator Site alias, Gateway automatically provides information from your Local Site definition. Enter the alias of the Site that is the source of the file transfer. Maximum: 31 alphanumeric characters. Alternatively, you can select an originator Site from the drop-down list.

Field	Description
Destination Site alias	Enter the alias of the Site that is the file transfer destination. If you specify a Diffusion List in this field, Gateway submits the Transfer to several destinations. Maximum: 31 alphanumeric characters. Alternatively, select a destination Site from the drop-down list. Destination Sites must support PeSIT. The predefined PeSIT Sites delivered with Gateway by default are: • TPHSD • TPHSE • PHSD • PHSE
File Name	Enter a name to identify the Message transfer. After the transfer operation, Gateway displays this name in the File Name column of the Mailbox record for the transfer.
Send text message in file	
Send text message in file	If the message to send is stored in an external file, select this option. Then complete the following fields.
Message file path	Enter the complete file and path of the external file that contains the message to send.
Application	Select a local Application to read the file specified in the Message file path field. Alternatively, enter the Application name manually. Maximum: 25 alphanumeric characters.
Send text message	
Send text message	If the message to send is not contained in an external file, select this option. Then complete the following fields.
Message text	Enter the text of the message to send directly in the text field. Maximum: 512 characters.
Message data code	Select the local data code for the message.
Transfer data code	Select the data code for the transferred message.

[Working with Application Messages](#)

[Transfers: Parameters List](#)

Options tab

The **Options** tab is identical for both **Application Message Requests** and **Acknowledgement Message Requests**.

Use this tab to specify a date and time, or a range of dates and times, when the message can be sent. If you leave all of these fields empty, Gateway sends the message immediately.

Field	Description
Timing	
Earliest start date	Use this field to defer the start date for the Transfer. Select the earliest start date for the Transfer.
time	Select the earliest start time for the Transfer.
Relative time	The time that you specify here is relative to the current time. Specify the time in the format: HH:MM .
Latest start date	Use this field to set a time limit on the Transfer. Select the latest date to start the Transfer.
time	Select the latest time to start the Transfer.
Relative time	Time relative to the current time. Specify the time in the format: HH:MM . For example, enter 01:10 to indicate one hour and ten minutes from the present time.
Maximum number of restarts	Enter the maximum number of Transfer restarts. Integer: 1 - 99 .
Priority	Select one of the following Transfer priority values: • High • Normal (default) • Low
User information	
Param 1	Use these two fields to enter user parameters. Maximum 40 alphanumeric characters in each field.
Param 2	

[Working with Application Messages](#)[Working with Acknowledgement Messages](#)[Transfers: Parameters List](#)

Axway Gateway: Managing Transfers

Acknowledgment messages

Working with acknowledgment Messages

Submitting acknowledgment Message Requests

To create and transfer an acknowledgment Message:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click **Transfers**, and then select **New > acknowledgment Message** from the context menu.

Gateway opens the *New acknowledgment message* window.

3. Complete the tabs of this window:

- [General](#) tab
- [Options](#) tab

Use this tab to specify a date and time, or a range of dates and times, when the message can be sent. If you leave all of these fields empty, Gateway sends the message immediately.

4. After completing the tabs, click **OK** to validate and initiate the transfer.

If the parameters are correct, Gateway processes the transfer and deposits a transfer record to the Mailbox.

Axway Gateway: Managing Transfers

New Acknowledgment Message: General tab

Field	Description
Transfer ID to ack	Enter the identifier of the Transfer that is being acknowledged.

Field	Description
Ack type	Select one of the following acknowledgement options: • EERP • RECEIPT for EDIINT Trading Partner and SWIFTNet exchanges
Negative acknowledgement	Select this option if you want to receive a negative acknowledgement from the Remote Site in the event of transfer failure. This option is valid only if the Remote Site negative acknowledgement is activated for the Remote Site.
Send text message in file	
Send text message in file	If the message to send is stored in an external file, select this option. Then complete the following fields.
Message file path	Enter the complete file and path of the external file that contains the message to send.
Application	Select an Application from the drop-down list. Alternatively, enter the Application name manually. Maximum: 25 alphanumeric characters.
Send text message	
Send text message	If the message to send is not contained in an external file, select this option. Then complete the following fields.
Message text	Enter the text of the message to send directly in the text field.
Message data code	Select the data code for the message.
Transfer data code	Select the data code for the transferred message.
SMTP/POP3 only	
SMTP Site	Select an SMTP Site.
From	Enter the SMTP From address. Maximum: 128 alphanumeric characters. If you do not specify an originating address in this field, Gateway uses the default email address that you specify in the Local Site associated with the Remote Site.

[Working with Acknowledgement Messages](#)

[Transfers: Parameters List](#)

Options tab

The **Options** tab is identical for both **Application Message Requests** and **Acknowledgement Message Requests**.

Use this tab to specify a date and time, or a range of dates and times, when the message can be sent. If you leave all of these fields empty, Gateway sends the message immediately.

Field	Description
Timing	
Earliest start date	Use this field to defer the start date for the Transfer. Select the earliest start date for the Transfer.
time	Select the earliest start time for the Transfer.
Relative time	The time that you specify here is relative to the current time. Specify the time in the format: HH:MM .
Latest start date	Use this field to set a time limit on the Transfer. Select the latest date to start the Transfer.
time	Select the latest time to start the Transfer.
Relative time	Time relative to the current time. Specify the time in the format: HH:MM . For example, enter 01:10 to indicate one hour and ten minutes from the present time.
Maximum number of restarts	Enter the maximum number of Transfer restarts. Integer: 1 - 99 .
Priority	Select one of the following Transfer priority values: • High • Normal (default) • Low
User information	

Field	Description
Param 1	Use these two fields to enter user parameters.
Param 2	Maximum 40 alphanumeric characters in each field.

[Working with Application Messages](#)

[Working with Acknowledgement Messages](#)

[Transfers: Parameters List](#)

Transfer icons

This topic describes the set of icons associated with Mailbox Transfer Request records in the GUI.

[Transfer record list](#) (in the right pane)

[Viewing a transfer](#) (in read-only mode in the Transfer Request record window)

Transfer record list

In the transfer record list displayed in the right pane, each line represents the record of a transfer. Each line, or record, contains an icon that indicates the following information about the nature and state of the transfer:

- Mode (Initiator or Responder)
- Direction (send or receive)
- Transfer state
- Use of Diffusion List
- Routing state

Each icon includes a blue ellipse that represents Gateway.

Transfer mode

If the Gateway symbol is positioned to the left of the arrow, the transfer mode is **Initiator**, otherwise the transfer mode is **Responder**:

	• mode: Initiator • direction: send • state: ENDED

	- mode: Responder - direction: send - state: ENDED

Direction

The transfer direction is indicated by an arrow. If the arrow points towards the Gateway symbol, the direction is **receive**, otherwise the direction is **send**.

	- mode: Initiator - direction: send - state: ENDED
	- mode: Initiator - direction: receive - state: ENDED

Transfer states

The color of the arrow indicates the transfer state:

- black**: To begin, To restart, To cancel, To suspend, To sign, Delayed, Frozen
- yellow**: Beginning, In progress, Servicing, Canceling, Suspending, Created
- white**: Deleted
- green**: Ended, Acked, Ended to ack
- red**: Suspended, Canceled, Interrupted
- purple**: Nack by user, Nack by XFB

	- mode: Initiator - direction: send - state: FROZEN, TO_BEGIN or TO_RESTART
	- mode: Initiator - direction: send - state: BEGINNING, IN PROGRESS, SERVICING, CANCELING, SUSPENDING, CREATED
	- mode: Responder - direction: receive - state: DELETED

	• mode: Initiator • direction: send • state: ENDED
	• mode: Initiator • direction: send • state: CANCELED, INTERRUPTED or SUSPENDED
	• mode: Initiator • direction: send • state: NACK BY USER, NACK BY XFB

Note: On receiver side an interrupted transfer has the state "Deleted".

Diffusion List

If a transfer uses a Diffusion List, the single arrow is replaced by three arrows:

	• mode: Initiator • direction: send • state: ENDED • uses a Diffusion List

Routing states

The following icon indicates that the transfer must be routed or has already been routed:

	• mode: Responder • direction: receive • state: ENDED • To route or routed

If the direction of the small arrow is reversed, it indicates that the transfer was routed from another source:

	• mode: Initiator • direction: send • state: ENDED • Routed from

Viewing a transfer

When you display a transfer in read-only mode, three icons can appear on the **General** tab of the Transfer Request window. The first icon is always displayed and is the same as the icon that appears in the transfer list. In addition, the window may display the following icons:

	Real File Directory type transfer
	Virtual File Directory type transfer
	Security implemented
	Security not implemented

Axway Gateway: Managing Transfers

Transfer states in Gateway

Introduction

Gateway offers the possibility to enable additional processing based on events that occur during its functioning. One such event, which is an important feature of Gateway, is routing. Routing is triggered by a change in state of a transfer.

This section examines the meaning of each transfer state, the difference between apparently similar states and the state changes which trigger a routing event.

Transfer states

During its life cycle a transfer can have different processing phases, called states. There are three major phases that are common to all transfers: the beginning phase, the processing phase and the end of transfer phase. Depending on configuration options and on the protocol used, different states can correspond to each of these phases.

Beginning phase	Processing phase	End of transfer phase
• To begin • To restart • Frozen • Delayed • Created • To pack • Packing • To sign	• Processing • Processing (responder) • Suspended • Interrupted	• Ended • To ack • Acknowledged • Nack by user • Nack by XFB • Ended routing • Cancelled routing • Cancelled • To unpack • Unpacking • Ended third party failed (*) • Ended third party authorized (*) • Acked third party failed (*) • Acked third party authorized (*)

(*) These states are specific to SWIFTNet transfers.

Calculated states

Some of these states are calculated by taking into account the state, route state and trade state. These states offer a global overview of what is actually happening with the transfer (trade decoding/encoding, routing).

Examples:

Cancelled routing means that the transfer will not be routed.

Unpacking means that the transfer is ended and a trade processing is in progress.

This table lists the calculated states:

Calculated state	State	Route state	Trade state
To pack	To begin		To process
Packing	To begin		Processing

Calculated state	State	Route state	Trade state
To Unpack	Ended Ended to ack Acknowledged Nack by user Nack by XFB		To process
Unpacking	Ended Ended to ack Acknowledged Nack by user Nack by XFB		Processing
Ended wait receipt	Ended Ended to ack Acknowledged Nack by user Nack by XFB		Trade wait receipt
Ended third party failed	Ended Ended to ack		Trade third party failed Trade third party rejected
Ended third party authorized	Acknowledged Nack by user Nack by XFB		Trade third party authorized
Acked third party failed	Ended Ended to ack		Trade third party failed Trade third party rejected
Acked third party authorized	Acknowledged Nack by user Nack by XFB		Trade third party authorized
Ended routing	Ended Nack by user Nack by XFB	May route	
Cancelled routing	Cancelled	May route	

The calculated state field is used in internal processing for handling state transitions. For all other cases not listed in this table the State field is used as calculated state.

Note that *Ended routing* means that the transfer is scheduled for routing and *Cancelled routing* means that the transfer will not be routed.

State transitions

During the life cycle of a transfer, each change of state represents a transition – an event which triggers notif (notification) processes within Gateway. These processes are responsible for:

- Routing
- Connection to other software products (such as Sentinel)
- Calendar event scheduling
- and so on

State transitions for transfers in responder mode

Created >>> Processing (responder) >>> Ended routing >>> Ended

State transitions for transfers in initiator mode

Frozen >>> To begin >>> Processing >>> Ended routing >>> Ended

The states are listed in the [Route transfer states configuration menu](#). When a transfer reaches a configured state, router processing is triggered, meaning that the Decision Rule selection is started. Each transition to one of these configured states leads to Rule Table evaluation and Decision Rule triggering.

Example: By default the *Ended routing* state is selected. Each time a transfer ends, whether it was routed or not, Rule Table evaluation takes place. In this case, if you configure the *Ended* state to be notified to the routing process, you will see two Rule Table events triggered - one for each state. This happens due to the transition from *Ended routing* to *Ended*, because this whole notification mechanism is based on calculated transfer state changes.

By default, to avoid double notifications and to cover the majority of use cases, only a few states are selected. This setting can be considered the safest and avoids any unexpected behavior. We therefore recommend that you leave this setting unchanged.

Routing with SWIFTNet

When using SWIFTNet, Gateway does not know immediately if a routed transfer has ended successfully or not. Protocol-specific acknowledgments may be received some time after a transfer ends. Initially Gateway assumes a positive ack and the transfer is considered ended (*Ended to ack*).

It could happen that a negative answer is received from the partner and the transfer has to be cancelled.

In the Route transfer states configuration menu, the *Cancelled routing* state is configured by default. However, the *Nack by XFB* state is not configured by default. This might not be enough to trigger Rule Table evaluation.

Even if the state transition occurs (*Ended to ack >>> Nack by XFB*), as the final state is not configured, Rule Table evaluation is not triggered.

Ensuring that routing is triggered with SWIFTNet

We recommend that you enable the *Nack by XFB* state in the [Route transfer states configuration menu](#).

Route State Transitions

The following table shows which results trigger which route state transitions:

Route state	Transitions to	When
XFER_TO_ROUTE	XFER_ROUTED	Successfully route transfer to XIB/another transfer
	XFER_EN_ROUTE	Transfer switched to ended state
XFER_ROUTED	Not applicable	
XFER_ROUTE_FAILED		
XFER_NOT_ROUTED / XFER_NOT_TO_ROUTE	XFER_ROUTED	Routing is done on ACKED state, incoming transfer
XFER_EN_ROUTE	XFER_NOT_ROUTED	Transfer passed into Ended-to-ACK and direction was incoming
	XFER_ROUTE_FAILED	Not able to perform routing, calling of a script does not count

Route state	Transitions to	When
XFER_MAY_ROUTE / XFER_EN_ROUTE	XFER_ROUTE_FAILED	Not able to perform routing, calling of a script does not count
	XFER_NOT_ROUTED	Transfer is ended and direction is incoming and no routing was done
	XFER_NOT_TO_ROUTE	Transfer is ended and direction is outgoing
XFER_ACKED	XFER_ACKED	It is a final state so there are no transitions from it. The <code>route_state</code> becomes Acknowledged ("A") if the routed transfer is acknowledged

[Overview: File Transfers and Transfer Requests](#)

[Overview: Routing](#)

[SWIFTNet state transitions](#)

[Managing Events on Gateway](#)

Axway Gateway: Managing Transfers

Working with Transfers (command line)

[peltrans](#)

[peladm update trans](#)

[pelpur](#)

[pelpur clean swha db](#)

[peldsp select trans](#)

[peldsp status trans](#)

[peldsp display trans](#)

[pelctl control trans](#)

Submitting a Transfer: peltrans

Syntax	<code>peltrans <keyword> value <keyword> value</code>

Description	Use this command to submit: • Transfer Requests • Application Message Requests • acknowledgment Requests • List Requests: if the <code>dest_alias</code> parameter is a Diffusion List The <code>peltrans</code> command returns a Transfer Request identifier named <code>local_ident</code>. Before you submit a Transfer Request, you must create the Site objects that the Transfer object references. Optionally, you can also create Application objects, Diffusion List objects and Model objects to use in the Transfer Request. Although these objects are not mandatory, they simplify Transfer Request definition. Command arguments are keywords, each of which is followed by corresponding values (except for <code>f_text</code> which does not have a value: the argument is either present or not). Keywords are prefixed by the "-" (hyphen) character. Most keywords have abbreviated forms.
Parameters	<i>Optional parameters</i> Refer to: • General parameters for creating and updating Transfer Requests • SWIFTNet Transfer Request parameters • Diffusion List Transfer Request parameters

Usage rules	Interactive mode: If you set the <code>interactive (-i)</code> parameter, Gateway waits for the end of the transfer (terminating normally, on error or cancellation) before it continues the script. However, this mode may lock the entire application program if the transfer cannot run. Application: The Application and the destination Site aliases must belong to the same group. If the Application and Transfer parameters are not the same, the Transfer parameters take precedence. If you do not specify the Application name: • The file transfer takes the parameters that you explicitly specify in the Request • You must specify the parameters <code>rec_fmt</code> and <code>rec_len</code> (record format and record length) Gateway respects the following priorities when completing Transfer parameters: 1. User parameters 2. Protocol Model parameters 3. General Model parameters Date: Use the optional parameter <code>date_to_begin</code> to defer the file transfer. Use the optional parameter <code>date_to_end</code> to define a validity period for the Transfer. If the Transfer has not taken place or is in progress when this period expires, it is considered timed out and is canceled. Specify dates in the following format: • Date in full: <code>YYYYMMDD HHMMSS</code> • Time of day: <code>hhmmss</code> • Relative to the current time: <code>+hhmmss</code>
Example 1: Transfer Request with an Application	This example submits a Transfer Request for sending a file to SITEA using Application object FB80. The Application object parameters must be compatible with the file system attributes of the file to send. <pre>peltrans -dest_alias SITEA -comments "fixed record length: 80" -appli FB80 -file_component <name_of_file_to_send></pre>

Example 2: Transfer Request without an Application	Although it is recommended that you use an Application or a Model object to submit a Transfer Request, you can submit the same Transfer Request above without using an Application. In this case, you specify the record format and record length parameters. <pre> peltrans -dest_alias SITEA -comments "fixed record length: 80" -f_org S -rec_fmt F -rec_len 80 -data_code A -x_rec_fmt F -x_rec_len 80 -x_data_code E -file_component <name_of_file_to_send> </pre>
Example 3: Transfer Request with a Diffusion List and a Model	Diffusion Lists and Models enable you to submit Transfer Requests more easily. Create these objects before submitting the Transfer Request. If required, you can combine both a Diffusion List and a Model in a Request. This example submits a Transfer Request that uses the Model object FBPELPC. This Model object supplies the parameters <code>dest_alias</code>, <code>appli</code>, <code>compression</code>, and so on. <pre> peltrans -org_alias LOC_SITE -model FBPELPC -comments "fixed record length: 255" -file_component <name_of_file_to_send> </pre> This example creates a Diffusion List Transfer Request using the Diffusion List SPAIN_L. <pre> peltrans -org_alias LOC_SITE -dest_alias SPAIN_L -appli FB80 -file_component <name_of_file_to_send> </pre>

Example 4: Transfer Request in frozen state	When you create a Transfer Request in frozen state, Gateway does not submit the Request immediately. To submit the Request, you must use the command <code>pelctl control_trans</code> and set the <code>action</code> parameter to <code>R</code> (restart). You can use frozen state in all kinds of Transfer Requests in initiator mode. Creating Transfer Request in frozen state: The following Transfer Request sends a file to SITEA with Application FB80. <pre>peltrans -dest_alias SITEA -comments "Create Transfer in frozen state" -frozen_state -appli FB80 -file_component <name_of_file_to_send></pre> Restarting a Transfer Request in frozen state This example switches the state of Transfer Request xxxx from FROZEN to TO_BEGIN state. <pre>pelctl control_trans -i xxxx -action R</pre>
Example 5: PEL protocol Transfer Request	<pre>peltrans -org_alias PARIS -dest_alias LILLE -fc /u/Gateway/profile -m I -dir O -ft 00002 -fn compta -comp B -date_to_begin +003000 -date_to_end "19990807 093000" -prio 1 -f_org S -rf F -rl 80 -dc A -alloc_unit R -snd_msg file_compta peltrans -server PARIS -type LOTS -org_alias PARIS -dest_alias LILLE -date_to_begin +003000 -appli FB80</pre>

Example 6: PeSIT HS protocol Transfer Request	This example is for a Local Site in Responder/Sender mode. The Local Site makes available the file named FILE to the Remote Site named REMOTE_SITE under the logical name APPLI: <pre> peltrans -org_alias SITE_LOCAL -dest_alias REMOTE_SITE -appli APPLI -file_name APPLI -file_type 0 -file_component FILE -mode R -direction 0 </pre>
Example 7: Local Site: Initiator/ Receiver	This example is for a PeSIT transfer, with a request for the reception of the logical file named APPLI from the Site named REMOTE_SITE. <pre> peltrans -org_alias REMOTE_SITE -dest_alias SITE_LOCAL -appli APPLI -file_name APPLI -file_type 0 -mode I -direction I </pre>
Example 8: Local Site: Initiator/ Sender	This example is for a PeSIT transfer, sending the logical file named APPLI to the destination Site named REMOTE_SITE. <pre> peltrans -org_alias SITE_LOCAL -dest_alias REMOTE_SITE -appli APPLI -file_name APPLI -file_type 0 </pre>
Example 9: SWIFTNet Transfer Request	This example is for a SWIFTNet message transfer. The local partner LSWTRADE is sending the message "Payload" through the destination Site SW_SITE to the partner SWTRADEIART. <pre> peltrans -type MSG -dest_alias SW_SITE -trade_org_alias LSWTRADE -trade_dest_alias SWTRADEIART -snd_msg "Payload" -request_type abcd.0000.0000 -appli FTP_B </pre>

Modifying a Transfer: *peladm update_trans*

Syntax	<code>peladm update_trans {-ident} [optional parameters]</code>

Description	Use this command to modify the properties of a Transfer Request.
Parameters	<code>-ident (-i)</code> : Enter the transfer identifier of the Transfer Request you want to modify.
	<i>Optional parameters</i> Refer to: • General parameters for creating and updating Transfer Requests • SWIFTNet Transfer Request parameters • Diffusion List Transfer Request parameters
	<code>-user_processed (-usr_proc)</code> : Enter Y (yes) or N (no) to activate a complementary status indicator showing whether or not the file has been correctly processed.
	<code>-user_state (-user_st)</code> : Enter a single character to create a customized indication of the transfer status. You can use the value of this parameter as a selection criteria.
	<code>-param1 (-p1)</code> : Enter up to 40 alphanumeric characters. Gateway uses the values you enter in this parameter locally. It does not transfer the values to your transfer partner. Use this parameter to locally store values that you can reuse via Decision Rules, Models and batch procedures.
	<code>-param2 (-p2)</code> : Enter up to 40 alphanumeric characters. Gateway uses the values you enter in this parameter locally. It does not transfer the values to your transfer partner. Use this parameter to locally store values that you can reuse via Decision Rules, Models and batch procedures.
	<code>-date_to_begin (-dtb)</code> : Enter a date in the format <code>YYYYMMDD HHMMSS</code>
	<code>-date_to_end (-dte)</code> : Enter a date in the format <code>YYYYMMDD HHMMSS</code>
Usage rules	All parameters except <code>ident</code> are optional. If not specified, a parameter keeps its previous value.
Example	<pre>peladm update_trans -i 2567 -purge_option Y</pre>

Purging a Transfer from the Mailbox: pelpur or pelpur xferPurge

Syntax	<pre>pelpur {-ident} [optional parameters] or pelpur xferPurge {-ident} [optional parameters]</pre> Note: Both formats can be used. They have the same result.

Description	Use this command to clear Transfer Request records that match the specified criteria from the Mailbox. To delete all matching Transfer Request records without creating an archive file, set the parameter <code>no_archive</code> . To archive records before purging the Mailbox, do not set this parameter.
Parameters	<code>-ident (-i)</code>: Enter the transfer identifier of the Transfer Request record you want to purge. <i>Optional parameters:</i> Refer to pelpur parameters.
Usage rules	Purge with archive option: If you do not set the archive option <code>no_archive</code> for pelpur, Gateway creates an archive file that contains information on selected and purged Transfers. The path name of the file is platform-dependent and is built from the current date and time. The archive file contains one record for each purged Transfer. Each record describes a subset of Transfer Request parameters, separated by blank characters, and ending with a new line character. Within the record, numeric parameters are translated into character strings.
Associated tasks	You can also configure Gateway to archive Transfers automatically. Refer to Configuration: Gateway parameters .
Example 1: Purge all ended Transfers	This example purges <i>all</i> ended Transfers (state Ended, Canceled and Aced) from the Mailbox: <pre>pelpur -all</pre>
Example 2: Purge according to status	This example purges Transfers with the status that you specify, in this case Canceled and Delayed: <pre>pelpur -list_trans_state CD</pre>
Example 3: Purge resolved script file	This example purges the resolved script file (EXPR) and its output files for the Transfer: <pre>pelpur -purge_xpr</pre>
Example 4: Purge transferred file	This example purges the file sent or received, on condition that the <code>purge_option</code> parameter in the Transfer Request is set to Y (yes): <pre>pelpur -purge_file</pre>
Example 5: Purge selected transfers	This example purges all Transfers with identifiers between 10 and 20 and in the Delayed, Canceled or Suspended states: <pre>pelpur -ident_inf 10 -ident_sup 20 -list_trans_state DCS</pre>

Note The payload file for a Transfer which was routed to Integrator is not deleted by a purge operation.

For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).

Purging a Transfer from the HA database (used by SWIFTNet High Availability): pelpur clean_swha_db

Syntax	<code>pelpur clean_swha_db [parameters]</code>
Description	Use this command to remove transfer entries that match the specified criteria from the HA database (used by SWIFTNet High Availability). To purge a transfer, the SWIFTNet High Availability feature has to be activated and connectivity with HA database must be functional.
Parameters	Refer to pelpur parameters .
Usage rules	At least one parameter (criteria) has to be specified in the subcommand.
Example 1: Purge a transfer with a specific gateway id	Purging a transfer with gateway id "7". <pre>pelpur clean_swha_db -gatewayID 7</pre>
Example 2: Purge transfers matching specific gateway id and transfer id range	Purging a transfer with gateway id "7" and transfer id between "5" and "11". <pre>pelpur clean_swha_db -gatewayID 7 -fromTransferID 5 -toTransferID 11</pre>
Example 3: Purge transfer with specific local partner alias and specific date	Purging a transfer with local partner alias "bnpparibas" and create date "11 June 2014". <pre>pelpur clean_swha_db -localPartner bnpparibas -createDate 20140611</pre>
Example 4: Purge transfers with specific request type and created between 2 dates	Purging a transfer with requestType "xsys.xxx.fa.nro.hdo" and that was created after "12 June 2014" and before "20 June 2014" <pre>pelpur clean_swha_db -requestType xsys.xxx.fa.nro.hdo -fromCreateDate 20140612 -toCreateDate 20140620</pre>

Selecting a Transfer: peldsp select_trans

Syntax	<code>peldsp select_trans [optional parameters]</code>

Description	Use this command to select the Transfer or List Requests that match the criteria entered as parameters. You can refine your search using the optional parameters.
Parameters	<i>Optional parameters:</i> Refer to peldsp parameters .
Example	This example displays all Transfer records for acknowledgment type Transfers. peldsp select_trans -type EERP 1 2 3

Displaying Transfer Request states: peldsp status_trans

Syntax	peldsp status_trans [optional parameters]																												
Description	This command displays the list of transfers and related attributes. It can be used for debugging purposes. Columns returned (that can be correlated with the <code>display_trans</code> sub-command): <table> <tr> <td>Number</td><td>x_local_ident</td></tr> <tr> <td></td><td>depends on the direction; if Incoming, its x_originator (or x_org_alias if x_originator is empty), while for Outgoing its x_destination (or x_dest_alias if x_destination is empty).</td></tr> <tr> <td>Site</td><td></td></tr> <tr> <td>Type</td><td>x_type</td></tr> <tr> <td>D</td><td>x_direction</td></tr> <tr> <td>Day</td><td>x_yday</td></tr> <tr> <td>Proto_Id</td><td>"transfer protocol key" computed depending on the protocol, based on x_xfer_id (it can also include x_file_name) except for PEL and ETB3 when it's x_sequence.</td></tr> <tr> <td>Appli</td><td>x_appli</td></tr> <tr> <td>State</td><td>x_state</td></tr> <tr> <td>File</td><td>x_file_component</td></tr> <tr> <td>R</td><td>x_retry_count</td></tr> <tr> <td>KBytes</td><td>x_f_bytes (in KB, rounded)</td></tr> <tr> <td>Sc</td><td>x_purge_option (Y/N)</td></tr> <tr> <td>U</td><td>x_user_processed (Y/N)</td></tr> </table>	Number	x_local_ident		depends on the direction; if Incoming, its x_originator (or x_org_alias if x_originator is empty), while for Outgoing its x_destination (or x_dest_alias if x_destination is empty).	Site		Type	x_type	D	x_direction	Day	x_yday	Proto_Id	"transfer protocol key" computed depending on the protocol, based on x_xfer_id (it can also include x_file_name) except for PEL and ETB3 when it's x_sequence.	Appli	x_appli	State	x_state	File	x_file_component	R	x_retry_count	KBytes	x_f_bytes (in KB, rounded)	Sc	x_purge_option (Y/N)	U	x_user_processed (Y/N)
Number	x_local_ident																												
	depends on the direction; if Incoming, its x_originator (or x_org_alias if x_originator is empty), while for Outgoing its x_destination (or x_dest_alias if x_destination is empty).																												
Site																													
Type	x_type																												
D	x_direction																												
Day	x_yday																												
Proto_Id	"transfer protocol key" computed depending on the protocol, based on x_xfer_id (it can also include x_file_name) except for PEL and ETB3 when it's x_sequence.																												
Appli	x_appli																												
State	x_state																												
File	x_file_component																												
R	x_retry_count																												
KBytes	x_f_bytes (in KB, rounded)																												
Sc	x_purge_option (Y/N)																												
U	x_user_processed (Y/N)																												
Parameters	<i>Optional parameters:</i> Refer to peldsp parameters .																												

Example	When you enter the following command, Gateway returns the file transfer numbers that match the selection criteria. <pre>peldsp status_trans</pre> Example of a return screen display.

Displaying Transfer details: *peldsp display_trans*

Syntax	<code>peldsp display_trans {-local_ident}</code>
Description	Use this command to display the details of a Transfer that you specify by its identifier. The command returns the Transfer parameters.
Parameters	<code>-local_ident (-i)</code> : Enter the transfer identifier of the Transfer Request you want to display.
Usage	Each parameter is displayed on a separate line in the form: <pre>name_of_field="value"</pre> Each field name is prefixed by the string "x_", relating to the xfer (Transfer) object.
Example 1: Displaying a Transfer Request	When you enter the command: <pre>peldsp display_trans -i 114</pre> Gateway returns the following type of information: <pre>x_local_ident=114 x_type=TRANS x_org_alias=GFL660 x_dest_alias=LPC x_appli=DEFAULT_B x_originator=GFL660 x_destination=TPHSE x_direction=0 x_mode=I (...)</pre>
Example 2: Displaying a Diffusion List Request	When you enter the command: <pre>peldsp display_trans -i 8</pre> Gateway returns the following type of information: <pre>x_local_ident=8 x_type=LIST x_org_alias=R65 x_dest_alias=SLIST1 (...)</pre>

Controlling a Transfer: *pelctl control_trans*

Syntax	<code>pelctl control_trans {-local_ident} {-action} [-retry_delay]</code>
Description	Use this command to suspend, restart or cancel a Transfer Request or a Diffusion List Request. You can only suspend a Request if its state is one of the following: • To Begin • Delayed • Processing To restart a Request, use <code>pelctl control_trans</code> with the Restart operation. A Restart operation only takes effect if the latest end-of-transfer date is already reached. If you perform a restart operation on a delayed Transfer Request: • <i>before</i> the start date, the Transfer state switches to Frozen • <i>after</i> the start date, the Transfer state switches to Ended (with error code)
Parameters	<code>-local_ident (-i)</code> : Enter the transfer identifier of the Transfer Request you want to display.
	<code>-action (-a)</code> : Enter one of the following values: • C = Cancel • S = Suspend • R = Restart
	<code>-retry_delay (-rd)</code> : Enter the number of seconds of delay before retry.
Example	This example suspends the Transfer with the identifier 36: <pre>pelctl control_trans -i 36 -a S</pre>

[Transfers: Parameters List](#)

Transfers: Parameters List

[General parameters for creating and updating transfers](#)

[Additional Transfer Request parameters for OFTP](#)

[Additional Transfer Request parameters for SWIFTNet](#)

[Additional Transfer Request parameters for X.400](#)

[Additional Transfer Request parameters for Diffusion Lists](#)

[Purging Transfers: `pelpur` command parameters](#)

[Displaying Transfers: `pe1dsp` command parameters](#)

General parameters for creating and updating transfers

The following table contains parameters you can use with the `pe1trans` and/or the `pe1adm update_trans` commands. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Meaning	Value
<code>-acknowledgment_option (-ao)</code> Ack option	<i>OFTP, PeSIT, SMTP and SWIFTNet only</i> Specify acknowledgement behavior for this Transfer. For SMTP, refer to Configuring SMTP: Requesting an acknowledgement .	One of: • TO_ACK = Wait for acknowledgement • NOT_TO_ACK = Do not wait for acknowledgement • ACK_UNKNOWN = Unspecified acknowledgement behavior
<code>-append (-app)</code> Append option	<i>FTP client mode only</i> Option that allows you to add data to the end of an existing file	One of: • Y = append allowed • N = append forbidden (default)
<code>-appli (-ap)</code> Application	Application name	Maximum: 25 alphanumeric characters
<code>-attach_extract_file (-aef)</code> Attach file	<i>SMTP only</i> Define whether to create and attach a file or not.	One of: • Y • N • Undefined (default)
<code>-block_size (-bs)</code> <i>(None)</i>	For BULL machines and AS400: Block size in bytes	Integer
<code>-compression (-comp)</code> Compression	Compression type	One of: • U = Undefined (default) • H = Horizontal • V = Vertical • B = Both • Z = Zlib (<i>PeSIT only</i>)

Parameter	Meaning	Value
-content_type (-dct)	Optional transfer content information	Free text description of the transfer content type. For example, you may choose to enter one of the following content types: • application/EDI-X12 • application/EDIFACT • application/XML
-data_code (-dc) Data code	Data code	One of: • B = Binary • A = ASCII • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E = EBCDIC • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • ISO = ISO • OEMPC = OEMPC • U = Undefined
-date_to_begin (-dtb) Earliest start date	Date to begin Transfer (requested)	In the format <i>YYYYMMDD HHMMSS</i>
-date_to_end (-dte) Latest start date	Date to end Transfer (requested)	In the format <i>YYYYMMDD HHMMSS</i>

Parameter	Meaning	Value
-destination_address (-addr)	Remote Site destination address to use together with the <code>username</code> and <code>password</code> parameters to create a temporary destination site for a transfer. Use these parameters in place of the association of a destination Site object.	The remote destination address for a transfer.
-dest_alias (-da) <i>Destination alias</i>	Destination Site alias	Maximum: 31 alphanumeric characters
-destination (-dest) <i>Destination</i>	Destination protocol identifier	Maximum: 31 alphanumeric characters
-dir_name (-dn) <i>Directory name</i>	<i>HTTP, FTP & SFTP only</i> Absolute path of the directory that contains the file to transfer. The directory depends on the mode: • Initiator mode: a directory on the remote system • Responder mode: a directory in the VFD If the Transfer mode is DIR, specify the name of the directory whose contents you want to list.	Maximum: 127 alphanumeric characters FTP DIR Transfer type: make sure you add the separator character <code>/</code> at the end of the directory name.

Parameter	Meaning	Value
-dir_name_alias (-dna) Directory alias	<i>HTTP, FTP & SFTP only</i> Only available when submitting a Transfer Request in Responder mode: Alias of the VFD directory that contains the file to transfer	Maximum: 32 alphanumeric characters
-dir_type (-dt) Directory listing type	<i>FTP only</i> This parameter is only significant for DIR type Transfers in Initiator/ Receiver mode. This parameter defines the level of detail in the directory listing.	One of: • NAMES: (default) sends an NLST command that returns a list of file and directory names. • LONG: sends a LIST command that returns more detailed information on each element.
-direction (-dir) Direction	Transfer direction	One of: • I = In • O = Out
-external_sentinel_ident (-esi) <i>(None)</i>	Sentinel identifier that the external application passes in batch mode only. This identifier is not stored in the Transfer record.	Maximum: 250 alphanumeric characters
-external_sentinel_objectname (-eso) <i>(None)</i>	Sentinel object name to be passed by external applications. Batch mode only. Not stored in the Transfer record.	Maximum: 32 alphanumeric characters
-file_component (-fc) File component	Local file name	Maximum: 127 alphanumeric characters

Parameter	Meaning	Value
-file_label (-fl) File label	<i>PeSIT HS only</i> Protocol label of the file	PeSIT PI 37 Maximum: 256 alphanumeric characters
-file_name (-fn) File Name	Protocol file name For FTP and HTTP DIR type Transfers, this parameter defines a filter on the file names that the server returns. For example: *.txt returns all .txt files in the directory that you specify.	Maximum: 76 alphanumeric characters
-file_org (-f_org) Organization	File organization	One of: • S = Sequential file (default) • I = Indexed file • R = Relative file
-file_type (-ft) File type	<i>PeSIT HS only</i> File type	Maximum: 16 alphanumeric characters
-frozen_state (-fs) Frozen state	Create the Transfer in frozen state	Parameter without a value

Parameter	Meaning	Value
-ftp_command (-ftpc) HTTP CGI parameters/FTP user command	<i>FTP, SFTP, HTTP and SWIFTNet InterAct only</i> - FTP: Send a special remote command or an unsupported command to the FTP server before or after the Transfer. <code>SITE</code> and <code>QUOTE</code> commands are supported. - SFTP: Send commands to remote server using <code>QUOTE</code> or <code>SITE</code> command - HTTP: Add CGI parameters to the Transfer Request - SWIFTNet InterAct: Specify payload attributes or compression information	Maximum: 80 alphanumeric characters FTP and SFTP: Enter the command in the following format: <pre>"SITE <fct1> <param1>; SITE <fct2> <param2>"</pre> You can overcome the 80 character limitation by calling a script that contains a set of FTP commands (<code>SITE</code>, <code>QUOTE</code>). HTTP: Enter CGI parameters in the following format: <pre>xfer_ param1=value1&xfer_ param2=value2</pre> SWIFTNet InterAct: Enter either: - payload attributes in the following format: <code>attrib1=\"value1\"</code> - compression information in the following format: <code>dt=\"GZIP-Base64\"</code>

Parameter	Meaning	Value
-generate_xfb_subject (-gxs) Add Gateway parameters to subject	<i>SMTP/POP3 only</i> Set this option to Yes to add the following Gateway-specific parameters to the email subject: • destination (-dest) • origin (-org) • protocol file name (-file_name) • application (-appli)	One of: • N • Y • U (default)
-hist_create_date (-hd) <i>(None)</i>	PeSIT creation date	In the format <i>YYYYMMDD HHMMSS</i>
-interactive (-i) <i>(None)</i>	Wait for the end of the Transfer Request before starting a new one	No value
-interval (-int) Interval	Retry interval • PEL: interval between two Remote Site polls (LOTS and POLL commands) • PeSIT HS: cyclic selection request only: repetition time interval in seconds, in case of selection failure	Integer
-keep_offset (-ko)	Indicative value for indexing on the remote machine	PeSIT PI 39

Parameter	Meaning	Value
-key_len (-kl)	Indicative value for indexing on the remote machine	PeSIT PI 38
-local_agent (-la) Local agent alias	Alias of the associated Local Site	Maximum: 31 alphanumeric characters
-max_file (-m_file) Maximum number of files	<i>PeSIT HS cyclic selection request only</i> Maximum number of files allowed	Decimal value: 0 - 32000 . 1 = no imposed limits.
-max_request (-m_req) Maximum number of requests	<i>PeSIT HS cyclic selection request only</i> Maximum number of requests allowed	Decimal value: 0 - 32000 . 1 = no imposed limits.
-mode (-m) Mode	Connection mode for the Transfer	One of: • I = Initiator • R = Responder
-model (-ml) Model name	Model used to submit the Transfer Request	Maximum: 25 alphanumeric characters

Parameter	Meaning	Value
-multi_filter (-muf) Filter	<i>FTP and SFTP multiple transfers only</i> Specify the filtering criteria to select files to send or receive. You can filter according to: • Time • File size • File name	Time time = [last rounded] N [day[s] hour[s] minute[s]] [yyyyymmdd <=] time [<= yyyyymmddss] [yyyyymmddhhmmss <=] time [<= yyyyymmddhhmmss] For example, if it is currently 2010/06/13 17:15:28, then: • last1day = 2010/06/12 17:15:28 • rounded1day = 2010/06/13 00:00:00 • last3days = 2010/06/10 17:15:28 • rounded3days = 2010/06/11 00:00:00 • last1hours = 2010/06/12 16:15:28 • rounded1hours = 2010/06/12 17:00:00 File size [min <] size [< max] File name name = regexp (use StringMatch) Note: all the spaces that comes after "name =" are treated as part of the file name. To search for a file that starts with file , you should use : name =file* Example: -muf "20100528 <= time <= 20100609 ; 1 < size < 100000 ; name =*.dat"

Parameter	Meaning	Value
-multi_hash_type (-muh) Hash type	<i>FTP and SFTP multiple transfers only</i> Specify the type of hashing to use. This parameter determines whether all files in a directory will be transferred a second time or only those files that have changed since the initial transfer. This is useful if a transfer has been interrupted or only updated files need to be sent.	One of: 0 : No hash. No check of duplicate file transfers is made. All files are transferred again. 1 : (default) Gateway checks the following criteria to determine if a file has changed since the initial transfer: • alias of the partner • direction • file size • file date • directory name of the master • file name (short for MGET, long for MPUT) If none of these criteria has changed, the file is considered as unchanged and is therefore not sent. 2 : <i>For future use</i> 3 : <i>For future use</i>

Parameter	Meaning	Value
-multi_model (- mum) <Model name> Model to use for child transfers	<i>FTP and SFTP multiple transfers only</i> Specify the name of the Model to use to submit <i>child</i> transfers. If this parameter is not specified, the Transfer Request for a <i>child</i> is created using the Application of the <i>master</i> . In all cases, a Transfer Request for a <i>child</i> is made with type = TRANS and mode, direction, org_alias and dest_alias derived from the <i>master</i> .	Maximum: 25 alphanumeric characters
-nack_by_user (- nack) Negative ack option	<i>PeSIT HS only</i> The Nack (Negative acknowledgement) option indicates whether the Remote Site supports the negative EERP extension or not. If this option is not set, all negative acknowledgement requests are refused.	• Y • N
-newline_ convention Newline convention	<i>SFTP only</i> Set the end-of-line code, depending on the remote system type. This parameter is valid for text file transfers in both send and receive mode.	One of: • U or u (UNIX) • W or w (Windows) • ' (undefined)

Parameter	Meaning	Value
<code>-org_alias (-oa)</code> Originator alias	Originator Site alias	Maximum: 31 alphanumeric characters
<code>-originator (-org)</code> Originator	Originator protocol identifier	Maximum: 31 alphanumeric characters
<code>-padding (-pad)</code> Padding character	Padding character code	Hexadecimal value: 00 - FF
<code>-param1 (-p1)</code> Param 1	User parameter 1. Gateway uses the values you enter in this parameter locally. It does not transfer the values to your transfer partner. Use this parameter to locally store values that you can reuse via Decision Rules, Models and batch procedures.	Maximum: 40 alphanumeric characters
<code>-param2 (-p2)</code> Param 2	User parameter 2. Gateway uses the values you enter in this parameter locally. It does not transfer the values to your transfer partner. Use this parameter to locally store values that you can reuse via Decision Rules, Models and batch procedures.	Maximum: 40 alphanumeric characters

Parameter	Meaning	Value
-password (- passwd)	Password to use together with the <code>username</code> and <code>dest_address</code> parameters to create a temporary destination site for a transfer. Use these parameters in place of the association of a destination Site object.	The password associated with the value entered in the <code>username</code> parameter.
-permanent (- perm) Permanent	This parameter controls the availability of files for download by remote clients when Gateway is acting in <i>server mode</i> . Usually (when <code>permanent = N</code>), Gateway makes a specified file available only to the first client that submits a Transfer Request for the file. If you activate this option (<code>permanent = Y</code>), Gateway responds to each client's Transfer Requests by sending a copy of the requested file, keeping the original in the database available for subsequent client Transfer Requests.	Select one of: • Y (to set Transfer availability to permanent) • N (default) The value that you set here overrides the setting in the Model definition.

Parameter	Meaning	Value
-priority (-prio) Priority	<i>Also for X.400 - described later in this topic</i> Transfer priority	One of: • 0 = High priority • 1 = Medium priority • 2 = Low priority
-purge_option (-po) Purge option	Enable purge of files associated with the Transfer Request, using the pelpur command.	• Y • N
-rcv_appli (-ra) Application	<i>PeSIT HSE Store-and-Forward only</i> Receiver Application name	Maximum: 8 alphanumeric characters
-rcv_text (-rt) Text	<i>PeSIT HSE Store-and-Forward only</i> Receiver free text	Maximum: 8 alphanumeric characters
-rcv_user (-ru) User	<i>PeSIT HSE Store-and-Forward only</i> Receiver user name	Maximum: 8 alphanumeric characters
-rec_fmt (-rf) Record format	<i>Record format</i>	One of: • F = Fixed • V = Variable • S = Stream • FT = Fixed Text • VT = Variable Text
-rec_len (-rl) Record length	<i>Record length</i>	Integer: 0 - 32768
-receipt_nack_msg (-rnm)	<i>EDIINT. Gateway operating in Responder mode on receipt of a transfer</i> Optionally, enter a text explaining a warning or error receipt.	Explanatory text. Maximum: 512 characters.

Parameter	Meaning	Value
-receipt_reply_to_xfer (-rrtx) New Acknowledgement Message: Transfer ID to ack	<i>EDIINT or SWIFTNet. Gateway operating in Responder mode on receipt of a transfer.</i> Use this parameter to activate the generation of a receipt on reception of a transfer.	The number of the transfer for which you want to generate and send a return receipt.
-receipt_request (-rr) Receipt request	<i>EDIINT, SWIFTNet FileAct Realtime/SnF and SWIFTNet InterAct SnF only</i> Specify whether a return receipt is required when sending a file to a partner.	One of: • undefined (default) • none = no receipt requested • signed = request signed receipt • unsigned = request unsigned receipt (<i>EDIINT only</i>)
-receipt_request_sync (-rrs)	<i>EDIINT only</i> Synchronize sending of receipts.	• Y • N
-receipt_request_to (-rrto) <i>Depending on protocol:</i> Receipt url/email Receipt DN	<i>EDIINT:</i> The URL or email address to which the acknowledgement is sent. <i>SWIFTNet FileAct Realtime:</i> The DN to which the acknowledgement is sent.	The URL, email address or DN to which you want the transfer receipt to be sent.
receipt_type (-rt)	<i>EDIINT. Gateway operating in Responder mode on receipt of a transfer</i> Specify the type of receipt to return to the initiator of the received transfer.	The receipt type. One of: • SUCCESS • ERROR

Parameter	Meaning	Value
-remote_agent (-ra) Remote agent alias	Alias of the associated Remote Site	Maximum: 31 alphanumeric characters
-reply_to_xfer (-rtx) Reply to	Local identifier of the Transfer Request to which the response must be sent	Integer
-request_method (-rm) Request method	<i>HTTP only</i> Method to use during Transfer in HTTP protocol	Incoming Transfers: • GET • POST Outgoing Transfers: • POST • PUT
-response_request_sync (-resprs) Request a response	<i>RosettaNet only</i> Send a message requesting a synchronous response	• Y • N (default)
-response_to (-resptx) Send a response	<i>RosettaNet only</i> Send a response	Enter the local transfer identifier of the answered RosettaNet message
-retry_count_max (-rc_max) Maximum number of retries	Maximum number of retries allowed	Integer: 0 - 999. Default = 10.
-route_from_xfer (-rfx) Route from	Local identifier of the Request from which the Transfer is routed	Integer
-route_state (-rs) Route state	<i>Updating Transfer:</i> Routing state	One of: • TO_ROUTE • ROUTED • FAILED • ACKNOWLEDGED

Parameter	Meaning	Value
<code>-sentinel_transfer_filter (-stf)</code> Sentinel Transfer Filter	Indicate notification level for Transfer state changes to send to Sentinel server.	One of: • ALL • SUMMARY • NONE • UNDEFINED (default)
<code>-server</code> <i>(None)</i>	Site name for the PEL LOT command	Maximum: 25 alphanumeric characters
<code>-smtp_bcc (-bcc)</code> Bcc	<i>SMTP only</i> Blind copy recipients	No max length
<code>-smtp_cc (-cc)</code> Cc	<i>SMTP only</i> Copied recipients	No max length
<code>-smtp_from (-from)</code> From	<i>SMTP/POP3 only</i> Email <i>from</i> field. This parameter is not required if you set the <code>attach_extract_file</code> parameter to <code>No</code> . The file is already formatted as an email and already contains the From address.	Maximum: 128 characters

Parameter	Meaning	Value
-smtp_subject (-subject) email subject	<i>SMTP/POP3, AS2, AS3</i> <i>Also for X.400 - described later in this topic</i> Email subject. This parameter is mandatory if Gateway must create the email (<code>attach_extract_file</code> parameter set to Yes). If you are sending a formatted email via a Gateway Transfer Request, do not complete this parameter.	Maximum: 128 characters
-smtp_to (-to) To	<i>SMTP/POP3 only</i> Email <i>to</i> field. This parameter is not required if you set the <code>attach_extract_file</code> parameter to No. The file is already formatted as an email and already contains the To address.	Maximum: 128 characters
-snd_appli (-sa) Application	<i>PeSIT HSE Store-and-Forward only</i> Sender Application name	Maximum: 8 alphanumeric characters

Parameter	Meaning	Value
-snd_msg (-sm) <i>Depending on protocol:</i> Message to send Send text message	- OFTP: User message to transmit at the OFTP connection phase - FTP: List of special commands to send to the remote server using SITE or QUOTE FTP command - PeSIT: User message to transmit at the connection phase - HTTP: User message to add CGI parameters or non-standard HTTP headers to the Transfer Request - EDIINT: User message providing details about a WARNING or ERROR return receipt - SWIFTNet FileAct: [0 - 255]: Transfer description, supplied by the initiator. Free-text description of the transfer. [256 - 511]: Transfer	- OFTP: Up to 8 alphanumeric characters - FTP: Up to 512 alphanumeric characters - PeSIT: Up to 512 alphanumeric characters - HTTP: Up to 512 alphanumeric characters. For CGI parameters, use the format: xfer_param1=value1&xfer_param2=value2 For non-standard HTTP headers, use the format: [xs4hh]X-Id1=v1&X-Id2=v2[/xs4hh]. - EDIINT: Up to 512 alphanumeric characters - SWIFTNet FileAct: Max 256 alphanumeric characters for each part - SWIFTNet InterAct: Max 512 alphanumeric characters

Parameter	Meaning	Value
	information, supplied by the initiator. Structured information concerning the transfer. For more details, refer to the SWIFT documentation. • SWIFTNet InterAct: ◦ [0 - 511]: User message.	
-snd_msg_file (-smf) Send text message in file	<i>Submitting an Application or Acknowledgement message:</i> The message content to send. Use this option if the message is stored in an external file (PI 91).	No value
-snd_text (-st) Text	<i>PeSIT HSE Store-and-Forward only</i> Sender free text	Maximum: 8 alphanumeric characters
-snd_user (-su) User	<i>PeSIT HSE Store-and-Forward only</i> Sender user name	Maximum: 8 alphanumeric characters
-sys_dep (-sd) <i>(None)</i>	For BULL, AS400, GCOS and TANDEM machines: File type is system-dependent.	No value
-text_file Text file	Text file flag	• Y • N
-tpm_dea_id (-tpmdea)	PassPort (TPM) Document Exchange Address identifier	Integer
-tpm_dea_id_2 (-tpmdea_2)	PassPort (TPM) Document Exchange Address identifier	Integer

Parameter	Meaning	Value
-trade_compress (-tradecomp) <i>EDIINT:</i> Is compressing <i>SWIFTNet:</i> Compress	<i>EDIINT and SWIFTNet only</i> Compression algorithm for use when sending a file or message to a partner.	• Y = Use the compression algorithm • N = No compression
-trade_dest (-traded) <i>EDIINT:</i> (None) <i>SWIFTNet:</i> Destination DN	<i>EDIINT and SWIFTNet only</i> The name of the destination partner of the trading partner exchange.	Maximum: 128 characters
-trade_dest_alias (-tradedea) Destination partner	<i>EDIINT and SWIFTNet only</i> Alias of the destination partner of the trading partner exchange.	Maximum: 128 characters
-trade_encrypt (-tradee) <i>EDIINT:</i> Is encrypting <i>SWIFTNet:</i> Encrypt	<i>EDIINT and SWIFTNet InterAct only</i> Ciphering algorithm for use when sending a file to a partner.	• Y • N
-trade_file_already_encoded (-tredefae)	<i>EDIINT only</i> The use of this parameter is not recommended when submitting Transfer Requests. Use this parameter in Models to be applied on Decision Rule routing via EDIINT.	• Y • N

Parameter	Meaning	Value
-trade_format (-trade f) (None)	<i>EDIINT and SWIFTNet only</i> Protocol version.	One of: • AS1 • AS2 • AS3 • SMIME • SWIFTNET
-trade_org (-trade o) <i>EDIINT:</i> (None) <i>SWIFTNet:</i> Originator DN	<i>EDIINT and SWIFTNet only</i> The name of the originating partner of the trading partner exchange.	Maximum: 128 characters
-trade_org_alias (-trade oa) Originator partner	<i>EDIINT and SWIFTNet only</i> Alias of the originating partner of the trading partner exchange.	Maximum: 128 characters
-trade_sign (-trade s) <i>EDIINT:</i> Is signing <i>SWIFTNet:</i> Sign	<i>EDIINT and SWIFTNet only</i> Certificate signing algorithm used when sending a file to a partner.	• Y • N
-trade_state (-trade ts)	<i>EDIINT only</i> Select transfer	One of: • TO_PROCESS • PROCESSING • PROCESSED • ACKED

Parameter	Meaning	Value
-truncating_ allowed (-ta) Truncating allowed	If the Transfer uses the truncating feature, this option allows you to truncate a record without taking into account the padding character.	• Y • N
-type (-ty) Type	Transfer type	One of: • TRANS • LIST • POLL • LOTS • EERP • SELECT • MSG (equivalent to INTERACT Transfer type when using the GUI) • DIR • RECEIPT • MULTI: MGET or MPUT for FTP or SFTP multiple transfer (the distinction between MGET and MPUT is indicated by the <i>direction</i> parameter)
-unique_filename (-uf) Unique filename option	<i>FTP only</i> Store unique option	• Y • N
-user_processed (-usr_proc) User processed	<i>Modifying a Transfer:</i> Transfer processed by user	• Y • N

Parameter	Meaning	Value
-user_state (-usr_st) User state	<i>Modifying, a Transfer:</i> User state field managed by the user application. Gateway does not interpret this field.	One character
-username (-un)	User login name to use together with the password and dest_address parameters to create a temporary destination site for a transfer. Use these parameters in place of the association of a destination Site object.	Enter the user login name, required for access to the destination Site identified in the dest_address parameter.
-x_data_code (-xdc) Data code	Transferred file data code	One of: • B = Binary • A = ASCII • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E = EBCDIC • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • ISO = ISO • OEMPC = OEMPC • U = Undefined

Parameter	Meaning	Value
-x_rec_fmt (-xrf) <i>Record format</i>	Transferred file record format	One of: • F = Fixed • V = Variable • T = Text (OFTP only) • S = Stream • U = Undefined
-x_rec_len (-xrl) <i>Record length</i>	Transferred file record length	Integer: 0 - 32768
-mail_xpriority (-mailxprio)	SMTP protocol only (with or without S/MIME) - the value for X-Priority mail header (SEPAmail standard)	Integer: 1 - 5
-sigalgo (-sga)	Signature and hash algorithm used in payload signature computation. Considered only when <code>payload_integrity_option</code> is set to yes. See Working with Payload integrity .	SHA256withRSA
-sigfile (-sgf)	File containing payload signature, in binary format. Considered only when <code>payload_integrity_option</code> is set to yes.	

Additional Transfer Request parameters for OFTP

The following table contains **peltrans** command parameters that are specific to OFTP file security (OFTP R2.0 only). The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Meaning	Value
-sfidsign <i>Signed EERP</i>	Is signed EERP required?	• Y • N (default)
-sfiddesc <i>Virtual file description</i>	Virtual file description	Free text
-trade_format (-tradeformat) <i>Format</i>	Trade format	CMS (Cryptographic Message Syntax)
-trade_org (-tradeorg) <i>Originator partner</i>	Original Trading Partner (sender)	Maximum: 128 characters
-trade_dest (-tradedest) <i>Destination partner</i>	Destination Trading Partner (receiver)	Maximum: 128 characters
-trade_sign (-tradesign) <i>Is signing</i>	Is file signature required?	• Y • N
-trade_sign_algo (-tradesign_algo) <i>Is signing</i>	Signing algorithm	SHA1withRSA
-trade_encrypt (-tradeencrypt) <i>Is encrypting</i>	Is file encryption required?	• Y • N
-trade_encryption_algo (-tradeencrypt_algo) <i>Is encrypting</i>	Encryption algorithm	• AES256 • 3DES
-trade_compress (-tradecompress) <i>Is compressing</i>	Is file compression required?	• Y • N

Additional Transfer Request parameters for SWIFTNet

The following table contains **peltrans** command parameters that are specific to SWIFTNet transfers. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Meaning	Value
-header_info (-hinf) Header info	Reference to file on disk containing Header Info. This file must respect a specific structure .	
-non_repudiation (-nr) Non repudiation	This option indicates whether SWIFT must keep proof of sending or not. (Availability depends on the service used.)	• Y • N
-payload_type (-pto) Payload type	<i>InterAct only</i> SWIFTNet Interact payload format. On the <i>Initiator</i> side, you can use this parameter to overwrite the Payload type parameter defined in the Trading Partner. On the <i>Responder</i> side, when using the Receipt request synchronized option you can respond with a custom payload type. Even if you saved the request payload as embedded in the Trading Partner, you can answer back with text mode.	• (empty) = undefined • text • embedded
-rcv_msg (None)	<i>FileAct only</i> [0 - 255]: File description, supplied by the sender. Free-text description of the file. [256 - 511]: File information, supplied by the sender. Structured information concerning the transfer. For more details, refer to the SWIFT documentation.	Maximum: 256 + 256 alphanumeric characters.

Parameter	Meaning	Value
-receipt_req_to (-rrto) Receipt DN	<i>FileAct, Real time only</i> If specified, name of the receipt reception address. This is the address to which the receiving partner sends the receipt.	Maximum: 100 alphanumeric characters
-receipt_request_type (rrt) Receipt request type	<i>FileAct, Real time only</i> Type of acknowledgement request	Maximum: 30 alphanumeric characters
-request_type (-rqt) Request type	Type of request (as specified in the XML header).	Maximum: 30 alphanumeric characters
-service (-sc) Service name	SWIFT business service name (as specified in the XML header).	Maximum: 30 alphanumeric characters
-sign_list (-signl) Signature list	Reference to file on disk containing a Signature List. This file must respect a specific structure .	
-sw_add_params (-swaddpar) Additional params	Reference to file on disk containing additional SWIFTNet structures (for example, Header Info and/or Signature List). This file must respect a specific structure .	
-sw_auth_indicator (-ai) Request authorisation notification	SWIFTNet Authorization notification indicator. Select this option to receive Authorization system messages for files or messages sent over Y-copy services .	One of: • Y = yes • N = no (default)

Parameter	Meaning	Value
-sw_copy_indicator (-ci) Request copy	SWIFTNet File or Message Copy service indicator. Select this option to request a copy of the file or message to be sent to a third party defined at the service level.	One of: • Y = yes • N = no (default)
-sw_creation_time (-swcreatm) (None)	<i>Store-and-Forward mode only</i> Use this parameter to set the same creation time as the original transfer when manually resending each lost transfer detected during the SWIFTNet system recovery procedure.	In the format <i>YYYYMMDD HHMMSS</i>
-sw_empty_message_id (-swemptymsgid) (None)	Empty message id parameter. For use with SWIFTNet Possible Duplicate . Use this parameter to prevent Gateway from automatically generating a message id. This is useful if you want to mark a transfer as a Possible Duplicate but <i>not</i> generate a message id for it, because it is unknown. Note: Do not use this parameter in combination with the <code>sw_message_id</code> parameter! If you specify both parameters, then the <code>sw_empty_message_id</code> parameter is not taken into account.	One of: • Y – Prevent Gateway from automatically generating a message id • N – Gateway automatically generates a message id (default)

Parameter	Meaning	Value
-sw_message_id (-swmsgid) Message id	SWIFTNet message id. For use with SWIFTNet Possible Duplicate. The value of this parameter overwrites the message id created by Gateway in the Transfer Request. Enter the SWIFTNet message id of the new transfer. This should be a Universal Unique Identifier (UUID). If the transfer is a Possible Duplicate, enter the message id of the initial transfer.	Maximum: 40 alphanumeric characters
-sw_overdue_time (-swot) Overdue time: Date and Time	SWIFTNet OverdueTime	In the format <i>YYYYMMDD HHMMSS</i>
-sw_pd_indication (-swpdind) Possible duplicate	Possible Duplicate indicator For use with SWIFTNet Possible Duplicate.	• Y - Select this value to indicate to the responder that the transfer is a Possible Duplicate. • N (default)
-sw_public_recipient_list_ind (None)	For use with a SWIFTNet Distribution List (distribution to multiple recipients). Use this parameter to specify whether or not the responder can receive the list of recipients.	

Parameter	Meaning	Value
-trade_delivery_mode (-tradedmo) <i>Delivery mode</i>	Indicates the delivery mode of the file or message. Real time = immediate Store-and-Forward = send and then store in the queue in the SWIFTNet domain.	• RT • SNF • undef (default)
-trade_encrypt_dn (-tradeendn) <i>Encrypt DN</i>	<i>InterAct only</i> If specified, name used for encryption in Initiator mode.	Maximum: 100 alphanumeric characters
-trade_notif_queue (-tradenq) <i>Notif queue</i>	<i>Store-and-Forward mode only</i> Reception queue for Store-and-Forward delivery notifications.	Maximum: 30 alphanumeric characters
-trade_sign_dn (-tradesidn) <i>Sign DN</i>	<i>InterAct only</i> If specified, name used for signing in Initiator mode. Otherwise the securityDN (supplied by SNL) is used	Maximum: 100 alphanumeric characters

Additional Transfer Request parameters for X.400

The following table contains **peltrans** command parameters that are specific to X.400 (X.420) transfers. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Meaning	Value
-content_id (-ctid) <i>Content Id</i>	Uniquely defines the message to be sent.	

Parameter	Meaning	Value
<code>-dn_request (-dnreq)</code> <i>Delivery notification request</i>	Delivery notification request.	• NDN = Non Delivery Notification (default) • DN_NDN = Delivery Notification and Non Delivery Notification
<code>-ipmid_user_id (-ipmiduid)</code> <i>Interpersonal messaging Id</i>	Define the IPM (Interpersonal Messaging) Identifier that uniquely identifies this IPM. Mandatory user relative identifier, which uniquely identifies the IPM.	
<code>-p1.originator</code> <i>Originator</i>	Override the originator X.400 address.	
<code>-p1.recipient</code> <i>Recipient</i>	Override the recipient X.400 address.	

Parameter	Meaning	Value
<code>-p2.bodyPartType</code> Body part type	Define the body part type to be used.	• <code>ia5</code> (default) • <code>g3facsimile</code> • <code>teletex</code> • <code>videotex</code> • <code>encrypted</code> • <code>bilaterallyDefined</code> • <code>externallyDefined</code>
<code>-p2.importance</code> Importance	Indicate the importance of the IPM.	• <code>low</code> • <code>normal</code> (default) • <code>high</code>
<code>-p2.ipmId.user</code>	Define the IPM Identifier that uniquely identifies this IPM. Optional part that is a serialized user O/R address.	
<code>-p2.originator.formalName</code>	Override the formal name part of the O/R Descriptor of the message originator.	

Parameter	Meaning	Value
<code>-p2.originator.freeFormalName</code>	Override the free formal name part of the O/R Descriptor of the message originator.	
<code>-p2.originator.telephoneNumber</code>	Override the telephone number part of the O/R Descriptor of the message originator.	
<code>-p2.primaryRecipient.recipient.freeFormalName</code>	Override the formal name part of the O/R Descriptor of the primary recipient.	
<code>-p2.primaryRecipient.recipient.freeFormalName</code>	Override the free formal name part of the O/R Descriptor of the primary recipient.	

Parameter	Meaning	Value
- <code>p2.primaryRecipient.recipient.telephoneNumber</code>	Override the telephone number part of the O/R Descriptor of the primary recipient.	
<code>-p2.repliedToIpmId.user</code>	Set the IPM identifier which identifies the message to which this message is the reply. Optional part.	
- <code>p2.repliedToIpmId.userRelativeId</code> <i>Replied to IPM Id</i>	Set the IPM identifier which identifies the message to which this message is the reply. Mandatory user relative identifier, which uniquely identifies the IPM.	

Parameter	Meaning	Value
<code>-p2.replyDateTime</code>	Request (but not demand) a time when any replies to the present IPM should have been originated. It comprises a date and time.	
<code>-p2.sensitivity</code> Sensitivity	Indicate the sensitivity of the IPM.	• <code>unspecified</code> (default) • <code>personal</code> • <code>private</code> • <code>companyConfidential</code>
<code>-priority (-prio)</code> Message transfer system priority	Defines the priority level for the message.	• 0 = High priority • 1 = Medium priority • 2 = Low priority
<code>-recipient_receipt_notification_request (-rcptrnr)</code> Notification request	Receipt request.	• None (default) • NRN = Non Receipt Notification • RN_NRN = Receipt Notification and Non Receipt Notification
<code>-recipient_reply_request (-rcptrr)</code> Reply request	Reply request.	• N (default) • Y
<code>-smtp_subject (-subject)</code> Subject	Subject of the message.	

Additional Transfer Request parameters for Diffusion Lists

Parameter	Meaning
-nb_total	Total number of Transfer Requests
-nb_ended	Number of Transfer Requests in ended state
-nb_to_begin	Number of Transfer Requests in to_begin state
-nb_suspended	Number of Transfer Requests in suspended state
-nb_canceled	Number of Transfer Requests in canceled state
-nb_frozen	Number of Transfer Requests in frozen state
-nb_progressing/servicing	Number of Transfer Requests in progressing or servicing

Purging Transfers: pelpur parameters

The following table contains parameters you can use with the **pelpur** command or **pelpur xferPurge** subcommand. (Both forms are functional and have the same result).

Parameter	Meaning	Value
-all (-a) (None)	Purges all ended Transfer Requests (state <i>Ended</i> , <i>Canceled</i> and <i>Acked</i>)	No value
-appli_name (-ap) Application name	Application name	Maximum: 25 alphanumeric characters
-dest_alias (-da) Destination alias	Destination Site alias	Maximum: 31 alphanumeric characters
-direction (-dir) Direction	Transfer direction	One of: • I = Input • O = Output

Parameter	Meaning	Value
-file_name (-fn) File Name	Protocol file name For FTP and HTTP DIR type Transfers, this parameter defines a filter on the file names that the server returns. For example: *.txt returns all .txt files in the directory that you specify.	Maximum: 76 alphanumeric characters
-file_type (-ft) File type	<i>PeSIT HS only</i> File type	Maximum: 16 alphanumeric characters
-from_date (-fd) <i>(None)</i>	Earliest date for the Transfer	Format: YYYYMMDD HHMMSS
-ident_inf (-id_inf) <i>(None)</i>	Minimum value of the Transfer identifiers to select	Integer
-ident_sup (-id_sup) <i>(None)</i>	Maximum value of the Transfer identifiers to select	Integer
-keep_last_day (-kld) <i>(None)</i>	Do not purge Transfers from the <i>ddd</i> previous days. Both the transfer <i>create_date</i> and <i>end_date</i> (if exists) are used for matching. They must be older than the <i>kld</i> value. The <i>Kld</i> option is not applied as a filter over the "in memory" transfer list but iterates through the mailbox to select the matching transfers. This may impact performance when there is a large mailbox.	Integer
-keep_mailbox_entries (-ke) <i>(None)</i>	Skip purging the transfers from mailbox.	No value

Parameter	Meaning	Value
<code>-force (-f)</code>	Force files removal, ignoring value of <code>purge_option</code> parameter from transfer.	No value
<code>-list_route_state (-lrs)</code> (None)	Select, purge or display all Transfers with the specified routing status	One of: • A = Acked • R = Routed • F = Failed • T = To Route • N = Not Routed • E = Empy • P = Processing
<code>-list_trans_state (-lts)</code> (None)	Select, purge or display all Transfers with the specified status	One of: • F = Frozen • D = Delayed • B = To Begin • P = Processing • R = To Restart • S = Suspended • E = Ended • C = Canceled • I = Interrupted • V = serVicing • A = Acked (OFTP, PeSIT, FTP) • T = CreaTed • Z = Deleted • W = Ended_to_ack • G = To_SiGn • U = Nacked_User
<code>-local_agent (-la)</code> Local agent alias	Alias of the associated Local Site	Maximum: 31 alphanumeric characters

Parameter	Meaning	Value
<code>-local_ident (-i)</code> Local Identifier	Local Transfer identifier	Integer
<code>-master (-ma)</code> Diffusion List: Local identifier	Diffusion List Request identifier	Integer
<code>-mode (-m)</code> Mode	Connection mode for the Transfer	One of: • I = Initiator • R = Responder
<code>-no_archive (-no_arc)</code> (None)	Purge the Transfer from the Mailbox without creating a Transfer archive file	No value
<code>-org_alias (-oa)</code> Originator alias	Originator Site alias	Maximum: 31 alphanumeric characters

Parameter	Meaning	Value
-protocol (-pr) Protocol	Protocol	One of: • PEL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • FTP • HTTP • FTP_HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_HTTP • RN_POP3 • RN_SMTP
-purge_file (-pf) <i>(None)</i>	Purges the transferred file	No value
-purge_tmp_file (-ptf) <i>(None)</i>	Purge only the temporary file related to the transfers.	No value
-purge_xpr (-px) <i>(None)</i>	Purges the resolved script file and its output files for the Transfer	No value
-remote_agent (-ra) Remote agent alias	Alias of the associated Remote Site	Maximum: 31 alphanumeric characters

Parameter	Meaning	Value
-req_name (-rn) (None)	Name of the purge request that will be used in the log file to print statistics	Maximum: 25 alphanumeric characters Default = PELPUR
-sequence_inf (-seq_inf) (None)	Lowest number in range of Transfers to select, purge or display	Integer
-sequence_sup (-seq_sup) (None)	Highest number in range of Transfers to select, purge or display	Integer
-to_date (-td) (None)	Latest date in the range of Transfers to select, purge or display. The only transfer parameter that is checked is <code>create_date</code> .	Format: YYYYMMDD HHMMSS
-type (-ty) Type	Transfer type	One of: • TRANS • LIST • POLL • LOTS • EERP • SELECT • MSG • DIR • RECEIPT
-user_state (-usr_st) User state	User state field managed by the user application. Gateway does not interpret this field.	One character
-username (-un) User name	User name	Maximum: 31 alphanumeric characters
-yday_inf (-yd_inf) (None)	Start of the date range	Format: <i>ddd</i> 1 - 366

Parameter	Meaning	Value
-yday_sup (yd_sup) (None)	End of the date range	Format: <i>ddd</i> 1 - 366

The following table contains parameters you can use with the `pelpur clean_swha_db` subcommand. The parameters refer only to SWIFTNet transfer entries saved in the HA database in the context of SWIFTNet High Availability.

Parameter	Meaning	Value
-gatewayID (-gtwID)	Enter a gateway id as selection criteria for the purge operation.	Integer 0-15
-fromGatewayID (-fgtwID)]	Enter a gateway id from which to start the selection, to select all fields with the gateway id greater or equal to it	Integer 0-15
-toGatewayID (-tgtwID)	Enter a gateway id from which to start the selection, to select all fields with the gateway id less or equal to it.	Integer 0-15
-transferID (-trID)	Enter a transfer id as selection criteria for the purge operation.	Integer
-fromTransferID (-ftrID)	Enter a transfer id from which to start the selection, to select all fields with the transfer id greater or equal to it.	Integer
-toTransferID (-ttrID)	Enter a transfer id from which to start the selection, to select all fields with the transfer id less or equal to it.	Integer
-flowType (-flTp)	Enter the flow type as selection criteria for the purge operation.	One of: • INTERACT • FILEACT

Parameter	Meaning	Value
-deliveryMode (-delMode)	Enter the transfer delivery mode as selection criteria for the purge operation.	One of: • RT • SNF
-localPartner (-lP)	Enter the local partner alias as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-remotePartner (-rP)	Enter the remote partner alias as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-messageID (-mID)	Enter the message identification number as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-service (-sv)	Enter the service name as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-requestType (-rqTp)	Enter the request type as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-requestorDN (-rquDN)	Enter the requester DN as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-responderDN (-rpDN)	Enter the responder DN as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-logicalName (-lgName)	Enter the logical name of the file as selection criteria for the purge operation.	Maximum: 100 alphanumeric characters
-createDate (-cDate)	Enter the create date of the transfer as selection criteria for the purge operation.	Format: YYYYMMDD

Parameter	Meaning	Value
<code>-fromCreateDate (-fcDate)</code>	Enter the start create date of the transfer. It will match the transfers with create date equal to this one or after this date.	Format: YYYYMMDD
<code>-toCreateDate (-tcDate)</code>	Enter the start create date of the transfer. It will match the transfers with create date equal with this one or before this date.	Format: YYYYMMDD
<code>-modifyDate (-mDate)</code>	Enter the date when transfer was last modified as selection criteria for the purge operation.	Format: YYYYMMDD
<code>-fromModifyDate (-fmDate)</code>	Enter the date when the transfer was last modified. It will match the transfers with modify date equal with this one or after this date.	Format: YYYYMMDD
<code>-toModifyDate (-tmDate)</code>	Enter the date when the transfer was last modified. It will match the transfers with modify date equal with this one or before this date.	Format: YYYYMMDD

Note To purge a transfer, the SWIFTNet High Availability feature has to be activated and connectivity with HA database must be functional.

Displaying Transfers: peldsp command parameters

The following table contains parameters you can use with the `peldsp status_trans` and `peldsp select_trans` commands.

Parameter	Meaning	Value
-appli (-ap) Application	Application name	Maximum: 25 alphanumeric characters
-dest_alias (-da) Destination alias	Destination Site alias	Maximum: 31 alphanumeric characters
-destination (-dest) Destination	Destination protocol identifier	Maximum: 31 alphanumeric characters
-direction (-dir) Direction	Transfer direction	One of: • I = Input • O = Output
-file_name (-fn) File Name	Protocol file name For FTP and HTTP DIR type Transfers, this parameter defines a filter on the file names that the server returns. For example: *.txt returns all .txt files in the directory that you specify.	Maximum: 76 alphanumeric characters
-file_type (-ft) File type	<i>PeSIT HS only</i> File type	Maximum: 16 alphanumeric characters
-from_date (-fd) <i>(None)</i>	Earliest date for the Transfer	Format: YYYYMMDD HHMMSS
-ident_inf (-id_inf) <i>(None)</i>	Minimum value of the Transfer identifiers to select	Integer
-ident_sup (-id_sup) <i>(None)</i>	Maximum value of the Transfer identifiers to select	Integer

Parameter	Meaning	Value
-list_route_state (-lrs) <i>(None)</i>	Select, purge or display all Transfers with the specified routing status	One of: • A = Acked • R = Routed • F = Failed • T = To Route • N = Not Routed • E = Empy • P = Processing
-list_trans_state (-lts) <i>(None)</i>	Select, purge or display all Transfers with the specified status	One of: • F = Frozen • D = Delayed • B = To_Begin • P = Processing • R = To_Restart • S = Suspended • E = Ended • C = Canceled • I = Interrupted • V = serVicing • A = Acked (OFTP, PeSIT, FTP) • T = CreaTed • Z = Deleted • W = Ended_to_ack • G = To_SiGn • U = Nacked_User
-local_agent (-la) Local agent alias	Alias of the associated Local Site	Maximum: 31 alphanumeric characters
-local_ident (-i) Local Identifier	Local Transfer identifier	Integer

Parameter	Meaning	Value
-master (-ma) <i>Diffusion List: Local identifier</i>	Diffusion List Request identifier	Integer
-mode (-m) <i>Mode</i>	Connection mode for the Transfer	One of: • I = Initiator • R = Responder
-model (-ml) <i>Model name</i>	Model used to submit the Transfer Request	Maximum: 25 alphanumeric characters
-org_alias (-oa) <i>Originator alias</i>	Originator Site alias	Maximum: 31 alphanumeric characters
-originator (-org) <i>Originator</i>	Originator protocol identifier	Maximum: 31 alphanumeric characters
-param1 (-p1) <i>Param 1</i>	User parameter 1	Maximum: 40 alphanumeric characters
-param2 (-p2) <i>Param 2</i>	User parameter 2	Maximum: 40 alphanumeric characters
-permanent (-perm) <i>Permanent</i>	Set to Yes to make the Transfer Request permanently available	Y/N
-proto_ident (-pid) <i>(None)</i>	Site protocol name	Maximum: 25 alphanumeric characters

Parameter	Meaning	Value
-protocol (-pr) Protocol	Protocol	One of: • PEL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • FTP • HTTP • FTP_HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_HTTP • RN_POP3 • RN_SMTP
-remote agent (-ra) Remote agent alias	Alias of the associated Remote Site	Maximum: 31 alphanumeric characters
-sequence_inf (-seq_inf) <i>(None)</i>	Lowest number in range of Transfers to select, purge or display	Integer
-sequence_sup (-seq_sup) <i>(None)</i>	Highest number in range of Transfers to select, purge or display	Integer

Parameter	Meaning	Value
-sw_marked_by_system_recovery (-swmksysrec)	Selection filter for SWIFTNet transfers marked during system recovery.	• U = Undefined (default) • Y = Transfers marked by system recovery • N = Transfers that were not marked
-to_date (-td) (None)	Latest date in the range of Transfers to select, purge or display.	In the format <i>YYYYMMDD HHMMSS</i>
-trade_state (-ts)	Processing state of an EDIINT transfer	One of: • TO_PROCESS • PROCESSING • PROCESSED • ACKED • NACKED • ERROR
-type (-ty) Type	Transfer type	One of: • TRANS • LIST • POLL • LOTS • EERP • SELECT • MSG • DIR • RECEIPT
-user_state (-usr_st) User state	User state field managed by the user application. Gateway does not interpret this field.	One character
-username (-un) User name	User name	Maximum: 31 alphanumeric characters

Parameter	Meaning	Value
<code>-yday_inf</code> (<code>-yd_inf</code>) (None)	Start of the date range	Format: <i>ddd</i> 1 - 366
<code>-yday_sup</code> (<code>-yd_sup</code>) (None)	End of the date range	Format: <i>ddd</i> 1 - 366

Related topics

[Working with Transfers \(command line\)](#)

Axway Gateway: Managing Transfers

Virtual directories

About Virtual File Directories

[About the Virtual File Directory](#)

[VFD running modes](#)

[Virtual File Directory tree](#)

[Mounted directories](#)

[Real File Directory tree](#)

[VFD name length limitations](#)

[VFD concurrent listing on page 382](#)

[Working with Virtual File Directories](#)

About the Virtual File Directory

Gateway can act as a file server for FTP, SFTP or HTTP transfers. You can use *Virtual File Directories* to control the views that users have from client machines of files stored on Gateway.

The Virtual File Directory (VFD) is a virtual device that enables you to provide an administrator-controlled view of data organization to specific users via a file directory tree. The VFD comprises directories and files.

When you start Gateway for the first time, it creates a root directory automatically. All new directories that you create subsequently belong to the root directory and can contain sub-directories or files.

You can only use the VFD:

- In Responder mode
- With [FTP](#), [SFTP](#) and [HTTP](#) file transfer protocols, where file organization in directories is meaningful

Note: This topic describes the VFD, but does not describe how to assign VFD access via CGate and CGateGroup objects. Refer to [CGates: Transfer phase](#).

VFD running modes

The VFD has two operating modes that you can run separately or simultaneously:

- **VFD (Virtual File Directory) mode:** The VFD is a logical representation of directories and files (**VFD Files**). The logical files in the VFD represent a duplication of a given Transfer Request extract in the Gateway Mailbox. From the Gateway client side (FTP, SFTP or HTTP clients), you can view the logical representation of these files, but cannot directly access them.
- **RFD (Real File Directory) mode:** You can use a mount mechanism to link the logical file system (VFD) to the physical file system (RFD) on the Gateway host machine. The files seen from the client side are then *real*/files that are located on the Gateway host machine. These files are optionally associated with a Gateway Mailbox Transfer.

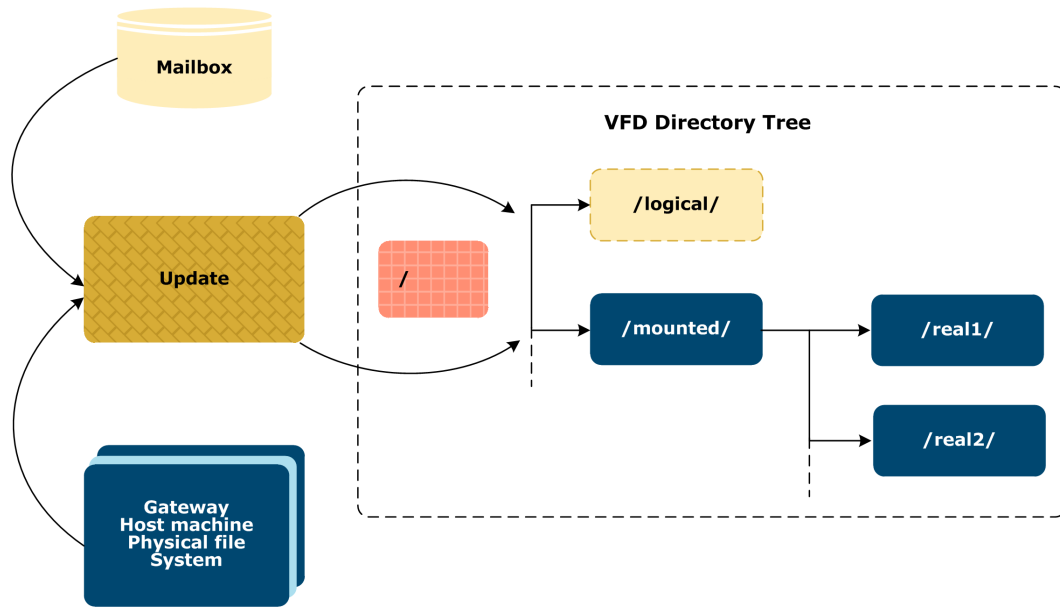
Directory contents

The contents of a VFD directory depend on its associated running mode.

- Logical directories contain:
 - Logical sub-directories
 - VFD Items (files)
 - Mounted sub-directories (that point to a real directory tree)
- Real directories and mounted directories contain:
 - Real sub-directories
 - Real files

Note: You can define the root directory as a mount point, but in this case it cannot contain any logical objects (file or directory).

The following diagram illustrates the Virtual File Directory tree.



Virtual File Directory tree

In virtual mode, directories and files are logical representations of real directories and files.

Virtual directories

From an FTP, SFTP or HTTP client view, the Gateway virtual directory tree works in the same way as any other directory tree provided by standard FTP, SFTP or HTTP servers. Virtual directories can contain sub-directories (virtual and mounted) and VFD Items. Virtual directories have the following attributes:

- Comment field
- Alias (a unique identifier independent of the directory location)
- User attribute (provided to the Gateway exits during the transfer phase)

VFD files

A VFD file is a duplicate extract of a Gateway Mailbox record that is directly linked to a Transfer Request. Since VFD files are only meaningful within Gateway, you can only modify them via Gateway. From a client view, a VFD file is a file that is linked to a record in the Gateway Mailbox. Its availability to a client depends on its:

- Protocol attributes (protocol, origin, destination, and so on)
- Direction (OUT)

The logical part of the VFD is updated depending on Gateway Mailbox modifications (VFD Item creation, update, deletion, and so on).

Note For security concerns, the way that the put-at-disposal transfers are listed is limited for the specified user (by Remote Site), or group of users (by Template Site). It is no longer possible to have a login with a physical remote site and list the put-at-disposal transfers for a Template remote site, such as TSFTP.

Downloading VFD files

You can download a VFD file from a directory if:

- The VFD file exists in the directory (VFD file created via a Transfer Request)
- The corresponding Gateway Mailbox Transfer direction is `OUT` and its state is `TO_BEGIN`
- The protocol parameters of the associated Gateway Mailbox record match those of the connected client

When the file transfer is terminated, the Gateway Mailbox state becomes `ENDED` and the file is no longer available for download by any other client. To make a file transfer available on a permanent basis for multiple client downloads, set the Transfer permanent flag to **YES**.

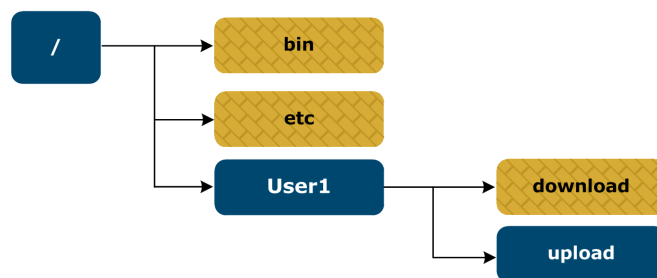
Uploading VFD files

When Gateway receives a file upload request, it checks that the reception directory exists, confirms that the user is authorized to write to the directory, and creates a record in the Mailbox. When Gateway updates the Mailbox, it creates a new VFD file in the corresponding directory. When the reception directory is virtual, this file is not available for any other client because no outgoing request is posted to make this download possible.

VFD naming conventions

The virtual part of the VFD is referenced using the UNIX path syntax. The absolute path of a VFD Item (or a virtual directory) is the concatenation of its parent directory names separated by a "/" character, followed by the name of the current VFD Item (or directory).

In the following diagram, the VFD absolute path for the `upload` directory is: `/user1/upload`.



VFD file name display option

For FTP and SFTP you can determine how to display the file name for each VFD. You can choose one of the following options:

- `s.XferIdent.filename` (default)
- File name only

VFD file name duplication policy

If you set the [VFD file name display option](#) to *file name only*, you also need to tell Gateway how to manage duplicate file names.

For FTP, HTTP and SFTP you can determine how to manage duplicate file names for each VFD. You can choose one of the following options:

- **allow** duplicate file names (default)
- **forbid** duplicate file names. A new transfer request is refused if the file name already exists in the same virtual directory.
- **override** an existing transfer request with a newer one if the file name already exists in the same virtual directory. However, if the old transfer cannot be canceled (update failed), the new request is rejected.

Mounted directories

A mounted directory is a virtual directory that serves as a logical link between the virtual directory tree and the physical file system on the Gateway host machine. Mounted directories have all the same attributes as virtual directories. In addition, mounted directories include a Mailbox recording option. This option enables you to specify whether or not to record all operations on the mounted directory in the Gateway Mailbox.

The mounted directory points to a *real* sub-directory via the absolute path of the directory on the physical file system.

Mounted directory contents

If the physical directory `C:\Temp` contains the files `meeting.txt` and `meeting.ppt`, then from a client connected to Gateway, the `/download` directory also contains the two files `meeting.txt` and `meeting.ppt`.



Real File Directory tree

In **RFD** (Real File Directory) mode, both directories and files are real and are located in the physical file system on the Gateway host machine. You can only manage these files via the corresponding operating system tools (copy, and so on). The RFD mechanism enables you to:

- Integrate the generic data presentation of most FTP, SFTP and HTTP servers into Gateway
- Optionally, use the file transfer phase tracing process

Gateway uses the RFD to provide the client with the image of the directory tree, based on the current state of the directory on the Gateway host machine.

The Gateway administrator must:

- Manually create the required directory tree on the Gateway host machine
- Link it with the VFD using a mount point

Real directories

RFD directories are not represented internally in Gateway. Since these directories are part of the local file system, you cannot manage them using Gateway tools. An RFD directory inherits its attributes from the logical mounted directory that points to the current directory.

Real directories can contain real sub-directories and real files.

Real files

From an FTP, SFTP or HTTP client view, a real file is presented as if it were directly accessible on the Gateway host machine. The transfer phase mechanisms for a real file depend on the value of the `mbx_recording` (**Monitoring**) attribute of its parent directory. This value is inherited from the associated VFD mounted directory.

When you create the VFD, if you set the `mbx_recording` parameter to:

- YES: the download or upload request dynamically creates a Mailbox record that Gateway updates during the data transfer phase
- NO: Gateway does not create a record in the Mailbox and transfers data directly

Files that you upload to or download from the Gateway host machine remain available for subsequent operations:

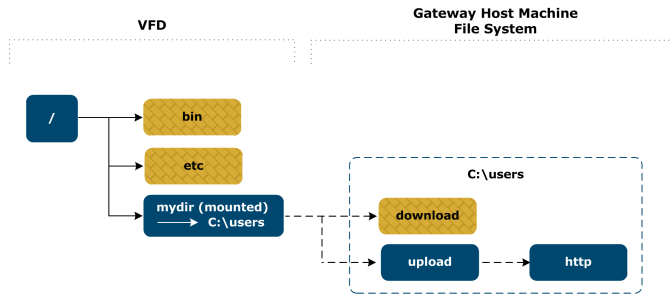
- Downloaded files are opened in read only mode and remain unchanged on the Gateway host machine
- Uploaded files are physically written to the Gateway host machine file system

RFD naming conventions

RFD file and directory naming conventions are as follows:

- The beginning of the absolute file path, up to the mount point, is built as described in [VFD naming conventions](#).
- The end of the absolute path comprises the relative path from the mount point to the designated object (the sub-directory names are separated by the "/" character regardless of the local operating system: UNIX or Windows).

VFD real directory absolute path



The physical directory `C:\users\upload\http` is referenced in the VFD using the following absolute path: `/mydir/upload/http`, where:

- `/mydir` is the VFD absolute path up to the mount point
- `/upload/http` is the relative path from the mount point (sub-directory names are separated by "/")

VFD name length limitations

All objects in the VFD are subjected to name length limitations. Individual directory names and file names are restricted to 127 characters, while the maximum length of a directory path is 255 characters.

As such, it is not possible to create a virtual directory node with a name longer than 127 characters, or in such a way that the absolute path of the new directory is longer than 255 characters.

Mounted RFD directories, subdirectories and files that exceed these limitations will not be handled correctly by the VFD mechanism. They will not appear in the Navigator view or in the output of the `vfd dsp list_dir` command. Furthermore they will not be made available to clients connecting to Gateway.

As an exception, Mailbox entries (as opposed to RFD mapped files) have a file name length limit of 255 characters inherited from the transfer properties.

VFD concurrent listing

You can set a limit to the number of concurrent listings for each protocol, using the `max_listings` parameter. This parameter can be controlled using the `peluconf` command:

```
peluconf set -s ft_ftp max_listing xxx
```

(values from 0 to 999, where 0 means "no limit")

This parameter is available for the following protocols: FTP, HTTP, SFTP, AS2, AS3, RN_HTTP.

[Overview: Virtual File Directories](#)

[Working with Virtual File Directories](#)

[Working with Virtual File Directories \(command line\)](#)

[Virtual File Directories: Parameters List](#)[Working with File Transfer Requests](#)[Transfers: Parameters List](#)

Axway Gateway: Managing Transfers

Working with Virtual File Directories

[Creating a new VFD](#)[Viewing VFD properties](#)[Modifying VFD properties](#)[Deleting a VFD](#)[Renaming a VFD](#)[Mounting a VFD](#)[Unmounting a VFD](#)

Creating a new VFD

To create a new VFD:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click **VFD**, and then select **New** from the context menu.
Gateway opens the *New Virtual File Directory* window.
3. Complete the fields of the *New Virtual File Directory* window:

Field	Description
PathName	When you create a new directory, Gateway automatically completes the path name field with the path of the parent directory. The absolute path of the root directory is "/". The total length of this parameter is limited to 255 characters (or slightly lower in some cases). Individual directory nodes in the path are limited to 127 characters.
Name	Enter the name of the directory. Maximum: 255 alphanumeric characters.

Field	Description
Alias	<i>Optional</i> Enter a VFD directory alias. Use the alias to designate a specific directory within the VFD, regardless of whether it was moved or renamed. You can either reference the VFD via the alias name or via the path name. The value that you enter in this field must be unique in the whole VFD. Maximum: 24 alphanumeric characters.
Comments	Enter a free-text description of the directory. Maximum: 80 alphanumeric characters.
Monitoring	<i>Only used in RFD mode.</i> Check this option to record Transfer operations on this directory in the Gateway Mailbox.
Mount name	Enter the absolute path of the physical directory to be mounted. The total length of this parameter is limited to 255 characters. Individual directory nodes in the path are limited to 127 characters.
User data	Enter a user message associated with the current VFD Item. Maximum: 80 alphanumeric characters.
File name display option	Select one of the following: • Gateway specific: Display VFD file names in the format: <code>s.XferIdent.filename</code> • Standard: Display the VFD file name only Refer to VFD file name display option.
File name duplication policy	Only available if File name display option is set to <i>Standard</i>. Select one of the following: • Duplicate: Allow duplicate file names • Forbid: Forbid duplicate file names. If the file name already exists, the new transfer request is not accepted. • Override: Override the old file name. If the file name already exists, the old transfer is canceled and the new one is added. Refer to VFD file name duplication policy.

Viewing VFD properties

To view the properties of an existing VFD in a read-only window:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. At the **VFD** node, click  to expand and display the existing VFD structure.
3. When you click a VFD, in the right frame Gateway displays any sub-directories that comprise that VFD.
4. In the right frame, right-click the VFD for which you want to view the properties. Then select **View...** from the context menu.

Gateway opens the *VFD Directory* window, containing a read-only display of the VFD properties.

5. After viewing the VFD properties, click **Close...**

Modifying VFD properties

To modify the properties of an existing VFD:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. At the **VFD** node, click  to expand and display the existing VFD structure.
3. Right-click the VFD you for which you want to modify the properties, then select **Properties...** from the context menu.

Gateway opens the *VFD Properties* window. This window contains the same fields as the *New Virtual File Directory* window described above. The properties that you cannot modify in this window are grayed out.

4. Modify the VFD properties as required.
5. Select **OK** to confirm and complete the modification procedure.
6. To refresh the display and view the changes, press **F5**.

Deleting a VFD

To delete an existing VFD:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. At the **VFD** node, click  to expand and display the existing VFD structure.
3. Right-click the VFD node you want to delete. Then select **Delete...** from the context menu.

Gateway displays a confirmation dialog box.

Note: You cannot delete a VFD node that contains any sub-directories. To delete a node or a directory that contains sub-directories, first delete all directories it contains.

4. In the confirmation dialog box, select **OK** to confirm and complete the delete process.
Alternately, select **NO** or **Cancel** to abandon the process without deleting the selected directory.

When you select **OK**, Gateway deletes the selected VFD and updates the display to take into account the change.

Renaming a VFD

To rename an existing VFD:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. At the **VFD** node, click  to expand and display the existing VFD structure.
3. When you click a VFD in the left frame, in the right frame Gateway displays any sub-directories that comprise that VFD.
4. In the right frame, right-click the VFD that you want to rename. Then select **Rename...** from the context menu.

Gateway highlights the current name of the VFD in an editable text frame.

5. Enter a new name for the VFD, and then click outside of the text frame, anywhere in the Gateway main window.

Gateway updates the displayed name of the VFD.

Mounting a VFD

When you *mount a VFD*, you link it to a Real File Directory.

To mount an existing VFD:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. At the **VFD** node, click  to expand and display the existing VFD structure.
3. When you click a VFD in the left frame, in the right frame Gateway displays any sub-directories that comprise that VFD.
4. In the right frame, right-click the VFD that you want to mount. Then select **Mount...** from the context menu.

Gateway opens the *VFD mount directory* window.

5. In the **Directory** frame, enter the path to the Real File Directory you want to link to your VFD.
6. Select **OK** to confirm and complete the mount procedure.

Gateway mounts the VFD. In the right frame, the icon accompanying the VFD entry contains a yellow square to indicate that the VFD is mounted.

Unmounting a VFD

A *mounted VFD* is a VFD that is linked to a Real File Directory.

To unmount a mounted VFD:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. At the **VFD** node, click  to expand and display the existing VFD structure.
3. When you click a VFD in the left frame, in the right frame Gateway displays any sub-directories that comprise that VFD.
4. In the right frame, right-click the VFD that you want to mount. Then select **Unmount...** from the context menu.
Gateway unmounts the VFD.

Axway Gateway: Managing Transfers

Working with Virtual File Directories (command line)

Managing VFD directories:

- [vfdadm create dir](#)
- [vfdadm update dir](#)
- [vfdadm move dir](#)
- [vfdadm delete dir](#)
- [vfddsp display dir](#)
- [vfddsp list dir](#)

Managing VFD files:

- [vfddsp display file](#)

Exporting and Importing VFDs:

- [vfdbase export](#)
- [vfdbase import](#)

Managing VFD directories

This section describes the Gateway commands you can use to manage VFD directories.

Creating a directory: vfdadm create_dir

Syntax	<code>vfdadm create_dir {-dir_path} [optional parameters - see parameters list below]</code>

Description

Use this command to create a new directory and set its attributes. The directory that you create can be either a virtual directory or a mounted directory. A mounted directory is associated with a real directory located in the Gateway host File System.

The absolute path of the directory that you create is a ***mandatory*** parameter.

Do not create the VFD root directory manually. Gateway automatically creates the root directory when you access the VFD for the first time. The absolute path of the root directory is: `"/`.

Parameters*Mandatory*

`-dir_path (-dp)` : Enter the absolute path of the directory you are creating.

`-alias (-a)` : VFD directory alias.

Enter a name for the directory, unique for the whole VFD.

`-comments (-cts)` : Enter a comment about the VFD directory.

Maximum: 80 alphanumeric characters.

`-mount_point (-mp)` : Enter a value to determine whether or not to mount the current directory:

- **Y**
- **N**

`-mount_dir_path (-mdp)` : Enter the absolute path of the physical directory (RFD) to be mounted.

`-user_attribute (-ua)` : Enter a user parameter associated with the current directory.

`-mbx_recording (-mr)` : In RFD mode only, enter the Gateway Mailbox recording option. This option indicates whether or not to trace the transfer operations on the current directory (downloading and uploading) in the Gateway Mailbox.

- **Y**
- **N**

`-filename_display_option (-fdo)` : Enter a value to indicate how file names are displayed:

- **Standard** = only the file name is displayed
- **XFB** = the file name is displayed in the format:
s.XferIdent.filename (default)

For more information, refer to [VFD file name display option](#).

`-filename_duplicate_policy (-fdp)` : Enter a value to indicate how duplicate file names are managed:

- **duplicate** = duplicate file names are allowed (default)
- **forbid** = if the file name already exists, the new transfer request is not accepted
- **override** = if the file name already exists, the old transfer is canceled and the new one is added

For more information, refer to [VFD file name duplication policy](#).

Example	The following command creates a virtual sub-directory of the "/" directory (automatically created). Its VFD absolute path is /Download and it does not have any associated aliases. <pre>vfdadm create_dir -dir_path "/Download" -mount_point "N" -comments "users default download directory"</pre>

Modifying a directory: vfdadm update_dir

Syntax	<code>vfdadm update_dir</code> [optional parameters - see parameters list below]
Description	Use this command to update the attributes (including the optional mount point attributes) of an existing VFD directory. You can reference the directory to be updated either via its VFD absolute path or its alias.

Parameters

`-dir_path (-dp)` : Enter the absolute path of the directory you want to modify.

`-alias (-a)` : VFD directory alias.

Enter the name of the directory that you want to modify.

`-new_alias (-na)` : Enter a new directory name for the updated directory.

`-comments (-cts)` : Enter a comment about the VFD directory.

Maximum: 80 alphanumeric characters.

`-mount_point (-mp)` : Enter a value to determine whether or not to mount the current directory:

- **Y**
 - **N**
-

`-mount_dir_path (-mdp)` : Enter the absolute path of the physical directory (RFD) to be mounted.

`-user_attribute (-ua)` : Enter a user parameter associated with the current directory.

`-mbx_recording (-mr)` : In RFD mode only, enter the Gateway Mailbox recording option. This option indicates whether or not to trace the transfer operations on the current directory (downloading and uploading) in the Gateway Mailbox.

- **Y**
 - **N**
-

`-filename_display_option (-fdo)` : Enter a value to indicate how file names are displayed:

- **Standard** = only the file name is displayed
- **XFB** = the file name is displayed in the format:
`s.XferIdent.filename (default)`

For more information, refer to [VFD file name display option](#).

`-filename_duplicate_policy (-fdp)`: Enter a value to indicate how duplicate file names are managed:

- **duplicate** = duplicate file names are allowed (default)
- **forbid** = if the file name already exists, the new transfer request is not accepted
- **override** = if the file name already exists, the old transfer is canceled and the new one is added

For more information, refer to [VFD file name duplication policy](#).

Example

The following command updates the VFD directory whose absolute path is `/Download`. This directory is set as a mounted directory that points to the `C:\temp` directory in the Gateway host file system.

```
vfdadm update_dir
-directory_path "/Download"
-mount_point "Y"
-mount_dir_path "C:\temp"
```

Moving or renaming a directory: vfdadm move_dir

Syntax

```
vfdadm move_dir [-dir_path] [-alias] [-new_dir_path]
```

Description

Use this command to:

- move a VFD directory (or mounted directory) from its current location in the VFD to a new location
- rename a VFD directory
- move and rename a VFD directory

You can use this command on a directory that is empty or not.

Internally, a VFD directory is identified by an ID node which remains the same during the renaming or moving process. This means that when you move or rename a directory, the match between the directory alias and its new location is automatically taken into account.

The command either renames or moves the directory, depending on the new absolute path to the directory:

- If the parent directory of the current directory remains unchanged, the command only renames the directory.
- Otherwise, the command moves the current directory to the specified location.

In addition, when you move a virtual directory:

- If the destination directory already exists, the current directory becomes a sub-directory of the destination directory and its name remains the same.
- If the destination directory does not exist, the current directory:
 - takes a new name (the last element of the specified new location path)
 - takes the absolute path that you specify

You can reference the directory to move or rename either by its VFD absolute path or by its alias.

Parameters

`-dir_path (-dp)` : Enter the absolute path of the directory you want to modify.

`-alias (-a)` : VFD directory alias.

Enter the name of the directory that you want to modify.

`-new_dir_path (-ndp)` : Enter the absolute path of the directory you want to modify.

Example 1

In this example, the `/smith` directory does not already exist.

```
vfdadm move_dir
-dp "/upload"
-ndp "/smith"
```

The command renames the directory from `upload` to `smith`, but does not move the directory. The parent directory of the directory to move remains unchanged (`"/`).

Example 2

In this example, the `/jones` directory already exists.

```
vfdadm move_dir
-dp "/upload"
-ndp "/jones"
```

The `/upload` directory keeps its name but becomes a new sub-directory of the `/jones` directory.

Its new VFD absolute path is now: `/jones/upload`.

Note: If a `/jones/upload` directory already exists before you run the command, an error is returned.

Example 3

In this example, the `/users` directory exists but `/users/smith` does not exist.

```
vfdadm move_dir
-dp "/upload"
-ndp "/users/smith"
```

The command:

- Renames the `/upload` directory as `/smith`
 - Moves the renamed directory and makes it a sub-directory of `/users`
-

Deleting a logical directory: `vfdadm delete_dir`

Syntax

```
vfdadm delete_dir [-dir_path] [-alias]
```

Description

Use this command to delete a VFD logical directory. You can only delete a directory if it is empty: it must not contain any sub-directories or VFD items.

You must enter a value for at least one of the two parameters.

Before you can delete a mounted directory, you must first set the parameters:

- `mount_point` to `N`
 - `mount_dir_path` to blank
-

Parameters	<code>-dir_path (-dp)</code> : Enter the absolute path of the directory you want to delete. <code>-alias (-a)</code> : VFD directory alias. Enter the name of the directory that you want to delete.
Example	This command deletes a logical VFD directory whose absolute path is <code>/Upload</code> <pre>vfdadm delete_dir -directory_path "/Upload"</pre>

Displaying a logical directory: `vfd dsp display_dir`

Syntax	<code>vfd dsp display_dir [-dir_path] [-alias]</code>
Description	Use this command to display the details, or attributes, of a VFD logical directory that is identified by its VFD absolute path or its alias. You must enter at least one of the two parameters. The command returns the VFD logical directory parameters.
Parameters	<code>-dir_path (-dp)</code> : Enter the absolute path of the directory for which you want to display details. <code>-alias (-a)</code> : VFD directory alias. Enter the name of the directory for which you want to display details.
Details for use	Gateway returns a display of display of parameters and values in which each parameter is displayed on a separate line in the format: <pre>name_of_field="value"</pre> Each field name is prefixed with the string <code>vfd_</code> (Virtual Directory).

Example

Enter the following command:

```
vfddsp display_dir  
-dp "/upload"
```

Gateway returns the following `/upload` logical directory attributes:

```
vd_dir_path=/upload  
vd_comments=Default user upload directory  
vd_alias=UPLOAD  
vd_user_attribute=  
vd_mount_dir_path=  
vd_mbx_recording=Y
```

Listing a logical directory: `vfddsp list_dir`

Syntax

```
vfddsp list_dir {-dir_path} [-alias] [-filter]  
[-no_dir] [-no_file]
```

Description

Use this command to list the contents of the directory that you specify.

You must enter values for at least one of the two parameters:

- `-dir_path`
- `-alias`

This command supports filters to help you find the required data. These filters can comprise:

- A type filter (`no_dir` or `no_file`)
- A name pattern (wildcard characters such as `?` and `*` are allowed). The command only displays the elements whose names match the pattern.

Note: This command works with both logical and real directories.

Parameters	<i>Mandatory</i> <code>-dir_path (-dp)</code> : Enter the absolute path of the directory whose attributes you want to display. <code>-alias (-a)</code> : VFD directory alias. Enter the name of the directory whose attributes you want to display. <code>-filter (-f)</code> : Enter a string. Gateway displays the inner components of the current VFD directory whose name matches the filter string. This string can contain <code>?</code> and <code>*</code> wildcard characters. <code>-no_dir (-nd)</code> : <i>(takes no value)</i> Enter this parameter to indicate that sub-directories must not be displayed. <code>-no_file (-nf)</code> : <i>(takes no value)</i> Enter this parameter to indicate that files must not be displayed.
Details for use	The screen displays the directory contents that match the selection criteria that you specify. A value X means that the corresponding attribute is meaningless for the current object. If expected directories or filenames do not appear in the output of the command, make sure that you have respected the VFD name length limitations.
Example	See the following examples.

Example 1

The following example of the `vfddsp list` command displays the contents of the VFD `/upload` directory without any selection criteria.

```
vfddsp list_dir -dp "/upload"
```

Lid	Filename	T	S	Xid	S	Last Modif Date
10	FTP_B	VF	B	10	4	2006/10/24 16.47
11	FTP_B	VF	B	11	4	2006/10/24 16.47
12	FTP_B	VF	B	12	4	2006/10/24 16.47
X	user1	VD	X	X	0	2006/10/24 16.47
X	user2	VD	X	X	0	2006/10/24 16.47
X	tmp	MD	X	X	0	2006/10/24 16.48

Example 2

The following example displays the contents of the same `/upload` directory, but in this case each element must be a directory (`no_file`).

```
vfddsp list_dir -dp "/upload" -nf
```

Lid	Filename	T	S	Xid	S	Last Modif Date
X	user1	VD	X	X	0	2006/10/24 16.47
X	user2	VD	X	X	0	2006/10/24 16.47
X	tmp	MD	X	X	0	2006/10/24 16.48

Managing VFD files

This section describes the operations allowed on VFD files. Gateway performs most of these operations automatically and internally. Operations that involve updating the Gateway Mailbox are not accessible to the end user, for example: creating, updating or deleting VFD files. VFD files are modified depending on updates to the Gateway Mailbox.

Displaying a logical file (VFD file): `vfddsp display_file`

Syntax	<code>vfddsp display_file {-local_ident} [-dir_path] [-alias]</code>
Description	Use this command to display the details of a VFD logical file identified by its VFD parent directory absolute path (or its alias) and its name. The command returns the VFD logical file parameters.
Parameters	<i>Mandatory</i> <code>-local_ident (-li)</code>: Local transfer identifier of the VFD file you want to display. <code>-dir_path (-dp)</code>: Enter the absolute path of the directory whose attributes you want to display. <code>-alias (-a)</code>: VFD directory alias. Enter the name of the directory whose attributes you want to display.
Details for use	Each returned parameter is displayed on a separate line. Each field name is prefixed with the string <code>vf_</code> (Virtual File).

Example

Enter the command:

```
vfddsp display_file -dp "/" -i 14
```

Gateway returns the attributes for the logical file 14. This file is located in the VFD "/" directory:

```
vf_local_ident=14
vf_dir_path=/
vf_date_create=20070122 154603
vf_file_size=0
vf_protocol=HTTP
vf_access_right=-----
vf_direction=0
vf_state=C
vf_originator=GFL660
vf_destination=THTTP
vf_remote_agent=THTTP
vf_file_name=DEFAULT_B
vf_xfer_ident=14
vf_route_state=
vf_user_state=
vf_permanent=N
vf_appli=default_b
vf_param1=
vf_param2=
vf_user_msg=Dummy Message
vf_data_code=B
vf_route_from_xfer=0
vf_route_to_xfer=0
vf_duplicate_from_xfer=0
vf_type=0
```

Exporting and Importing VFDs

This section describes Gateway commands you can use to move VFDs between Gateways.

vfdbase export

Syntax

```
vfdbase export {-output}
```

Description	When Gateway is running, use this command to export Virtual File Directories (VFDs) to an output file. By default, the resulting file is created in the <code><Gateway installation directory>/run_time/tmp</code> directory. You can then edit the file and import the file to Gateway using the vfdbase import command. Note: During the export process, VFD items are lost.
Parameters	-output (-f) : Enter the name of the resulting file on your host machine.
Example	Enter the command: <pre>vfdbase export -f VFDMove</pre> Gateway creates an output file named <code>VFDMove</code> that contains descriptions of all VFD files on your Gateway: <pre>[Overall] Version = 110 [Directory] Type = VIRTUAL_DIR NodeIdent = 1000 ... UserData =</pre>

vfdbase import

Syntax	<pre>vfdbase import {-input} [-discard_mount_ points] [-directories] [-transfers] [-ignore_ warnings]</pre>
Description	Use this command to import the complete set of Virtual File Directories into the Gateway database. The VFDs are described in a file you created using the vfdbase export command.

Parameters	<code>-input (-if)</code> : Enter the name of the file containing VFDs for import. <code>-discard_mount_points (-dmp)</code> : Enter a value to indicate whether or not the mount points are to be discarded on import: • Y = Directories are virtual and do not direct to physical directories. In that case, you must mount them manually. • N = Links are kept between directories (default) <code>-directories (dir)</code> : Enter a value to indicate whether or not the directory is to be imported: • Y (default) • N <code>-transfers (-tr)</code> : Enter a value to indicate whether or not you want Gateway to retrieve and import transfer records from the Mailbox: • Y (default) • N <code>-ignore_warnings (-iw)</code> : Enter a value to indicate whether or not you want to ignore warnings generated during the import process: • Y • N (default)
Details for use	Before importing VFDs: 1. Shut down the target Gateway. 2. On the target Gateway machine, manually delete the data files in the VFD library (<code>p_runtime_dir/vfd</code>). If you do not do this, the import procedure will not succeed and the message "VFD files already exist, delete all files before importing" will be displayed. 3. Create a VFD output file on the originating Gateway, using the vfdbase export command.
Example	Enter the following command (assuming the file containing the importable VFDs is named <code>VFDMove</code>): <pre>vfdbase import -if VFDMove</pre>

[Overview: Virtual File Directories](#)

[Virtual File Directories: Parameters List](#)

Working with Virtual File Directories (command line)

Managing VFD directories:

- [vfdadm create dir](#)
- [vfdadm update dir](#)
- [vfdadm move dir](#)
- [vfdadm delete dir](#)
- [vfddsp display dir](#)
- [vfddsp list dir](#)

Managing VFD files:

- [vfddsp display file](#)

Exporting and Importing VFDs:

- [vfdbase export](#)
- [vfdbase import](#)

Managing VFD directories

This section describes the Gateway commands you can use to manage VFD directories.

Creating a directory: vfdadm create_dir

Syntax	<code>vfdadm create_dir {-dir_path} [optional parameters - see parameters list below]</code>
Description	Use this command to create a new directory and set its attributes. The directory that you create can be either a virtual directory or a mounted directory. A mounted directory is associated with a real directory located in the Gateway host File System. The absolute path of the directory that you create is a mandatory parameter. Do not create the VFD root directory manually. Gateway automatically creates the root directory when you access the VFD for the first time. The absolute path of the root directory is: <code>"/</code>.

Parameters*Mandatory*

`-dir_path (-dp)` : Enter the absolute path of the directory you are creating.

`-alias (-a)` : VFD directory alias.

Enter a name for the directory, unique for the whole VFD.

`-comments (-cts)` : Enter a comment about the VFD directory.

Maximum: 80 alphanumeric characters.

`-mount_point (-mp)` : Enter a value to determine whether or not to mount the current directory:

- **Y**
- **N**

`-mount_dir_path (-mdp)` : Enter the absolute path of the physical directory (RFD) to be mounted.

`-user_attribute (-ua)` : Enter a user parameter associated with the current directory.

`-mbx_recording (-mr)` : In RFD mode only, enter the Gateway Mailbox recording option. This option indicates whether or not to trace the transfer operations on the current directory (downloading and uploading) in the Gateway Mailbox.

- **Y**
- **N**

`-filename_display_option (-fdo)` : Enter a value to indicate how file names are displayed:

- **Standard** = only the file name is displayed
- **XFB** = the file name is displayed in the format:
s.XferIdent.filename (default)

For more information, refer to [VFD file name display option](#).

`-filename_duplicate_policy (-fdp)` : Enter a value to indicate how duplicate file names are managed:

- **duplicate** = duplicate file names are allowed (default)
- **forbid** = if the file name already exists, the new transfer request is not accepted
- **override** = if the file name already exists, the old transfer is canceled and the new one is added

For more information, refer to [VFD file name duplication policy](#).

Example	The following command creates a virtual sub-directory of the "/" directory (automatically created). Its VFD absolute path is /Download and it does not have any associated aliases. <pre>vfdadm create_dir -dir_path "/Download" -mount_point "N" -comments "users default download directory"</pre>

Modifying a directory: vfdadm update_dir

Syntax	<code>vfdadm update_dir</code> [optional parameters - see parameters list below]
Description	Use this command to update the attributes (including the optional mount point attributes) of an existing VFD directory. You can reference the directory to be updated either via its VFD absolute path or its alias.

Parameters

`-dir_path (-dp)` : Enter the absolute path of the directory you want to modify.

`-alias (-a)` : VFD directory alias.

Enter the name of the directory that you want to modify.

`-new_alias (-na)` : Enter a new directory name for the updated directory.

`-comments (-cts)` : Enter a comment about the VFD directory.

Maximum: 80 alphanumeric characters.

`-mount_point (-mp)` : Enter a value to determine whether or not to mount the current directory:

- **Y**
 - **N**
-

`-mount_dir_path (-mdp)` : Enter the absolute path of the physical directory (RFD) to be mounted.

`-user_attribute (-ua)` : Enter a user parameter associated with the current directory.

`-mbx_recording (-mr)` : In RFD mode only, enter the Gateway Mailbox recording option. This option indicates whether or not to trace the transfer operations on the current directory (downloading and uploading) in the Gateway Mailbox.

- **Y**
 - **N**
-

`-filename_display_option (-fdo)` : Enter a value to indicate how file names are displayed:

- **Standard** = only the file name is displayed
- **XFB** = the file name is displayed in the format:
`s.XferIdent.filename` (default)

For more information, refer to [VFD file name display option](#).

`-filename_duplicate_policy (-fdp)`: Enter a value to indicate how duplicate file names are managed:

- **duplicate** = duplicate file names are allowed (default)
- **forbid** = if the file name already exists, the new transfer request is not accepted
- **override** = if the file name already exists, the old transfer is canceled and the new one is added

For more information, refer to [VFD file name duplication policy](#).

Example

The following command updates the VFD directory whose absolute path is `/Download`. This directory is set as a mounted directory that points to the `C:\temp` directory in the Gateway host file system.

```
vfdadm update_dir
-directory_path "/Download"
-mount_point "Y"
-mount_dir_path "C:\temp"
```

Moving or renaming a directory: `vfdadm move_dir`

Syntax

```
vfdadm move_dir [-dir_path] [-alias] [-new_dir_path]
```

Description

Use this command to:

- move a VFD directory (or mounted directory) from its current location in the VFD to a new location
- rename a VFD directory
- move and rename a VFD directory

You can use this command on a directory that is empty or not.

Internally, a VFD directory is identified by an ID node which remains the same during the renaming or moving process. This means that when you move or rename a directory, the match between the directory alias and its new location is automatically taken into account.

The command either renames or moves the directory, depending on the new absolute path to the directory:

- If the parent directory of the current directory remains unchanged, the command only renames the directory.
- Otherwise, the command moves the current directory to the specified location.

In addition, when you move a virtual directory:

- If the destination directory already exists, the current directory becomes a sub-directory of the destination directory and its name remains the same.
- If the destination directory does not exist, the current directory:
 - takes a new name (the last element of the specified new location path)
 - takes the absolute path that you specify

You can reference the directory to move or rename either by its VFD absolute path or by its alias.

Parameters

`-dir_path (-dp)` : Enter the absolute path of the directory you want to modify.

`-alias (-a)` : VFD directory alias.

Enter the name of the directory that you want to modify.

`-new_dir_path (-ndp)` : Enter the absolute path of the directory you want to modify.

Example 1 In this example, the `/smith` directory does not already exist.

```
vfdadm move_dir
-dp "/upload"
-ndp "/smith"
```

The command renames the directory from `upload` to `smith`, but does not move the directory. The parent directory of the directory to move remains unchanged (`/`).

Example 2 In this example, the `/jones` directory already exists.

```
vfdadm move_dir
-dp "/upload"
-ndp "/jones"
```

The `/upload` directory keeps its name but becomes a new sub-directory of the `/jones` directory.

Its new VFD absolute path is now: `/jones/upload`.

Note: If a `/jones/upload` directory already exists before you run the command, an error is returned.

Example 3 In this example, the `/users` directory exists but `/users/smith` does not exist.

```
vfdadm move_dir
-dp "/upload"
-ndp "/users/smith"
```

The command:

- Renames the `/upload` directory as `/smith`
- Moves the renamed directory and makes it a sub-directory of `/users`

Deleting a logical directory: `vfdadm delete_dir`

Syntax `vfdadm delete_dir [-dir_path] [-alias]`

Description Use this command to delete a VFD logical directory. You can only delete a directory if it is empty: it must not contain any sub-directories or VFD items.

You must enter a value for at least one of the two parameters.

Before you can delete a mounted directory, you must first set the parameters:

- `mount_point` to `N`
- `mount_dir_path` to blank

Parameters	<code>-dir_path (-dp)</code> : Enter the absolute path of the directory you want to delete.

<code>-alias (-a)</code> : VFD directory alias. Enter the name of the directory that you want to delete.

Example	This command deletes a logical VFD directory whose absolute path is <code>/Upload</code> <pre>vfdadm delete_dir -directory_path "/Upload"</pre>

Displaying a logical directory: vfddsp display_dir

Syntax	vfddsp display_dir [-dir_path] [-alias]
Description	Use this command to display the details, or attributes, of a VFD logical directory that is identified by its VFD absolute path or its alias. You must enter at least one of the two parameters. The command returns the VFD logical directory parameters.
Parameters	-dir_path (-dp) : Enter the absolute path of the directory for which you want to display details. -alias (-a) : VFD directory alias. Enter the name of the directory for which you want to display details.
Details for use	Gateway returns a display of display of parameters and values in which each parameter is displayed on a separate line in the format: <pre>name_of_field="value"</pre> Each field name is prefixed with the string vd_ (Virtual Directory).
Example	Enter the following command: <pre>vfddsp display_dir -dp "/upload"</pre> Gateway returns the following /upload logical directory attributes: <pre>vd_dir_path=/upload vd_comments=Default user upload directory vd_alias=UPLOAD vd_user_attribute= vd_mount_dir_path= vd_mbx_recording=Y</pre>

Listing a logical directory: vfddsp list_dir

Syntax	vfddsp list_dir {-dir_path} [-alias] [-filter] [-no_dir] [-no_file]

Description	Use this command to list the contents of the directory that you specify. You must enter values for at least one of the two parameters: • <code>-dir_path</code> • <code>-alias</code> This command supports filters to help you find the required data. These filters can comprise: • A type filter (<code>no_dir</code> or <code>no_file</code>) • A name pattern (wildcard characters such as <code>?</code> and <code>*</code> are allowed). The command only displays the elements whose names match the pattern. Note: This command works with both logical and real directories.
Parameters	<i>Mandatory</i> <code>-dir_path (-dp)</code> : Enter the absolute path of the directory whose attributes you want to display. <code>-alias (-a)</code> : VFD directory alias. Enter the name of the directory whose attributes you want to display. <code>-filter (-f)</code> : Enter a string. Gateway displays the inner components of the current VFD directory whose name matches the filter string. This string can contain <code>?</code> and <code>*</code> wildcard characters. <code>-no_dir (-nd)</code> : (<i>takes no value</i>) Enter this parameter to indicate that sub-directories must not be displayed. <code>-no_file (-nf)</code> : (<i>takes no value</i>) Enter this parameter to indicate that files must not be displayed.
Details for use	The screen displays the directory contents that match the selection criteria that you specify. A value X means that the corresponding attribute is meaningless for the current object. If expected directories or filenames do not appear in the output of the command, make sure that you have respected the VFD name length limitations.
Example	See the following examples.

Example 1

The following example of the `vfddsp list` command displays the contents of the VFD `/upload` directory without any selection criteria.

```
vfddsp list_dir -dp "/upload"
```

Lid	Filename	T	S	Xid	S	Last Modif Date
10	FTP_B	VF	B	10	4	2006/10/24 16.47
11	FTP_B	VF	B	11	4	2006/10/24 16.47
12	FTP_B	VF	B	12	4	2006/10/24 16.47
X	user1	VD	X	X	0	2006/10/24 16.47
X	user2	VD	X	X	0	2006/10/24 16.47
X	tmp	MD	X	X	0	2006/10/24 16.48

Example 2

The following example displays the contents of the same `/upload` directory, but in this case each element must be a directory (`no_file`).

```
vfddsp list_dir -dp "/upload" -nf
```

Lid	Filename	T	S	Xid	S	Last Modif Date
X	user1	VD	X	X	0	2006/10/24 16.47
X	user2	VD	X	X	0	2006/10/24 16.47
X	tmp	MD	X	X	0	2006/10/24 16.48

Managing VFD files

This section describes the operations allowed on VFD files. Gateway performs most of these operations automatically and internally. Operations that involve updating the Gateway Mailbox are not accessible to the end user, for example: creating, updating or deleting VFD files. VFD files are modified depending on updates to the Gateway Mailbox.

Displaying a logical file (VFD file): `vfddsp display_file`

Syntax	<code>vfddsp display_file {-local_ident} [-dir_path] [-alias]</code>

Description	Use this command to display the details of a VFD logical file identified by its VFD parent directory absolute path (or its alias) and its name. The command returns the VFD logical file parameters.

Parameters	<i>Mandatory</i> <code>-local_ident (-li)</code>: Local transfer identifier of the VFD file you want to display. <code>-dir_path (-dp)</code>: Enter the absolute path of the directory whose attributes you want to display. <code>-alias (-a)</code>: VFD directory alias. Enter the name of the directory whose attributes you want to display.
Details for use	Each returned parameter is displayed on a separate line. Each field name is prefixed with the string <code>vf_</code> (Virtual File).
Example	Enter the command: <pre>vfddsp display_file -dp "/" -i 14</pre> Gateway returns the attributes for the logical file 14. This file is located in the VFD "/" directory: <pre>vf_local_ident=14 vf_dir_path=/ vf_date_create=20070122 154603 vf_file_size=0 vf_protocol=HTTP vf_access_right=----- vf_direction=0 vf_state=C vf_originator=GFL660 vf_destination=THHTTP vf_remote_agent=THHTTP vf_file_name=DEFAULT_B vf_xfer_ident=14 vf_route_state= vf_user_state= vf_permanent=N vf_appli=default_b vf_param1= vf_param2= vf_user_msg=Dummy Message vf_data_code=B vf_route_from_xfer=0 vf_route_to_xfer=0 vf_duplicate_from_xfer=0 vf_type=0</pre>

Exporting and Importing VFDs

This section describes Gateway commands you can use to move VFDs between Gateways.

vfdbase export

Syntax	vfdbase export {-output}
Description	When Gateway is running, use this command to export Virtual File Directories (VFDs) to an output file. By default, the resulting file is created in the <Gateway installation directory>/run_time/tmp directory. You can then edit the file and import the file to Gateway using the vfdbase import command. Note: During the export process, VFD items are lost.
Parameters	-output (-f) : Enter the name of the resulting file on your host machine.
Example	Enter the command: <pre>vfdbase export -f VFDMove</pre> Gateway creates an output file named <code>VFDMove</code> that contains descriptions of all VFD files on your Gateway: <pre>[Overall] Version = 110 [Directory] Type = VIRTUAL_DIR NodeIdent = 1000 ... UserData =</pre>

vfdbase import

Syntax	vfdbase import {-input} [-discard_mount_points] [-directories] [-transfers] [-ignore_warnings]
Description	Use this command to import the complete set of Virtual File Directories into the Gateway database. The VFDs are described in a file you created using the vfdbase export command.

Parameters	<code>-input (-if)</code> : Enter the name of the file containing VFDs for import. <code>-discard_mount_points (-dmp)</code> : Enter a value to indicate whether or not the mount points are to be discarded on import: • Y = Directories are virtual and do not direct to physical directories. In that case, you must mount them manually. • N = Links are kept between directories (default) <code>-directories (dir)</code> : Enter a value to indicate whether or not the directory is to be imported: • Y (default) • N <code>-transfers (-tr)</code> : Enter a value to indicate whether or not you want Gateway to retrieve and import transfer records from the Mailbox: • Y (default) • N <code>-ignore_warnings (-iw)</code> : Enter a value to indicate whether or not you want to ignore warnings generated during the import process: • Y • N (default)
Details for use	Before importing VFDs: 1. Shut down the target Gateway. 2. On the target Gateway machine, manually delete the data files in the VFD library (<code>p_runtime_dir/vfd</code>). If you do not do this, the import procedure will not succeed and the message "VFD files already exist, delete all files before importing" will be displayed. 3. Create a VFD output file on the originating Gateway, using the <code>vfdbase export</code> command.
Example	Enter the following command (assuming the file containing the importable VFDs is named <code>VFDMove</code>): <pre>vfdbase import -if VFDMove</pre>

[Overview: Virtual File Directories](#)

[Virtual File Directories: Parameters List](#)

Virtual File Directories: Parameters List

[VFD directory parameters](#)

[VFD file parameters](#)

VFD directory parameters

The following table lists all VFD directory parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Meaning	Values
alias (-a) <i>Alias</i>	VFD directory alias. Use this alias to designate a specific directory (whether moved or renamed) within the VFD administration and Transfer Request command lines.	Up to 32 alphanumeric characters Its value must be unique in the whole VFD
comments (-cts) <i>Comments</i>	Comments about a VFD directory	Up to 80 alphanumeric characters
dir_path (-dp) <i>PathName + Name</i>	Absolute path of the VFD directory. You can either reference the VFD via the path name or via the alias.	The total length of this parameter is limited to 255 characters (or slightly lower in some cases). Individual directory nodes in the path are limited to 127 characters.
filename_display_option (-fdo) <i>File name display option</i>	Indicates how file names are displayed	One of the following values: • Standard = file name only • XFB = <code>s.XferIdent.filename</code> (default) Refer to VFD file name display option .

Parameter	Meaning	Values
filename_ duplicate_policy (-fdp) File name duplication policy	Indicates how duplicate file names are managed	One of the following values: • duplicate (default) • forbid • override Refer to VFD file name duplication policy .
filter (-f) (None)	Listing directory contents: String which can contain "?" and "*" wildcard characters. Only displays the inner components of the current VFD directory whose name matches the filter string.	Up to 24 alphanumeric characters
is_cgi_bin (- icb) (None)	VFD CGI directory	Y/N
mbx_recording (- mr) Monitoring	<i>RFD mode only</i> Gateway Mailbox recording option. This option indicates whether to trace the transfer operations on the current directory (downloading and uploading) in the Gateway Mailbox.	Y/N
mount_dir_path (-mdp) Mount name	Absolute path of the physical directory (RFD) to be mounted	The total length of this parameter is limited to 255 characters. Individual directory nodes in the path are limited to 127 characters.
mount_point (- mp) (None)	Indicates whether or not to mount the current directory	Y/N

Parameter	Meaning	Values
new_alias (-na) (None)	Updating a VFD directory alias: Indicates new alias value	Up to 24 alphanumeric characters
new_dir_path (-ndp) (None)	Moving or renaming a directory: VFD absolute path of the new location of the VFD directory	The total length of this parameter is limited to 255 characters. Individual directory nodes in the path are limited to 127 characters.
no_dir (-nd) (None)	Listing directory contents: Indicates that sub-directories must not be displayed	–
no_file (-nf) (None)	Listing directory contents: Indicates that files must not be displayed	–
tls_only (-tls) (None)	TLS only flag	Y/N
user_attribute (-ua) User data	User parameter associated with the current directory	Up to 80 alphanumeric characters

VFD file parameters

All of the following parameters are extracted from the related Gateway Mailbox record. The following table lists all VFD file parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Meaning	Values
access_rights (None)	Access rights associated with the current VFD Item	
appli Application	Application name of the current VFD Item	Up to 27 alphanumeric characters

Parameter	Meaning	Values
data_code Data code	Indicates the data code of transferred data	One of the following values: • B= Binary • A= ASCII • A1= ASCII1 • A2= ASCII2 • A3= ASCII3 • A4= ASCII4 • A5= ASCII5 • E= EBCDIC • E1= EBCDIC1 • E2= EBCDIC2 • E3= EBCDIC3 • E4= EBCDIC4 • E5= EBCDIC5 • ISO= ISO • OEMPC= OEMPC • U= Undefined
date_create Created on	VFD file creation date	Format YYYYMMDD HHMMSS
destination Destination	Destination protocol identifier of the current VFD file	Up to 25 alphanumeric characters
direction Direction	Direction of the Transfer related to the current VFD file	One of the following values: • I = Input • O = Output
dir_path Dir path	VFD absolute path of the current VFD Item parent directory	Up to 255 alphanumeric characters

Parameter	Meaning	Values
duplicate_from_xfer Duplicated from	Identifier of permanent Transfer Request used to generate the Transfer that corresponds to the current VFD file	Integer (automatically assigned by Gateway)
file_name Filename	Protocol file name of the current VFD file	Up to 76 alphanumeric characters
file_size File size	Size in bytes of the current VFD file	Integer
local_ident Local ident	Local transfer identifier of the current VFD file	Integer (automatically assigned by Gateway)
originator Originator	Originator protocol identifier of the current VFD file	Up to 25 alphanumeric characters
param1 Param 1	User parameter 1	Up to 40 alphanumeric characters
param2 Param 2	User parameter 2	Up to 40 alphanumeric characters
permanent Permanent	Transfer Request permanent flag indicating whether the current VFD file is permanent or not	Y/N
protocol Protocol	Protocol (restricted to FTP, SFTP and HTTP)	One of the following values: • HTTP = HyperText Transfer Protocol • FTP = File Transfer Protocol • SFTP = SSH File Transfer Protocol
remote_agent Remote agent alias	Remote Site alias associated with the current VFD file	Up to 25 alphanumeric characters

Parameter	Meaning	Values
<code>route_from_xfer</code> Route from	Request Local identifier used to route the Transfer related to the current VFD file	Automatically assigned by Gateway
<code>route_state</code> Route state	Store-and-forward state of the current VFD file	Automatically assigned by Gateway. One of the following values: • undefined • TO_ROUTE • ROUTED • FAILED • ACKNOWLEDGED
<code>route_to_xfer</code> Route to	Routed Transfer request identifier related to the current VFD file	Automatically assigned by Gateway

Parameter	Meaning	Values
state State	State of the Transfer Request associated with the current VFD file	One of the following values: • F = Frozen • D = Delayed • B = To_Begin • P = Processing • R = To_Restart • S = Suspended • E = Ended • C = Canceled • I = Interrupted • V = serVicing • A = Acked (OFTP, PeSIT, FTP) • T = CreaTed • Z = Deleted • W = Ended_to_ack • G = To_SiGn • U = Nacked_User • X = Nacked_Xfer
type Type	Transfer type of the current VFD file	One of the following values: • TRANS • LIST • POLL • LOTS • EERP • SELECT • MSG
user_msg User message	User message associated with the current VFD file	Up to 254 alphanumeric characters

Parameter	Meaning	Values
<code>user_state</code> User state	User state that the user application assigns to the current VFD file	Single character - value set in user exits
<code>vfd_nb_directory_max <new value></code>	The maximum number of virtual file directories. Note: You must restart Gateway for the new value to be taken into account.	Integer 1024 - 100 000 (default 8192)

[Overview: Virtual File Directories](#)

[Working with Virtual File Directories \(command line\)](#)

Axway Gateway: Managing Transfers

Parameters (applications and models)

You can use the Application object and/or Model object to provide optional additional configuration parameters for Gateway Transfer Requests.

Applications

Use the [Application object](#) to define file system attributes for a Transfer Request.

This object enables you to specify local attributes of the file system as well as transfer attributes for files sent over the network. Additionally, an Application object can contain parameters that define how an Application sends or receives files.

Models

Use the [Model object](#) to create reusable templates for the creation of:

- Transfer Requests
- Decision Rule objects
- Axway Messaging (XMS) connector objects

Axway Gateway: Managing Transfers

Parameters (applications and models)

You can use the Application object and/or Model object to provide optional additional configuration parameters for Gateway Transfer Requests.

Applications

Use the [Application object](#) to define file system attributes for a Transfer Request.

This object enables you to specify local attributes of the file system as well as transfer attributes for files sent over the network. Additionally, an Application object can contain parameters that define how an Application sends or receives files.

Models

Use the [Model object](#) to create reusable templates for the creation of:

- Transfer Requests
- Decision Rule objects
- Axway Messaging (XMS) connector objects

Axway Gateway: Managing Transfers

About applications

[Introduction to Applications](#)

[How Gateway assigns an Application](#)

Applications

When you define a File Transfer Request you can provide certain file system attributes via the *Application object*. Some of the attributes are used by the local file system (local attributes). Others are used to manage file data content that is transmitted over the network (transfer attributes). Additionally, an Application object contains parameters that define how this Application can send or receive files. For example:

- Local file system attributes such as name, file structure and type
- Record format
- Data encryption and padding
- Online record attributes
- Compression mode

The file transfer process may occur between different platforms, for example, Windows workstations, MVS Mainframes, UNIX servers and so on. Since the data storage format on these machines may be completely different, Gateway provides automatic character set conversion via the Application or the Model object. So, Gateway can transparently convert a file sent from an EBCDIC system (that is, MVS) whose final destination is an ASCII system, and vice versa.

It is recommended that you use an Application with certain protocols (for example, PEL). With other protocols, Applications are used for a specific purpose (for example, a link between the Application name and the protocol file name in PeSIT or FTP).

You can use the GUI or online commands to manage Applications.

How Gateway assigns an application

Before data transfer, Gateway assigns the appropriate Application to the Transfer, depending on the protocol.

PeSIT Transfers

The Application name is the incoming protocol file name. Depending on the type of data to be transferred, the default Application names are:

- DEFAULT_B (Binary)
- DEFAULT_A (ASCII)
- DEFAULT_E (EBCDIC)

If Gateway cannot locate the required Application (from those listed above), it uses the Application named DEFAULT.

FTP/SFTP Transfers

From the client station, you can define Application names with an FTP `SITE` command before using the `STOR` or the `RETR` command. Depending on the FTP data type, the default Application names are:

- FTP_A (ASCII)
- FTP_B (binary)
- FTP_E (EBCDIC)

Since ASCII and EBCDIC are meaningless for SFTP, Gateway always assigns the default Application FTP_B to SFTP Transfers.

HTTP Transfers

Since the client station does not provide the Application name, Gateway uses the default FTP Applications for file reception.

Sites and Applications always belong to a group. For a Transfer to be valid, its Remote Site and Application must belong to the same group. However, applications can be configured over http by setting the parameter `use_applications` to 1 in `<installation_folder>/run_time/etc/pelsetup.def` file, section `FT_HTTP` (configuration needs to be rebuilt)

Line endings conversion

Gateway offers the possibility to convert line-endings when receiving a file.

This functionality is configurable, activated by the `[ft_http]convert_ascii_line_endings` parameter (default is set to 0 - Disabled).

When this parameter is set to 1, the line endings are converted according to the target machine.

File organizations on Mainframe

The following notes on file organization relate to sending and receiving files to and from a mainframe.

Sequential

In a sequential file organization, records are organized in the sequence by which they were added. New records cannot be inserted between existing records, but only at the end of the last record. It is a simple file organization that allows processing of batches of records in the file without altering its structure. However, to access a particular record, processing must run through all the other records before it as there is no method to point to the location of a certain record. Searching for a record, especially when there are thousands of entries, may be time consuming. Note that after a record is written in the file it cannot be shortened, lengthened or deleted, it can only be rewritten if the length doesn't change.

Relative

Another approach of organizing records in a file would be to generate a key pointing to the relative position in the file where the record begins. The relative key is assigned to determine the order of records in the file. The first record would have a relative number of 1, the second record would have a relative number of 2 and so on. Records may have various sizes, in bytes, so they can be arranged relative to each other, with the restriction that care must be taken to always insert a new record with the next relative key in the sequence (Assuming that the last record has the relative number 7, the new record that would be inserted must and would have the relative number 8). Also, with the relative key, you can randomly access any record without starting from the top record. The disadvantage is its dependence on relative keys. If you do not know the relative key of a particular record, you won't be able to randomly access the file.

Indexed

An indexed file organization contains record reference numbers, like unique identifiers for each record in relation to other records. These references are similar to the primary keys in a RDBMS. Alternate keys can also be defined to allow alternate methods of accessing the record. For example, instead of accessing an employee's record using employee numbers, you can use an alternate key that reference

employees by departments. This allows greater flexibility for users to randomly search through thousands of records in a file. However, additional meta information has to be maintained for each of these files increasing their size on disk and requiring periodic maintenance to avoid wasted space and avoid performance degradation.

[Working with Applications](#)

[Working with Applications \(command line\)](#)

[Application objects: Parameters List](#)

[Overview: Applications](#)

Axway Gateway: Managing Transfers

Working with applications

[Creating a new Application object](#)

[Displaying a list of Application objects](#)

[Displaying a list of selected Application objects](#)

[Displaying the attributes of a specified Application object](#)

[Modifying an Application object](#)

[Deleting an Application object](#)

Creating a new Application object

To create a new Application object:

1. In the left pane of the GUI main window, click  to expand the nodes:
Transfer Management > Parameters
2. Right-click the **Application** sub-node, then from the context menu select **New**.

Gateway displays the *New Application* window with the following tabs:

- [General](#)
 - [Physical file attributes](#)
 - [Transfer attributes](#)
 - [Options](#)
 - [PeSIT E / OFTP](#)
 - [SMTP/POP3](#)
3. Complete the tabs according to your transfer requirements. The following tables describe the parameters and possible values you can attribute in each tab.

New Application: General tab

Field	Description
Name	Enter the Application name. Maximum: • 8 alphanumeric characters for PEL • 27 alphanumeric characters for all other protocols
Group	Enter the Group name. Maximum: 15 alphanumeric characters. Default = GDEFAULT .
Model	Enter the Model name used to create the Transfer Request. Maximum: 25 alphanumeric characters. Alternatively, select a Model from the drop-down list.
Comments	Enter a free-text description of the Application. Maximum: 80 alphanumeric characters.
Direction	Select one or both of the following Transfer directions: • Send • Receive

New Application: Physical file attributes tab

Use the fields on the **Physical file attributes** tab to define the attributes of files stored locally.

Field	Description
Reception	
Directory for received files	Enter the name of the storage directory for received files. Maximum: 127 alphanumeric characters.
Received file name	Enter the name for the received file. Maximum: 127 alphanumeric characters. You can use symbolic variables in this field. For example, you could specify the received file with the following path and name: <pre> C:\<Gateway installation directory> \run_time\tmp\&(x_file_name) \D&(x_destination)L&(x_file_label)I&(x_local_ ident).txt </pre>

Field	Description
Encrypt received files	Select this option to encrypt received files.
Local file attributes	
Organization	Select the type of local file organization: • Sequential • Indexed • Relative • Undefined
Record format	Select the record format for the local file: • Variable Text • Variable • Fixed Text • Fixed • Stream • Stream Text • Undefined
Record length	Specify the record length for the local file. Integer: 0 - 32 768. Default = 1024. The value that you enter in this field must be less than or equal to the Maximum data block size parameter (<code>-data_size_max</code>) defined in the corresponding Remote Site.
Data code	Select the data type for the local file.
Padding character (Hexa)	Specify the padding character. Hexadecimal number: 00 - FF. Default = 00.
Check records	This option determines whether or not records are checked in RFD (Real File Directory) mode. Deselect this option to skip the record checking phase. This is useful if the check takes too long (for example, with large ASCII files) and there is the risk of a time-out on the client side.

New Application: Transfer attributes tab

Use the fields on the **Transfer attributes** tab to define the attributes of files that you transfer over the network.

Field	Description
Transfer attributes	
Organization	Select the type of file organization: • Sequential • Indexed • Relative • Undefined
Record format	Select the record format for the transferred file: • Variable • Fixed • Stream • Text (<i>OFTP only</i>) • Undefined
Record length	Specify the record length for the transferred file. Integer: 0 - 32768 . Default = 1024 .
Data code	Select the data type for the Transfer.
Compression	Select the compression option: • No • Horizontal • Vertical • Both • Zlib
Responder mode	
Local file name	<i>Only complete this field if you are in Responder mode.</i> Specify the protocol name of the file sent or received. Maximum: 127 alphanumeric characters.

Field	Description
Local directory (VFD)	Only complete this field if: - the connection mode is Responder and the direction is Send - the file transfer protocol is <i>FTP, SFTP or HTTP</i> Specify the VFD absolute path of the local directory that stores the file to send. Maximum: 127 alphanumeric characters. Alternatively, click the Browse button to select a directory from the <i>VFD browse</i> window.
Local directory alias	Only complete this field if: - you did not complete the Local directory field - the connection mode is Responder and the direction is Send - the file transfer protocol is <i>FTP, SFTP or HTTP</i> Specify the VFD alias of the local directory that stores the file to send. Maximum: 32 alphanumeric characters.
Initiator mode	
Remote file name	Only complete this field if you are in Initiator mode. Enter the protocol name of the file sent to, or received from, the Remote Site. Maximum: 127 alphanumeric characters.
Remote directory	Enter the absolute path of the remote directory for the file sent or received. Maximum: 127 alphanumeric characters.

New Application: Options tab

Field	Description
Physical file attributes	
System file type	Specify the file type depending on the system: BULL, AS400, GCOS and TANDEM machines.
Physical block size	Specify the block size in bytes. Default = 1024 .
User parameters	

Field	Description
Param 1	Enter user parameter 1. Maximum: 40 alphanumeric characters.
Param 2	Enter user parameter 2. Maximum: 40 alphanumeric characters.
Sentinel	
Sentinel Transfer Filter	Use this option to send a record of Transfer state changes to Sentinel. Select one of the following: • All: sends all Transfer state changes to Sentinel • Summary: sends a summary of Transfer state changes (canceled, ended, acked, created, to begin) • None: sends no Transfer state changes • Undefined: Gateway searches for the definition first in the Remote Site, and then in the configuration file

New Application: PeSIT E / OFTP tab

Field	Description
Sender	
User ID	Specify the sender User name. Maximum: 8 alphanumeric characters.
Application ID	Specify the sender Application name. Maximum: 8 alphanumeric characters.
Message text	Enter a free-text description for the sender. Maximum: 8 alphanumeric characters.
Receiver	
User ID	Specify the receiver User name. Maximum: 8 alphanumeric characters.
Application ID	Specify the receiver Application name. Maximum: 8 alphanumeric characters.
Message text	Enter a free-text description for the receiver. Maximum: 8 alphanumeric characters.

Field	Description
Receiving a pre-existing file Presence-check indicator of the transferred file. The values in this section define the action of the monitor if the file being transferred already exists.	
File availability	Select one of the following values: • Undefined • New • Old • Both Gateway responds to the various possible pre-existing file conditions depending on the values of the File availability and File action parameters. Refer to the description of the File action parameter below.

Field	Description
File action	Select one of the following values: • Undefined • Erase • Delete • Verify To determine how to handle a pre-existing file, Gateway interprets the File availability and File action parameter values as follows: If File availability is set to: • NEW and a valid file with the same file name already exists, then: ◦ ERASE = refuse ◦ DELETE = refuse ◦ VERIFY = refuse • NEW and a file with the same file name does not already exist, then: ◦ ERASE = accept ◦ DELETE = accept ◦ VERIFY = accept • OLD and a valid file with the same file name already exists, then: ◦ ERASE = accept ◦ DELETE = accept ◦ VERIFY = accept if empty, refuse if not empty • OLD and a file with the same file name does not already exist, then: ◦ ERASE = refuse ◦ DELETE = refuse ◦ VERIFY = refuse • BOTH and a valid file with the same file name already exists, then: ◦ ERASE = accept ◦ DELETE = accept ◦ VERIFY = accept if empty, refuse if not empty • BOTH and a file with the same file name does not already exist, then:

Field	Description
	◦ ERASE = accept ◦ DELETE = accept ◦ VERIFY = accept where: accept = accept the transferred file refuse = refuse the transferred file

New Application: SMTP/POP3 tab

Field	Description
Attach/extract file	Use this field to indicate whether Gateway must create an email and attach the file or not. Gateway checks the Application object first for the Extract file parameter, and then the Remote Site. Note: If this value is defined in the Transfer Request, the Transfer Request definition overrides the value that you enter here. Select one of the following options: • Yes indicates that the file is included on transmission as an attachment and extracted on reception. • No indicates a file to transfer that is already formatted as an email. (The MIME header and body are complete and well-formed.) • Undefined - Gateway retrieves the value of the Extract file field defined in the Remote Site POP3 tab.
Add Gateway specific parameters to subject	Use this option to add the following specific Gateway parameters to the email subject: • destination (-dest) • origin (-org) • protocol file name (-file_name) • application (-appli)

Displaying all Application objects

To display a list of all Application objects stored in the database:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters

2. Click the **Application** sub-node.

Gateway displays a list of all existing Application objects in the right (display) pane.

Displaying a list of selected Application objects

You can reduce the number of Application objects listed in the display pane by applying selection criteria to the display.

To display a selected list of Application objects:

1. In the left pane of the GUI main window, click  to expand the nodes:
Transfer Management > Parameters
2. Right-click the **Application** sub-node, then choose **Select...** from the context menu.
Gateway displays the *Select Applications* window.
3. In the *Select Applications* window, use the data fields to enter criteria for Application display selection. You can use one or more of the following selection criteria:
 - Application name
 - Site
 - Group
4. Click **OK**.

Gateway regenerates the right window Application display, taking into account your selection criteria.

Viewing the attributes of an Application object

To view the attributes of a specific Application object:

1. In the left pane of the GUI main window, click  to expand the nodes:
Transfer Management > Parameters
2. Click the **Application** sub-node.
Gateway displays a list of all existing Application objects in the right (display) pane.
3. In the right pane, right-click the name of the Application object you want to view. Select **View...** from the context menu.
Gateway displays the *Application (read only)* window. This window provides you with a view of all attributes for the selected Application.
4. When you have finished viewing the information, click **Close** to quit the window.

Modifying an Application object

To modify the attributes of an existing Application object:

1. In the left pane of the GUI main window, click  to expand the nodes:
Transfer Management > Parameters
2. Click the **Application** sub-node.
Gateway displays a list of all existing Application objects in the right (display) pane.
3. In the display frame (right frame), right-click the Application object that you want to modify.
Then in the context menu click **Modify...**
Gateway opens the *Application* editing window.
4. Modify the value fields of the tabs according to your requirements.
5. Click **OK** to confirm changes and complete the modification procedure.

Deleting an Application object

To delete an existing Application object:

1. In the left pane of the GUI main window, click  to expand the nodes:
Transfer Management > Parameters
2. Click the **Application** sub-node.
Gateway displays a list of all existing Application objects in the right (display) pane.
3. In the right pane, right-click the name of the Application object you want to delete. Select **Delete...** from the context menu.
Gateway displays a confirmation dialog box.
4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
Alternatively, click **No** or **Cancel** to abandon the operation.
Gateway deletes the Application object from the database and removes the entry from the display window.

[About Applications](#)

[Working with Applications \(command line\)](#)

[Application Objects: Parameters List](#)

Axway Gateway: Managing Transfers

Working with Applications (command line)

[peladm create appli](#)

[peladm update appli](#)

[peladm delete appli](#)

[peldsp select appli](#)

[peldsp display appli](#)

Creating an Application: *peladm create_appli*

Syntax	<code>peladm create_appli {-name} [optional parameters]</code>
Description	Use this command to create an Application. An Application defines a type of file structure. Application attributes must be compatible with the local file system features. To create an Application, set up all necessary parameters, including platform-specific parameters if they are required. Some parameters have default values supplied either by the administrative interface, or internally by Gateway. For example, if you do not specify the group name, the Application takes the default group name: GDEFAULT.
Parameters	<code>-name (-n)</code> : Enter the name of the application you are creating. <i>Optional parameters:</i> Refer to Application objects: Parameters List.
About the examples	The following examples do not supply platform-specific parameters. You must adapt these specific parameters to your own configuration. Note: You can create two Applications with the same name if they each belong to a different group.
Example 1	The following command creates an Application object named APPLI that belongs to the group ALPHA. This Application object can be used to transfer files with horizontal compression, variable record format and record length of 160. The record format of the transferred file is fixed and the record length is 1024. <pre>peladm create_appli -name APPLI -gr ALPHA -comp H -rec_fmt V -rec_len 160 -text_file Y -xrf F -xrl 1024</pre>

Example 2

The following command creates an Application used to transfer ASCII files of 255 byte fixed record length, using the default group name:

```
peladm create_appli -n fix255
-cts "fixed record length: 255"
-f_org Sequential
-rec_fmt Fixed
-rec_len 255
-dc ASCII
-xrf F
-xrl 255
-xdc ASCII
-dp <your Received files directory>
```

Example 3

The following command creates an Application used to transfer ASCII files of fixed 80 byte record length, ending with '\n' or '\r' depending on the system, and that uses the default group name. The end-of-line characters are deleted from the file content and the file content is converted into EBCDIC before transmission:

```
peladm create_appli -n fix80
-cts "fixed record length: 80"
-f_org Sequential
-rec_fmt Fixed text
-rec_len 80
-dc ASCII
-xrf F
-xrl 80
-xdc EBCDIC
-dp <your Received files directory>
```

Example 4 The following example creates an Application used to transfer EBCDIC files in stream mode record format, record length of 4000 bytes (except for the last record that can be shorter) and with the group name: GRAPPLIS:

```
peladm create_appli& -n str4000
-g GRAPPLIS
-cts "Stream: 4000"
-f_org Sequential
-rec_fmt Stream
-rec_len 4000
-dc EBCDIC
-xrf V
-xrl 4000
-xdc EBCDIC
-dp <your Received files directory>
```

Modifying an Application: peladm update_appli

Syntax	<code>peladm update_appli {-name} [optional parameters]</code>
Description	Use this command to modify an existing Application object. Enter the Application name and the group name (optional parameter; default = GDEFAULT) as parameters, followed by all the fields you want to modify.
Parameters	<code>-name (-n)</code> : Enter the name of the application you are modifying. <code>-group (-gr)</code> : Enter the group name of the application. <i>Optional parameters:</i> Refer to Application objects: Parameters List.
Example	The following command modifies the group and the compression value for the Application APPLI: <pre>peladm update_appli -name APPLI -gr ALPHA -comp B</pre>

Deleting an Application: peladm delete_appli

Syntax	<code>peladm delete_appli {-name} [optional parameters]</code>
Description	Use this command to delete an Application. Since Applications in different Groups can have the same name, you must specify the Application name <i>and</i> the Group name.
Parameters	<code>-name (-n)</code> : Enter the name of the Application object you are deleting. <code>-group (-gr)</code> : Enter the group name of the Application object you are deleting.
Example	The following command deletes Application object APPLI in the group ALPHA. <pre>peladm delete_appli -name APPLI -gr ALPHA</pre>

Selecting Applications: peldsp select_appli

Syntax	<code>peldsp select_appli {-name} [-group] [-site]</code>
Description	Use this command to select the Applications that match the criteria entered as parameters. To list all Applications, enter no parameters. Use the optional parameters to refine your search.
Parameters	<code>-name (-n)</code> : Enter the name of the Application object you are deleting. <code>-group (-gr)</code> : Enter the group name of the Application object you are deleting. <code>-site (si)</code> : Enter the name of a Remote Site associated with the Application.
Example	Gateway returns all Application names that match the selection criteria. Each string parameter is seen as a truncated search key. The following command returns a list of the Applications whose names begin with B. <pre>peldsp select_appli -n B I_DEFAULT GDEFAULT I_FNAM_F3N GDEFAULT</pre>

Displaying an Application: peldsp display_appli

Syntax	<code>peldsp display_appli {-name} [-group] [-site]</code>
Description	Use this command to display the parameters of an Application object that you identify by its name. The Application parameters are displayed on the screen. If the Application group is not GDEFAULT, the group parameter is mandatory.
Parameters	<code>-name (-n)</code> : Enter the name of the Application object for which you want to display the attributes. <code>-group (-gr)</code> : Enter the group name of the Application object for which you want to display the attributes. <code>-site (-si)</code> : Enter the name of a Remote Site associated with the Application for which you want to display the attributes.
Results of the command	Gateway returns a display in which each parameter is displayed on a separate line, in the format: <pre>name_of_field="value"</pre> Gateway prefixes each field name with the string "a_", referring to the Application object.

[Application objects: Parameters List](#)

Axway Gateway: Managing Transfers

Working with Applications (command line)

[`peladm create appli`](#)

[`peladm update appli`](#)

[`peladm delete appli`](#)

[`peldsp select appli`](#)

[`peldsp display appli`](#)

Creating an Application: peladm create_appli

Syntax	<code>peladm create_appli {-name} [optional parameters]</code>

Description	Use this command to create an Application. An Application defines a type of file structure. Application attributes must be compatible with the local file system features. To create an Application, set up all necessary parameters, including platform-specific parameters if they are required. Some parameters have default values supplied either by the administrative interface, or internally by Gateway. For example, if you do not specify the group name, the Application takes the default group name: GDEFAULT.

Parameters	<code>-name (-n)</code> : Enter the name of the application you are creating. <i>Optional parameters:</i> Refer to Application objects: Parameters List.

About the examples	The following examples do not supply platform-specific parameters. You must adapt these specific parameters to your own configuration. Note: You can create two Applications with the same name if they each belong to a different group.

Example 1	The following command creates an Application object named APPLI that belongs to the group ALPHA. This Application object can be used to transfer files with horizontal compression, variable record format and record length of 160. The record format of the transferred file is fixed and the record length is 1024. <pre>peladm create_appli -name APPLI -gr ALPHA -comp H -rec_fmt V -rec_len 160 -text_file Y -xrf F -xrl 1024</pre>

Example 2

The following command creates an Application used to transfer ASCII files of 255 byte fixed record length, using the default group name:

```
peladm create_appli -n fix255
-cts "fixed record length: 255"
-f_org Sequential
-rec_fmt Fixed
-rec_len 255
-dc ASCII
-xrf F
-xrl 255
-xdc ASCII
-dp <your Received files directory>
```

Example 3

The following command creates an Application used to transfer ASCII files of fixed 80 byte record length, ending with '\n' or '\r' depending on the system, and that uses the default group name. The end-of-line characters are deleted from the file content and the file content is converted into EBCDIC before transmission:

```
peladm create_appli -n fix80
-cts "fixed record length: 80"
-f_org Sequential
-rec_fmt Fixed text
-rec_len 80
-dc ASCII
-xrf F
-xrl 80
-xdc EBCDIC
-dp <your Received files directory>
```

Example 4 The following example creates an Application used to transfer EBCDIC files in stream mode record format, record length of 4000 bytes (except for the last record that can be shorter) and with the group name: GRAPPLIS:

```
peladm create_appli&  -n str4000
-g GRAPPLIS
-cts "Stream: 4000"
-f_org Sequential
-rec_fmt Stream
-rec_len 4000
-dc EBCDIC
-xrf V
-xrl 4000
-xdc EBCDIC
-dp <your Received files directory>
```

Modifying an Application: peladm update_appli

Syntax	peladm update_appli {-name} [optional parameters]
Description	Use this command to modify an existing Application object. Enter the Application name and the group name (optional parameter; default = GDEFAULT) as parameters, followed by all the fields you want to modify.
Parameters	-name (-n) : Enter the name of the application you are modifying. -group (-gr) : Enter the group name of the application. Optional parameters: Refer to Application objects: Parameters List.
Example	The following command modifies the group and the compression value for the Application APPLI: <pre>peladm update_appli -name APPLI -gr ALPHA -comp B</pre>

Deleting an Application: peladm delete_appli

Syntax	<code>peladm delete_appli {-name} [optional parameters]</code>
Description	Use this command to delete an Application. Since Applications in different Groups can have the same name, you must specify the Application name <i>and</i> the Group name.
Parameters	<code>-name (-n)</code> : Enter the name of the Application object you are deleting. <code>-group (-gr)</code> : Enter the group name of the Application object you are deleting.
Example	The following command deletes Application object APPLI in the group ALPHA. <pre>peladm delete_appli -name APPLI -gr ALPHA</pre>

Selecting Applications: peldsp select_appli

Syntax	<code>peldsp select_appli {-name} [-group] [-site]</code>
Description	Use this command to select the Applications that match the criteria entered as parameters. To list all Applications, enter no parameters. Use the optional parameters to refine your search.
Parameters	<code>-name (-n)</code> : Enter the name of the Application object you are deleting. <code>-group (-gr)</code> : Enter the group name of the Application object you are deleting. <code>-site (si)</code> : Enter the name of a Remote Site associated with the Application.
Example	Gateway returns all Application names that match the selection criteria. Each string parameter is seen as a truncated search key. The following command returns a list of the Applications whose names begin with B. <pre>peldsp select_appli -n B I_DEFAULT GDEFAULT I_FNAM_F3N GDEFAULT</pre>

Displaying an Application: peldsp display_appli

Syntax	<code>peldsp display_appli {-name} [-group] [-site]</code>
Description	Use this command to display the parameters of an Application object that you identify by its name. The Application parameters are displayed on the screen. If the Application group is not GDEFAULT, the group parameter is mandatory.
Parameters	<code>-name (-n)</code> : Enter the name of the Application object for which you want to display the attributes. <code>-group (-gr)</code> : Enter the group name of the Application object for which you want to display the attributes. <code>-site (-si)</code> : Enter the name of a Remote Site associated with the Application for which you want to display the attributes.
Results of the command	Gateway returns a display in which each parameter is displayed on a separate line, in the format: <pre>name_of_field="value"</pre> Gateway prefixes each field name with the string "a_", referring to the Application object.

[Application objects: Parameters List](#)

Axway Gateway: Managing Transfers

Application objects: parameters list

The following table lists all Application command parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Meaning	Values
<code>alloc_unit (-au)</code> <i>(none)</i>	Allocation unit	One of the following values: R = expressed in Records K = expressed in Kilobytes

Parameter	Meaning	Values
attach_extract_file (-aef) Attach/extract file	<i>SMTP/POP3 only</i> Specify whether or not to attach a file	One of: • Y (if the file is not already formatted as an email) • N (if the file is already formatted as an email) • U (undefined)
block_size (-bs) Physical block size	Block size in bytes	Integer (default: 0)
check_records (-c_rec) Check records	This parameter determines whether or not records are checked in RFD (Real File Directory) mode.	One of: • Y = Check records in RFD mode (default) • N = Skip the record checking phase in RFD. This is useful if the check takes too long (for example, with large ASCII files) and there is the risk of a time-out on the client side.
comments (-cts) Comments	User comments	Up to 80 alphanumeric characters
compression (-comp) Compression	Compression	One of: • U = Undefined (default) • H = Horizontal • V = Vertical • B = Both • Z = Zlib

Parameter	Meaning	Values
data_code (-dc) Data code	Data code (local file attributes)	One of: • B = Binary • A = ASCII • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E = EBCDIC • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • ISO = ISO • OEMPC = OEMPC • U = Undefined (default)
dir_path (-dp) Directory for received files	Storage directory for received files. You can define the local storage directory using Transfer variables.	Up to 127 alphanumeric characters
direction_allowed (-dir_all) Direction	Direction allowed	One of: • O = Output • I = Input • B = Both (default)
etb5_auth_type (-5at) Authentication	Authentication type	One of: • UNUSED (default) • SIMPLE • RECIPR = reciprocal

Parameter	Meaning	Values
etb5_ciph_type (-5ct) Encryption	Encryption type to ensure data confidentiality	One of: • UNUSED (default) • DES_CBC_ART = DES simple article • DES_CBC_FIC = DES simple file • DES_TRI_ART = DES triple article • DES_TRI_FIC = DES triple file
etb5_id_bank (-5ba) Bank identifier	The bank identifier (PI 62)	Up to 24 alphanumeric characters
etb5_id_bank_sec (-5bs) Secondary bank ID	<i>Optional</i> Enter a secondary bank identifier (PI 4 in FPDU.CREATE or FPDU.ACK.SELECT)	Up to 24 alphanumeric characters
etb5_id_client (-5cl) Client identifier	The client identifier (PI 61)	up to 24 alphanumeric characters
etb5_id_client_sec (-5cls) Secondary client ID	<i>Optional</i> Enter a secondary client identifier (PI 3 in FPDU.CREATE or FPDU.ACK.SELECT)	Up to 24 alphanumeric characters

Parameter	Meaning	Values
<code>etb5_seal_type</code> (- 5st) <i>Sealing</i>	To ensure data integrity, enter a sealing option	One of: • UNUSED (default) • DES_PAR_ART = DES partial article • DES_FIN_ART = DES final article • DES_FIN_FIC = DES final file • MDC_FIN_FIC = MDC final file • SHA_FIN_FIC = SHA final file

Parameter	Meaning	Values
file_action (- faction) File action	<i>PeSIT protocol only</i> Gateway action on a pre-existing file before receiving the transfer of a file with the same name. Gateway responds to the various possible situations according to the value of the file_dispo parameter. If this parameter value is: - NEW and a valid file with the same file name already exists, then: ERASE=NOK / DELETE=NOK / VERIFY=NOK - NEW and a file with the same file name does not already exist, then: ERASE=OK / DELETE=OK / VERIFY=OK - OLD and a valid file with the same file name already exists, then: ERASE=OK / DELETE=OK / VERIFY=OK if empty or NOK if not empty - OLD and a file with the same file name does not already exist, then: ERASE=NOK / DELETE=NOK / VERIFY=NOK - BOTH and a valid file with the same file	One of: undefined = not applicable (default) erase = erases (resets) the existing file delete = deletes the existing file verify = checks that the exiting file is empty

Parameter	Meaning	Values
	name already exists, then: ERASE=OK / DELETE=OK / VERIFY=OK if empty or NOK if not empty BOTH and a file with the same file name does not already exist, then: ERASE=OK / DELETE=OK / VERIFY=OK where: OK = transfer accepted NOK = transfer refused	
file_component (- fc) (none)	Local file name You can define the local file name using Transfer variables. For example: <i>F&(x.local_ident)</i>.	Up to 127 alphanumeric characters
file_dispo (- fdisp) File availability	<i>PeSIT protocol only</i> Presence-check indicator of the receiver file used to determine the action of the monitor if the file being transferred already exists. This parameter operates together with the file action parameter.	One of: undefined = not applicable (default) new = the file must be created by the monitor. The transfer is refused if the file already exists. old = the file must already exist both = if the file does not exist, it will be created

Parameter	Meaning	Values
<code>file_org (-f_org)</code> Organization	Type of file organization	One of: • S = Sequential file (default) • I = Indexed file • R = Relative file
<code>file_type (-ft)</code> System file type	Specify the file type depending on the system: BULL, AS400, GCOS and TANDEM machines.	
<code>generate_xfb_subject (-gxs)</code> Add Gateway parameters to subject	<i>SMTP/POP3 only</i> Use this field to specify whether or not to add specific Gateway parameters to the email <i>Subject</i> field	One of: • Y • N • U (default)
<code>group (-gr)</code> Group	Group name	Up to 15 alphanumeric characters (default: GDEFAULT)
<code>local_dir_name (-ldn)</code> Local directory	<i>FTP, HTTP & SFTP only</i> VFD absolute path of the local directory that stores the file to be sent. Useful in Responder mode and direction OUT only.	Up to 127 alphanumeric characters
<code>local_dir_name_alias (-ldna)</code> Local directory alias	<i>FTP, HTTP & SFTP only</i> VFD alias of the local directory that stores the file to be sent. Useful in Responder mode and direction OUT only.	Up to 32 alphanumeric characters
<code>local_file_name (-lfn)</code> Local filename	Protocol name of the file sent or received. Useful in Responder mode only.	Up to 127 alphanumeric characters

Parameter	Meaning	Values
<code>model (-m1)</code> Model	<i>This parameter only exists for reasons of compatibility with earlier product versions.</i> Model name used to create a Transfer Request One specific use case is to route a transfer using a <i>Model inside an Application</i> . This option is only available via online commands. For more information, refer to Working with Decision Rules (command line) .	Up to 25 alphanumeric characters
<code>name (-n)</code> Name	Application name	• PEL: Up to 8 alphanumeric characters • Other protocols: Up to 16 alphanumeric characters
<code>param1 (-p1)</code> Param 1	User parameter 1	Up to 40 alphanumeric characters
<code>param2 (-p2)</code> Param 2	User parameter 2	Up to 40 alphanumeric characters
<code>received_file_ciphared (-rfc)</code> Encrypt received files	Received file encryption option	Y/N
<code>rcv_appli (-ra)</code> Application ID	<i>PeSIT HS E Store-and-Forward use only</i> Receiver Application name	Up to 8 alphanumeric characters
<code>rcv_text (-rt)</code> Message text	<i>PeSIT HS E Store-and-Forward use only</i> Receiver free text	Up to 8 alphanumeric characters

Parameter	Meaning	Values
<code>rcv_user (-ru)</code> User ID	<i>PeSIT HS E Store-and-Forward use only</i> Receiver user name	Up to 8 alphanumeric characters
<code>rec_fmt (-rf)</code> Record format	Record format	One of: • F = Fixed records • FT = Fixed Text records • V = Variable records • VT = Variable Text records • S = Stream
<code>rec_len (-rl)</code> Record length	Record length The record length value must be less than or equal to the <code>data_size_max</code> attribute defined for the corresponding Remote Site.	Integer between 0 and 32768 (default: 0)
<code>remote_dir_name (-rdn)</code> Remote directory	Absolute path of the remote directory that stores the file sent or received. Useful in Initiator mode only.	Up to 127 alphanumeric characters
<code>remote_file_name (-rfn)</code> Remote file name	Protocol name of the file sent or received. Useful in Initiator mode only.	Up to 127 alphanumeric characters
<code>sentinel_transfer_filter (-stf)</code> Sentinel Transfer Filter	Indicate notification level for Transfer state changes to send to Sentinel server.	One of: • ALL • SUMMARY • NONE • UNDEFINED (default)
<code>snd_appli (-sa)</code> Application ID	<i>PeSIT HS E Store-and-Forward use only</i> Sender Application name	Up to 8 alphanumeric characters

Parameter	Meaning	Values
<code>snd_text (-st)</code> Message text	<i>PeSIT HS E Store-and-Forward use only</i> Sender free text	Up to 8 alphanumeric characters
<code>snd_user (-su)</code> User ID	<i>PeSIT HS E Store-and-Forward use only</i> Sender user name	Up to 8 alphanumeric characters
<code>sys_dep (-sd)</code> (none)	For BULL, AS400, GCOS and TANDEM machines: File type is system-dependent.	No value
<code>text_file</code> (none)	Text file flag	Y/N (default: N)
<code>truncating_allowed (-ta)</code> Truncating allowed	Truncating option	One of: Y = The data transferred will be truncated even if the removed data does not match the padding character N = Truncates a record only if the padding character matches the data to remove (default)

Parameter	Meaning	Values
x_data_code (- xdc) Data code	Transfer data code	One of: • B = Binary • A = ASCII • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E = EBCDIC • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • ISO = ISO • OEMPC = OEMPC • U = Undefined (default)
x_file_org (-xfg)	Type of organization of the transferred file	One of: • S = sender (default) • I = initiator • R = receiver
x_padding (-xpad) Padding character	Padding character	Hexadecimal number between 0 and FF (default: 00)
x_rec_fmt (-xrf) Record format	Transfer format	One of: • F = Fixed records • V = Variable records • T = Text (<i>OFTP only</i>) • S = Stream • U = Undefined
x_rec_len (-xrl) Record length	Transfer record length	Integer between 0 and 32768 (default: 0)

[Working with Applications \(command line\)](#)

Axway Gateway: Managing Transfers

Managing models

[Models](#)

[General Models and Protocol Models](#)

[Models for Transfer Requests](#)

[Models for Decision Rules](#)

[Models for Axway Messaging connector objects](#)

[JMS Properties and the Model object](#)

[Working with Transfer Models](#)

[Working with XMS Models](#) (for Axway Messaging)

[Working with Purge Models](#) (to purge the Mailbox)

Models Introduction

Gateway Model objects enable you to group sets of attributes for the creation of:

- Transfer Requests
- Decision Rules
- Axway Messaging connector objects

General Models and Protocol Models

For Transfer Requests and Decision Rules, a Model can either be a *General Model* or a *Protocol Model*.

When you create a Model, you begin by creating a General Model. The General Model defines a set of parameters that can be reused by any protocol. Then you can optionally create a Protocol Model, with a set of protocol-specific parameters, that bears the same name.

When you specify the Model name to use in a Transfer Request or a Decision Rule, Gateway first extracts the parameters specified in the Protocol Model and then completes the definition with parameters taken from the General Model with the same name.

Note that parameters specific to a protocol will only be used if the resulting transfer uses that protocol.

Models for Transfer Requests

Model objects are not mandatory for Transfer Requests. However, when you define a Transfer Request, you can provide certain parameters via the Model object.

Models accelerate the process of defining Transfer Requests by grouping a set of reusable Transfer attributes. Rather than redefining frequently-used parameters each time you submit a Transfer Request, you can specify a previously defined Model object as a Transfer Request parameter. If the same parameters are defined in both the Model and the Transfer, the Transfer definition takes precedence.

For example, you can use a Model to define the following:

- Transfer sender and receiver
- Application name
- Transfer validity date and time
- Scheduling priority
- User information

Models for Decision Rules

Model objects are not mandatory for Decision Rules. However, they are useful when defining processing sequences in Decision Rules.

When you define routing behavior via Decision Rule objects, you can set the execution type to *Model* and then specify the name of the Model to apply. You can specify the name of an existing Model or a Model that is not yet defined. You must however complete the Model object definition before executing the Decision Rule.

When you implement Model-based routing via Decision Rules, you must specify the final destination for the transfer either by entering a value for the **destination alias** (`dest_alias`) or the **remote agent** (`remote_agent`). You can **only** define these two Transfer parameters in a General Model, and not in a Protocol Model.

When you use Gateway to route files via a Model object:

- The Model name that you specify must correspond to a General Model definition and can optionally correspond to a Protocol Model with the same name
- The specified General Model must include a final destination
- Either the General Model or the Protocol Model must contain values for the parameters: **file component** (`file_component`) and **application** (`appli`)

Models for Axway Messaging connector objects

Use Axway Messaging connector objects to handle Axway Messaging incoming and outgoing events in Gateway. When creating an Axway [Messaging connector](#) you require at least one XMS Model object.

JMS Properties and the Model object

You can enter JMS message properties in the Model object. Gateway adds these properties to the JMS message.

To help you complete the properties in the Model, first create a Property List.

[Overview: Models](#)

[Working with Transfer Models](#)

[Working with XMS Models](#) (for Axway Messaging)

[Working with Property Lists](#) (for Axway Messaging or JMS)

[Working with Purge Models](#) (to purge the Mailbox)

[Working with Models \(command line\)](#)

[Working with Purge Models \(command line\)](#)

[Model objects: Parameters List](#)

[Managing Events on Gateway](#) (Decision Rules and Rule Tables)

Axway Gateway: Managing Transfers

Working with Transfer Models

[Creating a new Transfer Model](#)

[Creating a new General Model](#)

[Creating a new Protocol Model](#)

[Displaying a list of all Transfer Models](#)

[Displaying a list of selected Transfer Models](#)

[Viewing details of a Transfer Model](#)

[Modifying a Transfer Model](#)

[Deleting a Transfer Model](#)

Creating a new Transfer Model

As described in [About Models](#), there are two types of Transfer Model:

- General Model
- Protocol Model

When you create a new Transfer Model, first [create a General Model](#). You can then optionally [create a Protocol Model](#) with the same name.

Creating a new General Model

To create a new General Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

2. Right-click **Transfer Model**, then select **New** from the context menu.

Gateway displays the *New General Model* window.

3. Complete the fields of this window as described in the table below.
4. Click **OK**.

New General Model window (also for Protocol Models)

Field	Description
Name	<i>Only for the General Model (this field is grayed out for the Protocol Model)</i> Enter the Transfer Model name. Maximum: 31 alphanumeric characters.
Property List (optional)	<i>Only for JMS</i> Select a Property List to use with this Model from the drop-down list. The Property List you select in this field determines the contents of the Name selection box below. Note: If you create a Model and later modify it, it is not possible to modify this field (grayed out).
Comments (optional)	Enter a free-text description of the Model object. Maximum: 80 alphanumeric characters.

Field	Description
Protocol	<i>Only for the Protocol Model (this field is grayed out for the General Model)</i> Select the protocol: • PEL • PeSIT E • PeSIT D • FTP • HTTP • OFTP (Odette) • SFTP • SMTP • POP3 • EDIINT • AS1_POP3 • AS1_SMTP • AS2 • AS3 • RN_POP3 • RN_SMTP • RN_HTTP • SWIFTNet • JMS

Argument	Value

This table displays all attributes that you add to the Model object.

All JMS properties are preceded by `property_`.

Model attributes

Complete the **Argument**, **Name** (*JMS only*) and **Value** fields as described below, then click **Add** for each new attribute.

Argument	Select an argument from the list. The arguments displayed depend on the protocol. For a complete list of parameters, refer to Model Objects: Parameters List. For JMS: select jms property, then complete the Name field.

Field	Description
Name	<i>Only for JMS</i> Enter a JMS property name. If you have completed the Property List field above, select a property from the drop-down list.
Value	Press <Ctrl + space> to display the list of values you can enter for the selected argument. Alternatively, enter & to access the list of symbolic variables directly. (Make sure you first select the Argument.) This list comprises existing Gateway objects, symbolic variables and specific values appropriate to the selected argument. Alternatively, you can enter property values directly. Note: You can also use User functions, such as <code>uppercase()</code> or <code>substring()</code> when defining a Model. However these user functions are only applicable when the Model is triggered by a routing function. User functions are not applied when the Model is used for a simple (non-routed) transfer.

Creating a new Protocol Model

After creating a General Model, you can optionally create a Protocol Model with the same name.

To create a new Protocol Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

2. Right-click the General Model for which you want to create a Protocol Model. Then select **New** from the context menu.

Gateway displays the *New General Model* window.

Note: If you can only see the name of your General Model in the right pane of the GUI, and not in the left pane, you need to refresh the screen. Alternatively, close the **Transfer Model** node and then expand it once more.

3. Complete the fields of this window as described in the [table](#) above.
4. Click **OK**.

Displaying all Transfer Models

To display all existing Transfer Models:

In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

Gateway displays all existing General Models in the display frame (right frame).

Protocol Models

If you have created one or more Protocol Models for a General Model, a  appears next to the Model name.

Click the  to display the Protocol Models for that General Model.

Displaying a selected Transfer Model

You can display a single Transfer Model listed in the display pane by applying selection criteria to the display.

To display a selected Transfer Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters

2. Right-click the **Transfer Model** sub-node, then choose **Select...** from the context menu.

Gateway displays the *Select Models* window.

3. In the *Select Models* window, use the **Name** data field to enter criteria for Transfer Model display.
4. Click **OK**.

Gateway regenerates the right window display list, taking into account your selection criteria.

Viewing Transfer Model details

General Model

To view the details of a General Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

Gateway displays all existing General Models in the display frame (right frame).

2. In the display frame (right frame), right-click the General Model that you want to view. Then from the context menu, select **View**.

Gateway opens the *General Model (Read only)* window, displaying details of the Model.

Protocol Model

To view the details of a Protocol Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

Gateway displays all existing General Models in the display frame (right frame).

2. Click the  next to the name of the General Model in the left frame.

Gateway displays all existing Protocol Models for that General Model in the display frame (right frame).

3. In the display frame (right frame), right-click the Protocol Model that you want to view. Then from the context menu, select **View**.

Gateway opens the *General Model (Read only)* window, displaying details of the Model.

Modifying a Transfer Model

To modify the attributes of an existing Model:

For a complete list of parameters and possible values, refer to [Model Objects: Parameters List](#).

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

Gateway displays all existing Transfer Models in the display frame (right frame).

2. In the display frame (right frame), right-click the Transfer Model that you want to modify. Then in the context menu click **Modify...** to display the *General Model* window.
3. In the list of attributes, select the Argument that you want to modify.
4. Complete the **Argument**, **Name** (*JMS only*) and **Value** fields
5. Click **Update** to confirm the changes.

Alternatively, you can add or remove attributes.

6. Click **OK** to confirm and complete the update procedure.

Deleting a Transfer Model

To delete an existing Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Transfer Model

Gateway displays all existing Transfer Models in the display frame (right frame).

2. In the display frame (right frame), right-click the Transfer Model that you want to delete. Then from the context menu, select **Delete**.
3. Click **OK** to confirm.

The Transfer Model is deleted.

[Model objects: Parameters List](#)

Working with XMS Models

[Creating an XMS Model](#)

[Displaying a selected XMS Model](#)

[Modifying an XMS Model](#)

[Deleting an XMS Model](#)

[Displaying all XMS Models](#)

For information about Property Lists, refer to [Working with Property Lists](#).

Use XMS-type Model objects to create a link between Gateway and Axway Messaging via an Axway Messaging connector. XMS Models work exclusively with the Axway Messaging connector and do not function with other types of transfer.

Before creating an XMS Model you **must** create a Property List. The XMS Model then uses the attributes defined in the Property List.

Creating an XMS Model

To transfer data between Axway Messaging and Gateway, you require an XMS-type Model.

To create a new XMS Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

Note: You may need to refresh the display by right-clicking **Property list** and selecting **Refresh**.

2. Right-click a Property List name and select **New** from the context menu to display the first *New XMS Model* window.
3. In the **Name** field, enter a name for the new XMS Model.
4. In the **Direction** field, select a server direction:
 - **To Gateway:** for transmitting incoming file transfers to an Axway Messaging server message queue
 - **From Gateway:** for transmitting the messages of an Axway Messaging server message queue to a file transfer application via a Gateway Transfer Request
5. Click **OK** to confirm.
Gateway displays a second *New XMS Model* window.

6. Complete the second *New XMS Model* window:

- In the **Comments** field, enter a comment (optional)
- Select an **Argument** from the drop-down list
- Use <Ctrl + space> to list possible values and <&> to list symbolic variables to define the argument
- Click **Add** to confirm this argument. Repeat until you have entered all arguments. You can use an existing transfer to aid you in deciding which arguments are useful for a given XMS Model.

The Axway Messaging connector attributes table below lists and defines commonly used arguments.

7. Click **OK** to confirm.

Axway Messaging connector attributes

Argument	Description
Type	For outgoing messages, enter one of the following values: • MSG = Axway Messaging message handling • ERROR_ACK = error and acknowledgment handling
Directory Name	For outgoing messages, enter the directory in which Axway Gateway stores Axway Messaging data content to be sent.
Max size	Enter the maximum message size to be handled.
Compression	For incoming files transferred to Axway Messaging, enter one of the following values: • Y (yes) • N (no)
Priority	For incoming files, enter one of the following values to indicate the priority an application can use as a criteria for getting messages from a source: • LOW = The message has low priority • MEDIUM = The message has intermediate priority • HIGH = The message has high priority

Displaying a selected XMS Model

To view the attributes of a selected XMS Model

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model

2. Right-click the **Property List** sub-node, then choose **Select...** from the context menu.

Gateway displays the *Select Models* window.

3. In the *Select Models* window, use the **Name** field to enter criteria for the Model display selection.
4. Click **OK**.

Gateway regenerates the right window XMS Model display, taking into account your selection criteria.

Modifying an XMS Model

To modify and update an XMS Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

2. Click to select the Property List that contains the Model you want to modify.

In the display pane (right pane) Gateway displays the XMS Models contained in the selected Property List folder.

3. Right-click the Model that you want to modify and select **Modify...** in the context menu. The corresponding Model window is displayed.
4. Use the **Add**, **Update**, and **Delete** buttons to update the Model.
5. Click **OK** to confirm these changes.

Deleting an XMS Model

To delete an XMS Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

2. Click to select the appropriate Property List that contains the Model you want to delete.

In the display pane (right pane) Gateway displays the XMS Models contained in the Property List folder you selected.

3. Right-click the Model that you want to delete and select **Delete** in the context menu.
4. Gateway displays a confirmation box. Click **Yes** to confirm.

Repeat for each XMS Model that you want to delete.

Displaying all XMS Models

To display all XMS Models:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

2. Click to select the Property List that contains the Model(s) you want to display.

In the display pane (right pane) Gateway displays the XMS Models contained in the Property List folder you selected.

[Axway Messaging connector](#)

[JMS connector](#)

[Working with Property Lists](#)

Axway Gateway: Managing Transfers

Working with Property Lists

[Displaying all Property Lists](#)

[Creating a Property List](#)

[Viewing the attributes of a Property List](#)

[Modifying a Property List](#)

[Deleting a Property List](#)

In Gateway, Property Lists are used for both Axway Messaging and JMS.

For Axway Messaging, you must create a Property List before you create an XMS Model. The XMS Model then uses the attributes you defined in the Property List.

For JMS, the use of Property Lists is optional. A Property List makes it easier to add properties in a Model or Decision Rule.

Gateway uses the elements in a Property List to:

- associate Transfer Request parameters with incoming Axway Messaging or JMS messages
- provide information to Axway Messaging or JMS message destination applications about the contents of messages sent via Gateway

Displaying all Property Lists

Before you can create an XMS Model, you must have at least one Property List.

To display all existing Property Lists, in the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays all existing Property Lists.

Note: You may need to refresh the display by right-clicking **Property list** and selecting **Refresh**.

Creating a Property List

If no Property List exists, or if you require a new list, create a list as follows:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

2. Right-click the **Property list** sub-node and select **New** from the context menu.

Gateway displays the *New Property List* window.

3. Complete the fields of the *New Property List* window as follows:

New Property List window

Field	Description
Name	Enter the name of the Property List. Maximum: 25 alphanumeric characters.
Comments	Enter a free-text description for the Property List. Maximum: 80 alphanumeric characters.
Property For each property you add, enter the Name and Type . Then click Add to include this property in the Property List.	
Name	Enter the property name.
Type	Select the property type: • Boolean • Char • Double • Float • Int • Long • Short • String

4. Click **OK** to confirm and create the new Property List.

Viewing the attributes of a Property List

To view the attributes of a specific Property List:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

2. In the Directory tree structure, right-click the Property List that you want to view, and select **View...** from the context menu.

Gateway displays the *Property List (read only)* window. This window provides you with a view of all attributes for the selected Application.

3. After viewing the information, click **Close** to quit the window.

Modifying a Property List

To modify and update an existing Property List:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

2. Right-click the **Property list** node and select **New** from the context menu.

Gateway displays the *New Property List* window.

3. Make the modifications in the Property List and click **Update** to take into account the new parameters. For details about the parameters of the Property List, refer to [New Property List window](#) (above).
4. Click **OK** to confirm.

Repeat the procedure for each Property List you want to update.

Deleting a Property List

To delete an existing Property List:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > XMS Model > Property list

In the Directory tree structure, Gateway displays the names of existing Property Lists.

2. In the Directory tree structure, right-click the Property List that you want to delete, and select **Delete...** from the context menu.

Note: You cannot delete a Property List that contains any Model objects. If the **Delete** command is not active in the context menu, check the display pane to see if it contains any Model objects for the Property List. Delete any Model objects, then you can delete the Property List.

3. Click **OK** to confirm.

Repeat the procedure for each Property List you want to delete.

[Axway Messaging connector](#)

[JMS connector](#)

[Working with Transfer Models](#)

[Working with XMS Models](#)

Axway Gateway: Managing Transfers

Working with Purge Models

[Overview](#)[Creating a Purge Model](#)[Display a list of all Purge Models](#)[Display a list of selected Purge Models](#)[View details of a Purge Model](#)[Modify a Purge Model](#)[Delete a Purge Model](#)

Overview

The *Purge Model* is a special category of Model object that you can use for selective purge operations. When you have created one or more Purge Models, you can associate the Purge Models with the *default execution results* of a Scheduling-type Decision Rule event. Refer to [Rule Table: General tab](#).

Example

- You create a Purge Model named *PurgeNonLocal* in which you associate the attributes of the originators of transfers to the values of your foreign agencies.
- You create a Scheduling-type Decision Rule in which:
 - You define a daily scheduled event at 6 p.m.
 - You set the Default execution type to *Mailbox cleaning*
 - You indicate the *PurgeNonLocal* Purge Model as the associated Model

Result: Each day at 6 p.m., Gateway purges all records of transfers in the Mailbox with originating addresses from your foreign offices.

Creating a new Purge Model

To create a new Purge Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Purge Model

2. Right-click **Purge Model**, then select **New** from the context menu to display the *New Purge Model* window.

3. Complete the fields of this window as described in the following table.
4. After completing the fields in this window, click **OK**.

New Purge Model window

Field	Description
Name	Enter the Purge Model name. Maximum: 31 alphanumeric characters.
Comments	<i>Optional</i> Enter a free-text description of the Purge Model object. Maximum: 80 alphanumeric characters.
Argument Value	
All arguments that you add to the Purge Model object are listed in this pane.	
Model attributes Use the Argument and Value fields in this section, in combination with the Add , Update , Delete and Clear buttons, to create your list of arguments for the Purge Model.	
Argument	Select an argument from the list and then complete the Value field.
Value	Press <Ctrl + space> to display the list of values you can enter for the selected argument. Alternatively, enter & to access the list of symbolic variables directly. (Be sure to first select the Argument .) This list comprises existing Gateway objects, symbolic variables and specific values appropriate to the selected argument. Click Add to add the argument to the list. The combination of an argument and associated value defines the filtering criteria for the purge operation. When you initiate a purge using this Model, Gateway purges all records from the Mailbox that contain this Argument/Value pair and any other pairs defined for this Model.

Displaying all Purge Models

To display all existing Purge Models, in the left pane of the GUI main window click  to expand the nodes:

Transfer Management > Parameters > Model > Purge Model

Gateway displays any existing Purge Models in the display frame (right frame).

Displaying a selected Purge Model

You can display a single Purge Model listed in the display pane by applying selection criteria to the display.

To display a selected Purge Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters

2. Right-click the **Purge Model** sub-node, then choose **Select...** from the context menu.

Gateway displays the *Select Models* window.

3. In the *Select Models* window, use the **Name** data field to enter criteria for Purge Model display.
4. Click **OK**.

Gateway regenerates the right window display list, taking into account your selection criteria.

Viewing Purge Model details

To view the details of a specific existing Purge Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Purge Model

Gateway displays any existing Purge Models in the display frame (right frame).

2. In the display frame (right frame), right-click the Purge Model whose characteristics you want to view. Then from the context menu, select **View**.

Gateway opens the *Purge Model (Read only)* window, displaying details of the Purge Model.

Modifying a Purge Model

To modify the parameter contents of an existing Purge Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Purge Model

Gateway displays any existing Purge Models in the display frame (right frame).

2. In the display frame (right frame), right-click the Purge Model that you want to update. Then in the context menu click **Modify...** to display the *Purge Model* window.
3. Select the [Argument](#) that you want to modify. You can also add or delete Arguments.
4. Complete the **Argument** and **Value** fields. Click **Update** to confirm these changes.
5. Click **OK** to confirm and complete the update procedure.

Deleting a Purge Model

To delete an existing Purge Model:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Parameters > Model > Purge Model

Gateway displays any existing Purge Models in the display frame (right frame).

2. In the display frame (right frame), right-click the Purge Model that you want to delete. Then from the context menu, select **Delete** to remove the Purge Model.
3. Click **OK** to confirm and complete the delete procedure.

Axway Gateway: Managing Transfers

Command line operations

Working with models (command line)

[peladm create model](#)

[peladm update model](#)

[peladm delete model](#)

[peladm check model](#)

[peldsp display model](#)

[peldsp select model](#)

Creating a model: peladm create_model

Syntax	<code>peladm create_model {-model} [optional parameters]</code>

Description	Use this command to create a Model. Each Model is uniquely defined by the following two attributes: • <code>-model (-ml)</code> • <code>-protocol (-pr)</code> For XMS-type Models: To create an XMS-type Model for transfers to Gateway from an Axway Messaging server, use the following parameters: • <code>-model (-ml)</code> • <code>-protocol (-pr)</code> : Enter the value to_gtw • <code>-prop_list_name (-pln)</code> • <code>-appli (-app)</code> • one or more attributes necessary to enable Gateway to reroute the message to the destination application To create an XMS-type Model for transfers from Gateway to an Axway Messaging server, use only the following parameters: • <code>-model (-ml)</code> • <code>-protocol (-pr)</code> : Enter the value from_gtw • <code>-prop_list_name (-pln)</code> • <code>-add_map (-am)</code>
Parameters	<code>-model (-ml)</code> : Enter the name of the Model. (max 25 characters)
	<i>Optional parameters:</i> Refer to Model objects: Parameters List
Example 1	The following example creates a General Model named MODELA. <pre>peladm create_model -ml MODELA -pr GENERAL -dest_alias TPHSE -appli DEFAULT_B</pre>
Example 2	The following example creates an XMS-type Model for a transfer from an Axway Messaging server towards Gateway. <pre>peladm create_model -ml XMStoGTW -pr to_gtw -pln plist -appli &(application)</pre>

Example 3	The following example creates an XMS-type Model for a transfer from Gateway towards an Axway Messaging server. The <code>add_map</code> parameter directs Gateway to retrieve <i>appli</i> and <i>originator</i> values from the current transfer to use as <i>application</i> and <i>server</i> values of the proplist. The <code>add-map</code> parameter also sets the <code>cost</code> parameter of the properties list to the value 2000. <pre> peladm create_model -ml GTWtoXMS -pr from_gtw -pln plist -add_map "application=&(appli) ; cost=2000 ; server=& (originator) " </pre>

Modifying a Model: peladm update_model

Syntax	<code>peladm update_model {-model} [optional parameters]</code>
Description	Use this command to modify an existing Model. If you update a Model using a parameter without a value, the command deletes the last parameter value. Enter the Model name and Model protocol as parameters, followed by all the fields you want to modify. The default Model protocol is GENERAL.
Parameters	<code>-model (-ml)</code> : Enter the name of the existing model you want to modify. <i>Optional parameters:</i> Refer to Model objects: Parameters List.
Example	The following example modifies the Application object used by a General Model named MODELA. <pre> peladm update_model -ml MODELA -pr GENERAL -appli DEFAULT_A </pre>

Deleting a model: peladm delete_model

Syntax	<code>peladm delete_model {-model} [-protocol] [optional parameters]</code>
Description	Use this command to delete a Model. Since you can create a General Model and a protocol Model with the same name, you should specify both the name and protocol of the Model that you want to delete.

Parameters	<code>-model (-ml)</code> : Enter the name of the existing Model that you want to delete.
	<code>-protocol (-pr)</code> : Enter the name of the protocol of the Model that you want to delete. Enter one of the following values: • GENERAL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • PEL • FTP • HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_POP3 • RN_SMTP • RN_HTTP
	<i>Optional parameters:</i> Refer to Model objects: Parameters List
Example	<pre>peladm delete_model -ml MODELA -pr GENERAL</pre>

Checking Model properties: peladm check_model

Syntax	<code>peladm check_model {-model} [-protocol]</code>
Description	Use this command to verify the coherence between the properties of a Model and the names of the properties used in the Model. Since a General Model and a Protocol Model can share the same name, you should specify both the name and protocol of the Model that you want to check.

Parameters	<code>-model (-ml)</code> : Enter the name of the existing Model that you want to check.
	<code>-protocol (-pr)</code> : Enter the name of the protocol of the Model that you want to check. Enter one of the following values: • GENERAL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • PEL • FTP • HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_POP3 • RN_SMTP • RN_HTTP
Example	<code>peladm check_model -ml MODELA -pr GENERAL</code>

Selecting models: peldsp select_model

Syntax	<code>peldsp select_model {-model} [-protocol] [-prop_list_name]</code>
Description	Use this command to select the Models that match the criteria entered as parameters. To list all Models, enter no parameters. You can refine your search using the optional parameters.

Parameters	<code>-model (-ml)</code> : Enter the name of the existing Model that you want to select.
	<code>-protocol (-pr)</code> : Enter the name of the protocol of the Model that you want to select. Enter one of the following values: • GENERAL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • PEL • FTP • HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_POP3 • RN_SMTP • RN_HTTP
	<code>-prop_list_name (-pln)</code> : Enter the name of the properties list that you want to select.
Example	When you enter the command, Model names and protocols that match the selection criteria are listed on screen. The following command returns all Models with the FTP protocol. <pre>peldsp select_model pr FTP</pre> returns (for example): <pre>MODEL B MODEL_FTP</pre>

Displaying a model: `peldsp display_model`

Syntax	<code>peldsp display_model {-model} [-protocol]</code>
Description	Use this command to display the details of a Model identified by its name and protocol. The Transfer parameters are displayed by return.

Parameters	<code>-model (-ml)</code> : Enter the name of the existing Model that you want to display. <code>-protocol (-pr)</code> : Enter the name of the protocol of the Model that you want to display. Enter one of the following values: • GENERAL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • PEL • FTP • HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_POP3 • RN_SMTP • RN_HTTP
Example	Each parameter is displayed on a separate line: <pre>name_of_field="value"</pre> Each field name is prefixed with the string "m_", relating to the Model object. Only supplied fields are displayed. <pre>peldsp display_model -ml MODELA -pr GENERAL m_model='MODELA' m_protocol='GENERAL' m_comments='' m_dest_alias='TPHSE' m_appli='DEFAULT_A'</pre>

[Model objects: Parameters List](#)

Model objects: Parameters List

This topic lists the parameters that are available for managing Model objects.

Although a set of suggested values is provided, keep in mind that you can use symbolic variables to specify all values. However, if you use symbolic variables in a Model, Gateway does not substitute these variables when you submit a Transfer Request via the `peltrans` command.

[Using symbolic variables](#)

[Using User functions](#)

[Model object parameters](#)

[Additional Model parameters for SWIFTNet](#)

Using symbolic variables

To retrieve values specified in the incoming Transfer, use symbolic variables in the **Value** field for a given parameter. For example:

```
destination alias = &(dest_alias)
```

Gateway retrieves the **destination alias** value from the incoming Transfer and applies it to the outgoing Transfer.

You can also use symbolic variables to retrieve local system time values:

- `&(date)`
- `&(time)`
- `&(yday)`

Using User functions

In addition, you can use the [user functions](#) `substring()`, `substr()`, `sepstring()`, `strep`, `lowercase()` and `uppercase()`.

Model object parameters

The following table lists Model object parameters (in alphabetical order) that you can use with the `peladm` and `peldsp` commands. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Meaning	Values
<code>acknowledgment_option (-ao)</code> <i>acknowledgment option</i>	Specifies acknowledgement behavior for this transfer	One of the following values: • TO_ACK = Awaiting acknowledgement • NOT_TO_ACK = Do not wait for acknowledgement • ACK_UNKNOWN = Unspecified acknowledgement behavior • Symbolic variable

Parameter	Meaning	Values
<code>add_map (-am)</code>	<i>For XMS-type Models only,</i> when you set the value of <code>-protocol (-pr)</code> to FROM_GTW. Use this parameter to map values to one or more attributes of a properties list. The associated properties list is the list you specify in the <code>prop_list_name</code> parameter. To map each value, use the syntax: <code>name=value</code> where: <code>name</code>: is the name of one of the attributes of the associated parameters list <code>value</code>: is either a value you specify or a value called from the current transfer attributes via a symbolic variable using the syntax & <code>(attribute_name)</code> Example: To apply the value of the <code>-appli</code> attribute of the current transfer to the <code>application</code> attribute of the parameter list, enter: <pre>-add_map "application=& (appli) "</pre>	Associate a name of one or more parameter list attributes with a value. The value can be called from the current transfer values by means of a symbolic variable. Use the syntax name1=value1 name2=value2 ... nameN=valueN

Parameter	Meaning	Values
append (-app) append	<i>FTP client only</i> Authorize the addition of data to the end of the file when received on the server.	One of: • Y = append allowed • N = append forbidden • Symbolic variable
appli (-ap) application	Application name	Maximum: 25 alphanumeric characters
attach_extract_file (-aef) attach extract file	<i>SMTP & POP3 only</i> Use this option to specify whether to attach or extract a file or not.	One of: • Y if the file is not formatted as an email • N if the file is formatted as an email • U must open the file to determine how to proceed • Symbolic variable
block_size (-bs) block size	BULL machines: Block size in bytes	Integer
comments (-cts) Comments	User comment	Maximum: 80 alphanumeric characters
compression (-comp) compression	Compression type	One of: • U = Undefined • H = Horizontal • V = Vertical • B = Both • Z = Zlib • Symbolic variable
content_type (-dct) content type	Optional transfer content information field	A free text description of the transfer content type

Parameter	Meaning	Values
data_code (-dc) data code	Data code	One of: • B = Binary • A = ASCII • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E = EBCDIC • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • ISO = ISO • OEMPC = OEMPC • U = Undefined • Symbolic variable
date_to_begin (-dtb) date to begin	Start date (requested)	In the format YYYYMMDD
date_to_end (-dte) date to end	End date (requested)	In the format YYYYMMDD
delete_map	Delete one or more mapping	• name1 • name2 • ... • nameN
dest_alias (-da) destination alias	<i>General Model only</i> Destination Site alias name	Maximum: 25 alphanumeric characters

Parameter	Meaning	Values
destination (-dest) destination	Destination protocol identifier	Maximum: 25 alphanumeric characters
direction (-dir) direction	Transfer direction	One of: • I = Input • O = Output • Symbolic variable
dir_name (-dn) directory name	<i>FTP & HTTP only</i> Absolute path of the directory that stores the file to be transferred. It can be used in one of the following modes: • Initiator mode, in this case it is a directory loaded on the remote system • Responder mode, in this case it is a directory in the VFD device	Maximum: 127 alphanumeric characters
dir_name_alias (-dna) dir name alias	<i>FTP and HTTP only</i> Alias of the VFD directory that stores the file to be transferred. Only available in Responder mode.	Maximum: 32 alphanumeric characters

Parameter	Meaning	Values
dir_type (-dt) dir type	<i>FTP only</i> DIR Transfer types in Initiator / Receiver mode only. Specifies how to list the directory contents.	One of: • LONG: the LIST command returns a detailed list of directory contents. For example, this option differentiates between files and directories. The LONG format depends on the server and is typically in a UNIX format. • NAMES (default): the NLST command returns a basic list of directory contents. This option does not differentiate between files and directories, and provides no extra information such as the last modification date, the file size, and so on. • Symbolic variable
external_sentinel_ident (-esi)	Sentinel identifier that the external application passes in batch mode only. This identifier is not stored in the Transfer record.	Maximum: 250 alphanumeric characters
external_sentinel_objectname (-eso)	Sentinel object name to be passed by external applications. Batch mode only. Not stored in the Transfer record.	Maximum: 32 alphanumeric characters
file_component (-fc) file component	Local physical file name	Maximum: 127 alphanumeric characters

Parameter	Meaning	Values
file_label (-fl) file label	<i>PeSIT only</i> Protocol label of the file	Maximum: 256 alphanumeric characters
file_name (-fn) file name	Protocol file name	Maximum: 76 alphanumeric characters
file_org (-f_org) file organization	File organization	One of: • S = Sequential file • I = Indexed file • R = Relative file • Symbolic variable
file_type (-ft) file type	<i>PeSIT only</i> File type (protocol)	Maximum: 16 alphanumeric characters
frozen_state (-fs) frozen state	Create the Request in frozen state	State F = Frozen
ftp_command (-ftpc) ftp command	<i>FTP, SFTP & HTTP only</i> FTP and SFTP: Use to define site and quote commands HTTP: Use to add CGI parameters to the Transfer Request (format: <i>xfer_param1=value1&xfer_param2=value2</i>)	Maximum: 80 alphanumeric characters For FTP and SFTP, you can overcome the 80 character limitation by calling a script that contains a set of FTP commands (site , quote)
generate_xfb_subject (-gxs) generate xfb subject	<i>SMTP & POP3 only</i> Specify whether or not to include specific Gateway parameters in the <i>Subject</i> field.	One of: • Y (yes) • N (no) • U (undefined) • Symbolic variable
hist_create_date (-hd) hist create date	<i>PeSIT only</i> PeSIT creation date	In the format <i>YYYYMMDD HHMMSS</i>

Parameter	Meaning	Values
interval (-int) interval	<i>PEL & PeSIT only</i> • PEL: interval between two Remote Site polls (LOTS and POLL commands) • PeSIT HS: repetition interval in seconds, in case of selection failure	Value in seconds
local_agent (-la) local agent	Alias of the associated Local Site	Maximum: 25 alphanumeric characters
local_site (-local) local Site	<i>PEL, PeSIT & FTP only</i> Local Site name	Maximum: 25 alphanumeric characters
max_file (-m_file) max file	<i>PeSIT only</i> PeSIT HS cyclic selection request only: maximum number of files allowed.	Decimal value: 0 - 32000 . -1 = no imposed limits.
max_request (-m_req) max request	<i>PeSIT only</i> PeSIT HS cyclic selection request only: maximum number of requests allowed.	Decimal value: 0 - 32000 . -1 = no imposed limits.
mode (-m) mode	Connection mode for the Transfer	One of: • I = Initiator • R = Responder • Symbolic variable
model (-ml) Name	Model name	Maximum: 25 alphanumeric characters
nack_action (-nack) nack action	<i>PeSIT only</i> Indicates whether the Remote Site supports the negative EERP extension or not.	One of: • Y • N • Symbolic variable

Parameter	Meaning	Values
newline_ convention (-nlc) newline convention	<i>SFTP only</i> Line feed convention, depending on remote system type.	One of: • U / u (UNIX) • W / w (Windows) • empty • Symbolic variable
org_alias (-oa) originator alias	<i>General Model only</i> Originator Site alias name	Maximum: 25 alphanumeric characters
originator (-org) originator	Originator protocol identifier	Maximum: 25 alphanumeric characters
padding (-pad) padding	Padding character code	Integer between 00 and FF or a symbolic variable
param1 (-p1) param1	User parameter 1	
param2 (-p2) param2	User parameter 2	
permanent (-perm) permanent	Transfer Request permanent flag	One of: • Y • N • Symbolic variable
priority (-prio) priority	Transfer priority	One of: • 0 = High priority • 1 = Medium priority • 2 = Low priority
prop_list_name (- pln)	<i>XMS-type Models only</i> Property List Name	Maximum: 25 alphanumeric characters
property (-prop) jms property	<i>JMS only</i> JMS property	

Parameter	Meaning	Values
protocol (-pr) Protocol	Protocol Name	One of: • GENERAL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • PEL • FTP • HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • POP3 • TO_GTW (<i>For XMS-type Models only</i>) • FROM_GTW (<i>For XMS-type Models only</i>) • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_HTTP • RN_POP3 • RN_SMTP • SWIFTNET • JMS
purge_option (-po) purge option	Enable purge of Transfer Requests from Mailbox using the pelpur command.	One of: • Y • N • Symbolic variable
rcv_appli (-ra) rcv appli	<i>PeSIT HS E, Store-and-Forward only</i> Receiver Application name.	Maximum: 8 alphanumeric characters

Parameter	Meaning	Values
rcv_text (-rt) rcv text	<i>PeSIT HS E, Store-and-Forward only</i> Receiver free text.	Maximum: 8 alphanumeric characters
rcv_user (-ru) rcv user	<i>PeSIT HS E, Store-and-Forward only</i> Receiver user name.	Maximum: 8 alphanumeric characters
rec_fmt (-rf) rec format	Record format	One of: • F = Fixed • FT = Fixed Text • V = Variable • VT = Variable Text • S = Stream • Symbolic variable
rec_len (-rl) rec length	Record length	Integer: 0 - 32768
receipt_nack_msg (-rnm) receipt nack msg	<i>EDIINT only</i> <i>Gateway operating in Responder mode on receipt of a transfer.</i> Optionally, enter a text explaining a warning or error receipt.	Maximum: 512 characters
receipt_reply_to_xfer (-rrtx) receipt reply to xfer	<i>EDIINT and SWIFTNet only</i> <i>Gateway operating in Responder mode on receipt of a transfer.</i> Use this parameter to activate the generation of a receipt on reception of a transfer.	The number of the transfer for which you want to generate and send a return receipt.

Parameter	Meaning	Values
receipt_request (-rr) receipt request	<i>EDIINT and SWIFTNet only</i> Specify whether or not you want to receive a receipt for the reception of a file you send. If you choose to receive a receipt, specify whether the receipt is signed or unsigned.	One of: • none = no receipt requested (default) • signed = request signed receipt • unsigned = request unsigned receipt (<i>EDIINT only</i>)
receipt_request_to (-rrto) receipt request to	<i>EDIINT and SWIFTNet only</i> EDIINT: URL or email address to which the acknowledgement is sent. SWIFTNet: Name of the receipt reception address. This is the address to which the receiving partner sends the receipt.	The URL, email address or DN to which you want the transfer receipt to be sent. Maximum: 256 alphanumeric characters
receipt_request_sync (-rrs) receipt request sync	<i>EDIINT only</i> Synchronize sending of receipts.	One of: • Y • N
receipt_type (-rt) receipt type	<i>EDIINT only</i> <i>Gateway operating in Responder mode on receipt of a transfer.</i> Specify the type of receipt to return to the initiator of the received transfer.	One of: • SUCCESS • ERROR
remote_agent (-ra) remote agent	<i>General Model only</i> Alias of the associated Remote Site.	Maximum: 25 alphanumeric characters
reply_to_xfer (-rtx) reply to xfer	Local identifier of the Transfer Request to which the response must be sent.	Integer

Parameter	Meaning	Values
<code>request_method (-rm)</code> <i>request method</i>	<i>HTTP only</i> HTTP method to use during transfer.	• GET or POST for incoming transfers • POST or PUT for outgoing transfers • Symbolic variable
<code>retry_count_max (-rc_max)</code> <i>retry count max</i>	Maximum number of connection retries.	Integer: 0 - 999
<code>smtp_bcc (-bcc)</code> <i>smtp bcc</i>	<i>SMTP & POP3 only</i> Blind carbon copy addresses.	No max limit
<code>smtp_cc (-cc)</code> <i>smtp cc</i>	<i>SMTP & POP3 only</i> Carbon copy addresses.	No max limit
<code>smtp_from (-from)</code> <i>smtp from</i>	<i>SMTP & POP3 only</i> From address.	Maximum: 128 characters
<code>smtp_subject (-subject)</code> <i>smtp subject</i>	<i>SMTP & POP3, AS2 & AS3 only</i> Email subject.	Maximum: 128 characters
<code>smtp_to (-to)</code> <i>smtp to</i>	<i>SMTP & POP3 only</i> Destination addresses.	Maximum: 128 characters
<code>snd_appli (-sa)</code> <i>snd appli</i>	<i>PeSIT HS E, Store-and-Forward only</i> Sender Application name.	Maximum: 8 alphanumeric characters

Parameter	Meaning	Values
snd_msg (-sm) snd msg	User message to send	• OFTP: User message to be transmitted at OFTP connection phase (maximum: 8 alphanumeric characters) • FTP: List of special commands to send to remote server using SITE or QUOTE FTP command • PeSIT: User message to be transmitted at the connection phase (maximum: 254 alphanumeric characters) • HTTP: User message is used to add CGI parameters to Transfer Requests, formatted as follows: <i>xfer_param1=value1</i> <i>& xfer_param2=value2</i> • SWIFTNet: Refer to Additional Model parameters for SWIFTNet.
snd_text (-st) snd text	<i>PeSIT HS E, Store-and-Forward only</i> Sender free text.	Maximum: 8 alphanumeric characters
snd_user (-su) snd user	<i>PeSIT HS E, Store-and-Forward only</i> Sender user name.	Maximum: 8 alphanumeric characters
sys_dep (-sd) sys dep	For BULL, AS400, GCOS and TANDEM machines: File type is system-dependent.	No value

Parameter	Meaning	Values
text_file text file	Text file flag	One of: • Y • N • Symbolic variable
tpm_dea_id (-tpmdea) tpm dea id	<i>EDIINT only</i> PassPort (TPM) Document Exchange Address identifier.	Integer
trade_ack_req (-tradeack) trade ack req	<i>EDIINT only</i> Acknowledgement required when sending a file to a partner.	One of: • undef (default) • none • signed • unsigned
trade_ack_sync (-tradeas) trade ack sync	<i>EDIINT only</i> Synchronize acknowledgement.	Y/N
trade_compress (-tradecomp) trade compress	<i>EDIINT and SWIFTNet only</i> Compression algorithm for use when sending a file to a partner.	Y/N
trade_dest (-traded) trade destination	<i>EDIINT and SWIFTNet only</i> The name of the destination partner of the trading partner exchange.	Maximum: 128 characters
trade_dest_alias (-tradedea) trade destination alias	<i>EDIINT and SWIFTNet only</i> Alias of the destination partner of the trading partner exchange.	Maximum: 128 characters
trade_encrypt (-tradee) trade encrypt	<i>EDIINT and SWIFTNet only</i> Encryption algorithm for use when sending a file to a partner.	Y/N

Parameter	Meaning	Values
trade_file_ already_encoded (-tradedfae) trade file encoded	<i>EDIINT only</i> The use of this parameter is not recommended for transfer deposits. Use this parameter when creating a Model to be applied on Decision Rule routing via EDIINT.	Y/N
trade_format (-tredef) trade format	<i>EDIINT and SWIFTNet only</i> Indicate protocol version.	One of: • AS1 • AS2 • AS3 • SMIME • SWIFTNet • RN
trade_org (-tradeo) trade originator	<i>EDIINT and SWIFTNet only</i> The name of the originating partner of the trading partner exchange.	Maximum: 128 characters
trade_org_alias (-tradeoa) trade originator alias	<i>EDIINT and SWIFTNet only</i> Alias of the originating partner of the trading partner exchange.	Maximum: 128 characters
trade_sign (-trades) trade sign	<i>EDIINT and SWIFTNet only</i> Certificate signing algorithm used when sending a file to a partner.	One of: • Y • N
truncating_allowed (-ta) truncating allowed	If the transfer uses the truncating feature, this option allows you to truncate a record without taking into account the padding character.	One of: • Y • N • Symbolic variable

Parameter	Meaning	Values
type (-ty) type	Transfer type	One of: • DIR • TRANS • LOTS • POLL • LIST • EERP • SELECT • RECEIPT • MSG • Symbolic variable
unique_filename (-uf) unique filename	<i>FTP only</i> FTP store unique option	One of: • Y • N • Symbolic variable
update_map (-um)	Update one or more mappings	• name1=value1 • name2=value2 • nameN=valueN

Parameter	Meaning	Values
x_data_code (- xdc) xfer data code	Transferred data code	One of: • B = Binary • A = ASCII • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E = EBCDIC • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • ISO = ISO • OEMPC = OEMPC • Symbolic variable
x_rec_fmt (-xrf) xfer rec format	Transfer format	One of: • F = Fixed • V = Variable • T = Text (<i>OFTP only</i>) • S = Stream • Symbolic variable
x_rec_len (-xrl) xfer rec length	Transfer record length	Integer: 0 - 32768
mail_xpriority (-mailxprio)	SMTP protocol only (with or without S/MIME) - the value for X-Priority mail header (SEPA mail standard)	Integer, between 1 and 5

Parameter	Meaning	Values
<code>sigalgo (-sga)</code>	Signature and hash algorithm used in payload signature computation. Considered only when <code>payload_integrity_option</code> is set to <code>yes</code> . See Working with Payload integrity .	SHA256withRSA
<code>sigfile (-sgf)</code>	File containing payload signature, in binary format. Considered only when <code>payload_integrity_option</code> is set to <code>yes</code> .	

Additional Model parameters for SWIFTNet

The following table lists `peladm` and `peldsp` command parameters that are specific to SWIFTNet transfers. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Meaning	Value
<code>-payload_type (-pto)</code> <i>sw payload type</i>	<i>InterAct only</i> SWIFTNet Interact payload format. On the <i>Initiator</i> side, you can use this parameter to overwrite the Payload type parameter defined in the Trading Partner. On the <i>Responder</i> side, when using the Receipt request synchronized option you can respond with a custom payload type. Even if you saved the request payload as embedded in the Trading Partner, you can answer back with text mode.	• (empty) = undefined • <code>text</code> • <code>embedded</code>

Parameter	Meaning	Value
-rcv_msg rcv msg	<i>FileAct only</i> [0 - 255]: File description, supplied by the sender. Free-text description of the file. [256 - 511]: File information, supplied by the sender. Structured information concerning the transfer. For more details, refer to the SWIFT documentation.	Maximum: 256 + 256 alphanumeric characters
-snd_msg (-sm) snd msg	FileAct [0 - 255]: Transfer description, supplied by the initiator. Free-text description of the transfer. FileAct [256 - 511]: Transfer information, supplied by the initiator. Structured information concerning the transfer. For more details, refer to the SWIFT documentation. InterAct [0 - 511]: User message.	FileAct: Maximum 256 + 256 alphanumeric characters InterAct: Maximum 511 alphanumeric characters
-sw_add_params (-swaddpar) sw add params	Reference to file on disk containing additional SWIFTNet structures (for example, Header Info and/or Signature List). This file must respect a specific structure .	
-sw_auth_indicator (-ai) sw auth indicator	SWIFTNet Authorization notification indicator. Select this option to receive Authorization system messages for files or messages sent over Y-copy services .	One of: • Y = yes • N = no (default)

Parameter	Meaning	Value
-sw_copy_indicator (-ci) sw copy indicator	SWIFTNet File or Message Copy service indicator. Select this option to request a copy of the file or message to be sent to a third party defined at the service level.	One of: • y = yes • n = no (default)
-sw_message_id (-swmsgid) swift message id	The value of this parameter overwrites the message id created by Gateway in the Transfer Request. Enter the SWIFTNet message id of the new transfer. This should be a Universal Unique Identifier (UUID). If the transfer is a Possible Duplicate, enter the message id of the initial transfer.	Maximum: 40 alphanumeric characters
-sw_overdue_time (-swot) sw overdue time	SWIFTNet OverdueTime	In the format YYYYMMDD HHMMSS
-sw_pd_indication (-swpdind) possible duplicate	Possible Duplicate indicator.	One of: • y = yes. Select this value to indicate to the responder that the transfer is a Possible Duplicate. • n = no (default) • Symbolic variable

Parameter	Meaning	Value
-trade_delivery_mode (-tradedmo) trade delivery mode	The delivery mode of the file or message.	One of: • RT = Real time (immediate) • SNF = Store-and-Forward (send and then store in the queue in the SWIFTNet domain) • Symbolic variable
-trade_encrypt_dn (-tradeendn) trade encrypt dn	<i>InterAct only</i> If specified, name used for encryption in Initiator mode.	Maximum: 100 alphanumeric characters
-trade_non_repudiation (-tradenr) trade non repudiation	Request SWIFT to keep proof of sending.	One of: • Y • N • Symbolic variable
-trade_notif_queue (-tradenq) trade notif queue	<i>Store-and-Forward mode only</i> Reception queue for Store-and-Forward delivery notifications.	Maximum: 30 alphanumeric characters
-trade_receipt_request_type (-traderrrt) trade receipt request type	Type of acknowledgement request.	Maximum: 30 alphanumeric characters
-trade_request_type (-traderqt) trade request type	Type of request (as specified in the XML header).	Maximum: 30 alphanumeric characters
-trade_service (-tradesvc) trade service	SWIFT business service name (as specified in the XML header).	Maximum: 30 alphanumeric characters

Parameter	Meaning	Value
<code>-trade_sign_dn</code> (<code>-tradesidn</code>) <i>trade sign dn</i>	<i>InterAct only</i> If specified, name used for signing in Initiator mode. Otherwise the securityDN (supplied by SNL) is used	Maximum: 100 alphanumeric characters

Axway Gateway: Managing Transfers

Working with Purge Models (command line)

[`peladm create purge model`](#)

[`peladm update purge model`](#)

[`peladm delete purge model`](#)

[`peldsp display purge model`](#)

[`peldsp select purge model`](#)

Creating a Purge Model: `peladm create_purge_model`

Syntax	<code>peladm create_purge_model {-model} [optional parameters, see list below]</code>
Description	Use this command to create a Model that you can associate with either: - A scheduling-type Decision Rule <code>purge_model</code> parameter - A scheduling-type Rule Table <code>default_purge_model</code> parameter

Parameters*Mandatory*

`-model (-ml)` : Enter a Purge Model name as selection criteria for the purge operation.

`-comments (-cts)` : Enter a free text description of the Purge Model.

Maximum: 80 alphanumeric characters.

`-appli_name (-ap)` : Enter an application name as selection criteria for the purge operation.

Maximum: 25 alphanumeric characters.

`-dest_alias (-da)` : Enter a destination Site alias as selection criteria for the purge operation.

Maximum: 31 alphanumeric characters.

`-direction (-dir)` : Enter a transfer direction as selection criteria for the purge operation.

Enter one of the values:

- **I** = Input
- **O** = Output

`-file_name (-fn)` : Enter a Protocol file name extension.

For FTP and HTTP DIR type Transfers, use this parameter to define a filter for the file names that the server returns as selection criteria for the purge operation.

For example: `*.txt` purges all `.txt` files in the directory that you specify.

`-file_type (-ft)` : *For PeSIT HS transfers only.*

Enter a file type as selection criteria for the purge operation.

Maximum: 16 alphanumeric characters.

`-from_date (-fd)` : Enter the earliest date for the transfers to be selected for the purge operation, in the format *YYYYMMDD HHMMSS*.

`-ident_inf (-id_inf)` : Enter an integer to indicate the minimum value of the transfer identifiers to select for the purge operation.

`-ident_sup (-id_sup)` : Enter an integer to indicate the maximum value of the transfer identifiers to select for the purge operation.

`-keep_last_day (-kld)` : Do not purge transfers from the *ddd* previous days.

Both the transfer `create_date` and `end_date` (if exists) are used for matching. They must be older than the `kld` value. The `Kld` option is not applied as a filter over the "in memory" transfer list but iterates through the mailbox to select the matching transfers. This may impact performance when there is a large mailbox.

Enter an integer value.

`-keep_mailbox_entries (-ke)` : Enter this parameter with Y value to skip purging the transfers from mailbox.

`-force (-f)` : Enter this parameter with Y value to force files removal.

`-list_route_state (-lrs)` : Purge all transfers with the specified routing status:

- **A** = Acked
 - **R** = Routed
 - **F** = Failed
 - **T** = To Route
 - **N** = Not Routed
 - **E** = Empty
 - **P** = Processing
-

`-list_trans_state (-lts)` : Purge all transfers with the specified status:

- **F** = Frozen
- **D** = Delayed
- **B** = To_Begin
- **P** = Processing
- **R** = To_Restart
- **S** = Suspended
- **E** = Ended
- **C** = Canceled
- **I** = Interrupted
- **V** = servicing
- **A** = Acked (OFTP, PeSIT, FTP)
- **T** = Created
- **Z** = Deleted
- **W** = Ended_to_ack
- **G** = To_Sign
- **U** = Nacked_user

`-local_agent (-la)` : Enter an alias of the associated Local Site as selection criteria for the purge operation.
Maximum: 31 alphanumeric characters.

`-local_ident (-i)` : Enter a Local Transfer Identifier as selection criteria for the purge operation.

`-master (-ma)` : Enter a Diffusion List Request identifier as selection criteria for the purge operation.

`-mode (-m)` : Enter a Transfer Connection mode for the Transfer as selection criteria for the purge operation.
Enter one of the values:

- **I** = Initiator
- **R** = Responder

`-no_archive (-no_arc)` : Purge the Transfer from the Mailbox without creating a Transfer archive file.

`-org_alias (-oa)` : Enter an Originator Site alias as selection criteria for the purge operation.
Maximum: 31 alphanumeric characters.

`-protocol (-pr)` : Enter one of the following Protocol values as selection criteria for the purge operation:

- **PEL** (Default)
- **PHSE** = PeSIT HS E
- **PHSD** = PeSIT HS D
- **FTP**
- **HTTP**
- **FTP_HTTP**
- **ODETTE** = OFTP
- **SFTP**
- **SMTP**
- **POP3**
- **TO_GTW**
- **FROM_GTW**
- **AS2**
- **AS3**
- **AS1_POP3**
- **AS1_SMTP**
- **RN_HTTP**
- **RN_POP3**
- **RN_SMTP**

`-purge_file (-pf)` : Enter this parameter with Y value to purge the transferred file.

`-purge_tmp_file (-ptf)` : Enter this parameter with Y value to purge only the temporary file related to the transfers.

`-purge_xpr (-px)` : Enter this parameter with Y value to purge the resolved script file and its output files for the selected transfers.

`-remote agent (-ra)` : Enter the alias of the associated Remote Site as selection criteria for the purge operation.
Maximum: 31 alphanumeric characters.

`-req_name (-rn)` : Enter a name of the purge request to be used in the log file to print statistics.
Maximum: 25 alphanumeric characters (Default = PELPUR).

`-sequence_inf (-seq_inf)` : Enter the lowest number in a range of Transfers to purge.

`-sequence_sup (-seq_sup)` : Enter the highest number in a range of Transfers to purge.

`-to_date (-td)` : Enter the latest date in the range of Transfers to purge. The only transfer parameter that is checked is `create_date`.

Use the date format YYYYMMDD HHMMSS

`-type (-ty)` : Enter one of the following Transfer type values as selection criteria for the purge operation:

- TRANS
 - LIST
 - POLL
 - LOTS
 - EERP
 - SELECT
 - MSG
 - DIR
 - RECEIPT
 - MULTI
-

`-user_state (-usr_st)` : Enter a value for the User state field managed by the user application to be used as selection criteria for the purge operation. (Gateway does not interpret this field in transfer processing.)

Enter a single character.

`-username (-un)` : Enter a user name as selection criteria for the Transfer records to be purged.

Maximum: 31 alphanumeric characters

`-yday_inf (-yd_inf)` : Enter a start date in the format ddd for the range of Transfer records to be purged.

Integer: 001 - 366.

`-yday_sup (yd_sup)` : Enter an end date in the format ddd for the range of Transfer records to be purged.

Integer: 001 - 366.

Modifying a Purge Model: *peladm update_purge_model*

Syntax	<code>peladm update_purge_model {-model} [optional parameters]</code>
Description	Use this command to modify an existing Purge Model. If you update a Model using a parameter without a value, the command deletes the last parameter value. Enter the Purge Model name (<code>-model</code>) followed by all the fields you want to modify.
Parameters	<i>Mandatory</i> <code>-model (-ml)</code> : Enter the name of the existing Purge Model you want to modify. <i>Optional parameters:</i> The optional parameters are the same as in the peladm create_purge_model command above.

Deleting a Purge Model: *peladm delete_purge_model*

Syntax	<code>peladm delete_purge_model {-model}</code>
Description	Use this command to delete a Purge Model.
Parameters	<i>Mandatory</i> <code>-model (-ml)</code> : Enter the name of the existing Purge Model that you want to delete.
Example	The following command deletes the Purge Model named ModelIncoming: <pre>peladm delete_purge_model -ml ModelIncoming</pre>

Selecting Purge Models: *peldsp select_purge_model*

Syntax	<code>peldsp select_purge_model [-model]</code>
Description	Use this command to list all the Purge Models available on your Gateway, or to confirm the presence of a specified Purge Model.
Parameters	<i>Optional</i> <code>-model (-ml)</code> : Enter the name of the existing Purge Model that you want to select. Gateway returns the name of this Purge Model if the Purge Model exists.

Displaying a Purge Model: peldsp display_purge_model

Syntax	<code>peldsp display_purge_model {-model}</code>
Description	Use this command to display the details of a Purge Model identified by its name. The Transfer parameters are displayed by return.
Parameters	<code>-model</code> (<code>-ml</code>) : Enter the name of the existing Purge Model for which you want to display details.

Axway Gateway: Managing Transfers

Monitoring Transfers

This topic introduces the features that enable you to supervise, analyze, and maintain Gateway transfer operations and processes.

- [Mailbox](#)
- [Connections and Connected Users](#)
- [Logs](#)
- [Statistics](#)
- [Automatic maintenance operations](#)
- [Axway Sentinel](#)

Mailbox

To understand how to use the Mailbox in Gateway, it is important that you understand what it is.

The *Mailbox* is a permanent file in which Gateway stores all data relating to Transfer Requests. During the transfer process, all Transfer Request statuses and data are updated in the Mailbox. This stored information enables Gateway to restart and complete transfers in the case of failure during the transfer process.

The Mailbox contains a complete description of the Transfers.

Each Transfer, successful or not, has its own record in the Mailbox. This record stores information such as:

- Transfer origin and destination
- Application used
- Transfer status

- Scheduling parameters:
 - start
 - date
 - priority

To view the Mailbox via online commands, you must display details of Transfer Requests using the `peldsp select trans` command.

Backup Mailbox: Mirroring

The loss of an unduplicated Mailbox file can provoke the loss of all of the collected monitored information about Gateway. The Gateway *Mirroring* feature creates a second version of the Mailbox for backup purposes. This protects the Gateway Mailbox from catastrophic failures such as disk crashes. Refer to [Managing the Mailbox \(command line\)](#).

Connections and Connected Users

To monitor the activities of the Gateway server, you supervise current connections. These are connections that are still open, whether they are active (transferring data) or not.

Gateway can provide you with the following types of information about your current connections:

- Identification
- Localization
- Resource consumption
- Error ratio
- Average transfer speed

To retrieve this information, use *Connection* and *Connected User* objects. Gateway creates these objects for connections and users that use Gateway. The life of the object is limited to the life span of the Connection or Connected User that the object describes. Unlike most other Gateway objects, you cannot create and store Connections and Connected Users. They only display current state information.

To select and display attributes of Connection and Connected User objects, use either the GUI or online commands.

Connections

A Connection object contains information describing a protocol session. Monitored information for connections enables you to detect any exceptional activity (excessive resource consumption, low transfer performance, important error ratios, intrusions, and so on). Use either the GUI or online commands to list current connections, display connection attributes, or stop the progress of a connection (kill the connection).

Connected Users

In the context of monitoring, a Connected User is an entity that is responsible for a set of connections. A Connected User is a virtual object that is identified with the Site used for these connections. All information that single connections carry is cumulated at the Connected User level. As for Connections, use either the GUI or online commands to list and display attributes, and disconnect Connected Users. A disconnection consists in killing all related connections with a single operation.

Logs

Gateway stores traces of all monitoring, protocol, and network events in a *Log* file. You can configure Gateway to write different levels of detail to the file. You consult log contents either using the GUI or by opening the Log file.

Statistics

If you select the corresponding option for your Gateway configuration, each transfer termination (correct ending or cancellation) triggers the deposit of information to a *Statistics file*. You can use the information in your Statistics files to analyze transfer performance.

Automatic maintenance operations

Automatic maintenance operations enable Gateway to run continuously, while at the same time:

- Purging the Mailbox
- Archiving the Log file
- Archiving the Statistics file

Gateway executes automatic maintenance operations periodically according to the definition you set for the date/time cycle, and according to the volume of data that transits the Mailbox. Additionally, you can schedule batch jobs at the end of monitor maintenance processing.

Axway Sentinel

Sentinel is a Business Activity Monitoring (BAM) product, part of the Axway Platform platform.

If your Axway product package includes both Gateway and Sentinel, you can configure Sentinel to monitor Gateway transfer activities.

Monitoring Transfers

This topic introduces the features that enable you to supervise, analyze, and maintain Gateway transfer operations and processes.

- [Mailbox](#)
- [Connections and Connected Users](#)
- [Logs](#)
- [Statistics](#)
- [Automatic maintenance operations](#)
- [Axway Sentinel](#)

Mailbox

To understand how to use the Mailbox in Gateway, it is important that you understand what it is.

The *Mailbox* is a permanent file in which Gateway stores all data relating to Transfer Requests. During the transfer process, all Transfer Request statuses and data are updated in the Mailbox. This stored information enables Gateway to restart and complete transfers in the case of failure during the transfer process.

The Mailbox contains a complete description of the Transfers.

Each Transfer, successful or not, has its own record in the Mailbox. This record stores information such as:

- Transfer origin and destination
- Application used
- Transfer status
- Scheduling parameters:
 - start
 - date
 - priority

To view the Mailbox via online commands, you must display details of Transfer Requests using the [peldsp select trans](#) command.

Backup Mailbox: Mirroring

The loss of an unduplicated Mailbox file can provoke the loss of all of the collected monitored information about Gateway. The Gateway *Mirroring* feature creates a second version of the Mailbox for backup purposes. This protects the Gateway Mailbox from catastrophic failures such as disk crashes. Refer to [Managing the Mailbox \(command line\)](#).

Connections and Connected Users

To monitor the activities of the Gateway server, you supervise current connections. These are connections that are still open, whether they are active (transferring data) or not.

Gateway can provide you with the following types of information about your current connections:

- Identification
- Localization
- Resource consumption
- Error ratio
- Average transfer speed

To retrieve this information, use *Connection* and *Connected User* objects. Gateway creates these objects for connections and users that use Gateway. The life of the object is limited to the life span of the Connection or Connected User that the object describes. Unlike most other Gateway objects, you cannot create and store Connections and Connected Users. They only display current state information.

To select and display attributes of Connection and Connected User objects, use either the GUI or online commands.

Connections

A Connection object contains information describing a protocol session. Monitored information for connections enables you to detect any exceptional activity (excessive resource consumption, low transfer performance, important error ratios, intrusions, and so on). Use either the GUI or online commands to list current connections, display connection attributes, or stop the progress of a connection (kill the connection).

Connected Users

In the context of monitoring, a Connected User is an entity that is responsible for a set of connections. A Connected User is a virtual object that is identified with the Site used for these connections. All information that single connections carry is cumulated at the Connected User level. As for Connections, use either the GUI or online commands to list and display attributes, and disconnect Connected Users. A disconnection consists in killing all related connections with a single operation.

Logs

Gateway stores traces of all monitoring, protocol, and network events in a *Log* file. You can configure Gateway to write different levels of detail to the file. You consult log contents either using the GUI or by opening the Log file.

Statistics

If you select the corresponding option for your Gateway configuration, each transfer termination (correct ending or cancellation) triggers the deposit of information to a *Statistics file*. You can use the information in your Statistics files to analyze transfer performance.

Automatic maintenance operations

Automatic maintenance operations enable Gateway to run continuously, while at the same time:

- Purging the Mailbox
- Archiving the Log file
- Archiving the Statistics file

Gateway executes automatic maintenance operations periodically according to the definition you set for the date/time cycle, and according to the volume of data that transits the Mailbox. Additionally, you can schedule batch jobs at the end of monitor maintenance processing.

Axway Sentinel

Sentinel is a Business Activity Monitoring (BAM) product, part of the Axway Platform platform.

If your Axway product package includes both Gateway and Sentinel, you can configure Sentinel to monitor Gateway transfer activities.

Axway Gateway: Managing Transfers

Viewing and managing Mailbox contents

[About Mailbox contents](#)

[Viewing Mailbox contents](#)

[Managing the Mailbox display](#)

[Displaying Transfer Request record details](#)

[Managing Mailbox contents](#)

[Purging Transfer Request records contained in the Mailbox](#)

About Mailbox contents

You can use the Mailbox display and its associated context menus to view and manage the Mailbox contents.

Each time Gateway processes a Transfer Request it creates a record of the Request and stores it in the Mailbox. These records remain in the Mailbox until you:

- Select and delete records
- Purge the Mailbox
- Reinitialize Gateway using [pelmon init base](#)

Viewing Mailbox contents

To view the Transfer Request records currently stored in the Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** node.

Gateway displays all currently stored Mailbox records in a table in the display pane. By default, Gateway displays a set of columns containing information about each transfer. The table columns contain the following information for each recorded Transfer (if the data exists for that transfer).

Hint: If you do not see the columns containing transfer details in your display, select **View/Details** from the GUI [Menu bar](#).

Column heading	Meaning
Local ident	Local Transfer Identifier: The number attributed by Gateway to identify the transfer. Gateway begins by attributing the number 1, and increments the number by 1 for each successive transfer.
File name	According to the transfer context, this name can be the: • Name of the Application object associated with the transfer • Name of the transferred file • Name of the local Gateway system (VFD) • Physical or logical name of the file on the remote system
Dir name	<i>For HTTP, FTP and SFTP only</i> The absolute path of the directory that contains the file to transfer. The location of the directory depends on the mode: • Initiator mode: a directory on the remote system • Responder mode: a directory in the VFD
Remote agent	Alias of the associated Remote Site

Column heading	Meaning
Originator	Originator protocol identifier
Destination	Destination protocol identifier
Application	The name of the Application object associated with the transfer
Protocol	Protocol supporting the transfer
Type	Transfer type Gateway displays one of the following values: • TRANS • LIST • POLL • LOTS • EERP • SELECT • MSG • DIR • MDN
Mode	Connection mode for the transfer (Initiator/Responder)
Direction	Transfer direction (Send/Receive)
Xfer ident	Protocol identifier of the Transfer. In Initiator mode, the identifier is the local protocol identifier. In Responder mode, the identifier is the partner protocol identifier.

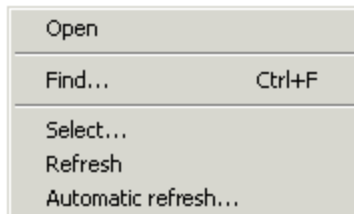
Column heading	Meaning
State	Transfer state One of: • Frozen (F) • Delayed (D) • To_Begin (B) • Processing (P) • To_Restart (R) • Suspended (S) • Ended (E) • Canceled (C) • Interrupted (I) • serVicing (V) • Acked (A) • CreaTed (T) • Deleted (Z) • Ended_to_ack (W) • To_SiGn (G) • NackeD_User (U)
File component	Local name of the transferred file. The default the reception file identity is a seven-character number in the forma: <code>F[local identity number]</code>. For example, the first transfer has the number <code>F000001</code>
Dir path	Local directory path of the transferred file
Size	Size in bytes of the transferred file
Route state	Routing state • TO_ROUTE • ROUTED • FAILED • ACKNOWLEDGED
User state	User state field managed by the user application. Gateway does not interpret this field
Param1	User parameter 1

Column heading	Meaning
Param2	User parameter 2
Model	Model used to submit the Transfer Request
File label	Protocol label of the file, for PeSIT HS only (PI 37)

Managing the Mailbox display

As described in the previous section, when you click the **Transfers** node of the Directory tree structure, you display a list of the Transfer Request records that are contained in the Mailbox.

To manage the contents of the display, right-click anywhere in the display pane (except in the *Local Ident* column). Gateway displays the following context menu:



The following table lists and describes the commands you can use in this context menu to manage the display of Transfer Request records.

Menu command	Description
Open	Displays the view of the Mailbox contents in a separate window. This enables you call up a view of the Mailbox contents while viewing other windows of the GUI. Example: Use this command to generate a separate window display of the Mailbox records, then click the Log icon to display the list of log messages in the display pane. You can now view the Mailbox as needed by selecting the Transfer window name from the drop-down list of the Window menu in the GUI Menu bar.
Find...	Opens the search window, enabling you to search record entries for a given word or number.

Menu command	Description
Select...	Opens <i>Select Transfer</i> window, enabling you to filter the display of Transfer Request Records by one or more criteria. Alternatively, to open this window, in the left pane right-click Transfers and then choose Select from the context menu. Use any combination of the following criteria to generate a filtered list: • Protocol • Direction (Send/Receive) • Mode (Send/Receive) • File name • Application • Model • Originator Site Alias • Destination Site Alias • Originator Site Protocol ID • Destination Site Protocol ID • Transfer ID • Diffusion Request ID • Rank • Permanent transfer • User state • User parameter 1 • User parameter 2 • Creation date and time • Transfer status • Route status
Refresh	Refreshes the display of Transfer Request records, taking into account the most recently processed records.
Automatic refresh...	Opens the Refresh dialog box, enabling you to set an automatic refresh interval in increments of 10 seconds.

Displaying additional Transfer Request information

To obtain additional information about a Transfer Request, double-click the row representing the Transfer Request record in the display pane.

Gateway displays the *Transfer Request Detail* window.

Alternatively, to open this window you can right-click the status icon at the beginning of the row that represents the Transfer Request you want to view. Then select **View** from the context menu that appears.

The window contains several tabs. Each tab displays a specific set of information about the Transfer Request, as summarized in the following table.

Tab name	Contains...
General	General information about the Transfer Request: • Local identifier (identity of the transfer) • Transfer type • Mode (initiator or receiver) • Protocol type • Associated Application object • Group name of Application object • Transfer processing state • acknowledgment option status • Purge option status • Associated transfer model

Tab name	Contains...
Attributes	Information about the transferred file: • File formatting: ◦ Padding ◦ Truncating • Local attributes: ◦ Directory path ◦ File component ◦ Record format ◦ Record length ◦ Data code ◦ Text option status • Transfer attributes: ◦ Transfer identifier ◦ RFD directory option status ◦ Directory name ◦ File name ◦ Record format ◦ Record length ◦ New line convention ◦ Data code ◦ Compression option status
User	Information about the local user: • User name • User state • User processed • End transfer script • User specific parameters 1 and 2 • attached message on send • attached message on receive

Tab name	Contains...
Security	Security-related information: • Local encryption: ◦ File encryption status • TLS: ◦ Security profile name ◦ Cipher suite name ◦ Client authentication option status ◦ Server authentication option status

Tab name	Contains...
Link	Routing characteristics: • Routing ◦ Route state ◦ Route from ◦ Route to ◦ Reply by ◦ Reply to • Diffusion list ◦ Local identifier ◦ Next local ID • Generic reception ◦ Local identifier ◦ Previous local ID ◦ Next local ID ◦ Maximum requests ◦ Maximum files ◦ Total requests ◦ Total files • Permanent ◦ Permanent option status ◦ Duplicated from • Connection parameters ◦ Called address ◦ Called SAP ◦ HTTP Host Name
Sentinel	Sentinel-related information: • Transfer filter

Tab name	Contains...
Statistics	Statistical information about transfer processing: • General data: ◦ Last end reason ◦ Last end diag (last protocol error) ◦ Maximum number of retries ◦ Number of retries ◦ Number of bytes ◦ Number of bytes transferred ◦ File size ◦ Number of records ◦ Allocation size ◦ Creation date ◦ Extraction date • Timing data: ◦ Priority ◦ Earliest start date ◦ Latest start date ◦ Creation date and time ◦ End date and time ◦ Beginning date and time ◦ Day of year (n/365) • Diffusion Lists
[<i>protocol name</i>]	The contents of this tab vary depending on the protocol used for the transfer.

Managing Mailbox contents

In the right (display) pane of the GUI main window, right-click the status icon located at the beginning of the row that represents any Transfer Request. Gateway opens the following context menu.

View...	Ctrl+W
Modify...	Ctrl+U
Delete	Ctrl+Del
Cancel a transfer	
Restart a transfer	
Suspend a transfer	
Ack a transfer	
Expand Diffusion List...	
Associated routed transfer...	
Associated permanent transfer...	
Associated log messages	
Print the selection	
Copy the selection onto the clipboard	

The commands that are available to you depend on the characteristics of the Mailbox record you have selected. In the above example, some commands are deactivated because they do not relate to the Mailbox record of the Transfer Request.

The following table lists and describes the commands you can use in this context menu to manage your Mailbox contents.

Menu command	Description
View...	Opens the <i>Transfer Request Detail</i> window, enabling you to view details of the Transfer Request record.
Modify...	Opens an editing window that enables you to modify the following data related to the Transfer Request record: - General information: - User parameter 1 - User parameter 2 - User state - Purge option - User processed state - Timing parameters - Earliest date and time or relative time - Latest date and time or relative time
Delete	Opens the <i>Delete confirmation</i> window, enabling you to confirm or abandon the deletion of a Transfer Request record.

Menu command	Description
Cancel a transfer	Cancels the transfer (if the selected record represents a transfer in a state that permits cancellation).
Restart a transfer	Restarts the transfer (if the selected record represents a transfer in a state that permits restarting).
Suspend a transfer	Suspends the transfer (if the selected record represents a transfer in a state that permits suspension of processing).
Ack a transfer	Opens the <i>New acknowledgment message</i> window (if the selected record represents a transfer in a state that permits acknowledgment). You can use the <i>New acknowledgment message</i> window to create and send an acknowledgment. Refer to Working with acknowledgment Messages.
Expand Diffusion List...	This command is active in the menu when you right-click the record of a Transfer Request with a Diffusion List selected as destination alias. When you select this command Gateway regenerates the Mailbox display to show the records of transfers to each member of the Diffusion List.
Associated routed transfer...	This command is active in the menu when you right-click the record of a Transfer Request that is linked via a Decision Rule to the deposit of a new Transfer Request. When you select this command, Gateway displays a new window that contains details of the associated Transfer Request.

Menu command	Description
Associated permanent transfer...	This command is active in the menu when you right-click the record of a Transfer Request that was made with the <i>Permanent</i> option active (Permanent = Yes). When Gateway acts as a server, in normal mode (Permanent = No) Gateway makes a specified file available file only to the first client that deposits a Transfer Request for the file. When you activate the Permanent option (Permanent = Yes) Gateway responds to each client Transfer Request by sending a copy of the requested file, keeping the original in the database available for subsequent client Transfer Requests. When you right-click any client Transfer Request record for a file that is referenced by a Permanent server Transfer Request, Gateway displays the original server Transfer Request containing the Permanent option selection. Use this command, for example, when you want to locate and terminate the Transfer Request that makes a given file permanently available to one or more clients.
Associated log messages	When you select this command, Gateway displays log messages of the associated Transfer Request in the Log display pane.
Print the selection	Prints the selected record(s) as displayed in the Mailbox frame.
Copy the selection onto the clipboard	Copies the selected records to the clipboard.

Purging records of Transfer Requests

To purge one or more Transfer Request records from the Gateway Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click **Transfers**, and then select **Purge...** from the context menu.
Gateway opens the *Purge Transfer Requests* window.
3. Complete the **Purge Options** tab. (See table below.)
4. Complete the **Select Transfer** tab. (See table below.)
5. Click **OK** to validate and apply the selection criteria.

Gateway executes the purge based on all criteria entered in both tabs and regenerates the Mailbox display to show the new results.

Purge Transfer Requests: Purge Options tab

Field	Description
Purge Request Name	Enter a name for the Purge request. Gateway uses this name in any log messages concerning the purge operation.
Keep transfers more recent	Enter the number of days that you want to keep transfer records. Gateway deletes all records older than this specified period.
Purge the file sent or received	Select this option to delete the sent or received file as well as the associated record. There is an exception when only the associated record is deleted, and not the files sent or received: when the transfer is routed to Integrator. For more information, refer to Purging for transfers routed from Gateway to Integrator
Purge generated XPR script and output files	For transfers that are associated with XPR scripts that create output files on end of transfer, select this option to purge the generated output files.
Purge only temporary files	Select this option to delete only the temporary files related to the transfers.
Create archive file	Select this option to generate an archive of the deleted Mailbox records.
Keep mailbox entries	Select this option to keep xfer records in mailbox.

Purge Transfer Requests: Select Transfer tab

Field	Description
Protocol	Select a protocol as criteria for the purge operation. To choose all protocols, select (empty) .
Direction	Click to select one or both of the transfer directions: • Send • Receive

Field	Description
Mode	Click to select one or both of the transfer modes: • Initiator • Responder
File Name	Enter the name of a transferred file to use as criteria for the purge operation.
Application	Select the name of an Application to use as criteria for the purge operation.
Model	Select the name of an Application to use as criteria for the purge operation.
Sites Alias:	
Originator	Select an Originating Site alias to use as criteria for the purge operation.
Destination	Select a Destination Site alias to use as criteria for the purge operation.
Sites Protocol Identifier:	
Originator	Select an Originating Site protocol identifier to use as criteria for the purge operation.
Destination	Select a Destination Site protocol identifier to use as criteria for the purge operation.

Field	Description
Creation Date	Use the two date selection boxes to specify a date range for the purge operation. - Beginning Date: Click to select this box, then enter a beginning date limit. To use a calendar to select the beginning date, click the down arrow displayed in the box. Messages created before this limiting date are not selected for the purge operation. - Ending Date box: Click to select this box, then an ending date limit. To use a calendar to select the ending date, click the down arrow displayed in the box. Messages created after this limiting date are not selected for the purge operation. You can use only one of the two boxes to set only a beginning or only an ending date selection criteria. If you enter dates in both boxes, messages with creation dates that fall within this range are selected for purging.
Transfer status	To use a transfer status as message selection criteria, select one or more options from the displayed list. To select records of all terminated transfers in the Mailbox, select all terminated transfers.
Route status	To use a routing status as message selection criteria for the purge operation, click to select one or more statuses from the displayed list. To select records of all routed transfers in the Mailbox, select All.
Transfer ID	Use the two Transfer ID boxes to create a range of ID numbers as selection criteria for the purge operation. Enter the minimum range limit number in the left box, and the maximum range limit number in the right box.
Rank	Use the two Rank boxes to create a range of rank numbers as selection criteria for the purge operation. Enter the minimum rank range number in the left box, and the maximum rank range number in the right box.
User State	Enter a user state as message selection criteria for the purge operation.
Param 1	Enter a User Parameter1 as message selection criteria for the purge operation.

Field	Description
Param 2	Enter a User Parameter2 as message selection criteria for the purge operation.
Permanent Transfer	To select all permanent transfers for the purge operation, select this option.
Diffusion request ID	Enter a Diffusion List type ID number as selection criteria for the purge operation.

Axway Gateway: Managing Transfers

Monitoring Connections

[Viewing all Connections](#)

[Filtering displayed Connections](#)

[Refreshing the Connections display](#)

Viewing all Connections

To view information about the currently open connections on Gateway:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Click the **Connection** node.

Gateway displays all currently open Connections in a table in the display pane. The table columns contain the following information for each Connection.

Column	Contents
ID	Connection identifier.
State	Status of the connections: • T = Transferring • S = Still
Connected user	The Connected User responsible for the Connection.
Speed	Average transfer speed for this Connection. Only significant transfers (more than one second) are taken into account.
Data	Cumulated transferred data volume for this Connection.
Downloads	Number of files downloaded during this Connection.
Uploads	Number of files uploaded during this Connection.

Filtering displayed Connections

To reduce the number of Connections presented in the display pane, or to obtain for a particular display result, you can apply filtering criteria to the display of Connections:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Right-click the **Connection** node.

Gateway displays the *Connection* context menu.

3. In the context menu, click **Select...**

Gateway displays the *Select Connections* dialog box.

Use one or more of the fields in the *Select Connections* dialog box to create selection criteria for the display of Connections:

Field	Use
Connection ID	Enter a connection ID number to select a specific Connection for display
Connection time	
Start	You can select either starting time or starting date criteria, or combine the two. You can combine starting time and/or starting date criteria with ending time and/or ending date criteria. To set a starting date as selection criteria: 1. Click the down arrow on the starting date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the start month and start date of your choice. To set a starting time as selection criteria: 1. Click the check box next to the starting time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

Field	Use
End	You can set either ending time or ending date criteria, or combine the two. You can combine ending time and/or ending date criteria with starting time and/or starting date criteria. To set an ending date as selection criteria: 1. Click the down arrow on the ending date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the end month and end date of your choice. To set an ending time as selection criteria: 1. Click the check box next to the ending time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.
State	To select a Connection state as display selection criteria: 1. Click the down arrow located in the State field. 2. Select the state to use as display selection criteria: • Inactive • Transferring • Both
Mode	To select a Site mode as display selection criteria: 1. Click the down arrow located in the Mode field. 2. Select the Site mode you want to use as display selection criteria: • Initiator • Responder • Both

4. Click **OK**.

Gateway applies your criteria and regenerates the display results.

Refreshing the Connections display

To refresh the display of Connections to take into account the most recent network events, do one of the following:

- In the left pane of the GUI main window, click  to expand the nodes **Transfer Management > Monitoring**. Then right-click **Connection**. In the displayed context menu, select **Refresh**.
- Press **F5**.

[Monitoring Connections \(command line\)](#)

[Monitoring Transfers](#)

Axway Gateway: Managing Transfers

Monitoring Connected Users

[Viewing all Connected Users](#)

[Filtering displayed Connected Users](#)

[Refreshing the Connected Users display](#)

Viewing all Connected Users

To view information about the user entities that are currently using connections on the Gateway:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Click the **Connected User** node.

Gateway displays all Connected Users in a table in the display pane. The table columns contain the following information for each Connected User.

Column	Contents
Connected user name	Identity of the Connected User.
Upload size used	Percentage of upload data size used within the user quota. This parameter is related to FTP defense against hammer attacks.

Column	Contents
Download size used	Percentage of download data size used within the user quota. This parameter is related to FTP defense against hammer attacks.
Upload requests used	Number of upload requests used within the user quota. This parameter is related to FTP defense against hammer attacks.
Download requests used	Number of download requests used within the user quota. This parameter is related to FTP defense against hammer attacks.
Command credits used	Percentage of command credit used. This parameter is related to FTP defense against hammer attacks.
Connections count	Number of currently active (transferring data) Connections.
Max connections	Maximum number of Connections allowed for this Connected User.
Data count	Cumulated transferred data volume for this Connected User.
Uploads count	Number of uploaded files for this Connected User.
Downloads count	Number of downloaded files for this Connected User.
Last remote address	Outgoing network address of the last Connection.
Speed	Average transfer speed for this Connected User. Only takes into account significant transfers (more than one second).
Errors count	Number of transfer errors for this Connected User.
Last connection date	Date and time the last Connection started. Format <i>YYMMDD HHMMSS</i>

Filtering displayed Connected Users

To reduce the number of Connected Users listed in the display pane, or to obtain for a particular display result, you can apply filtering criteria to the display of Connected Users.

To select filter criteria for the display of Connected Users:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Right-click the **Connected User** node.

Gateway displays the *Connected User* context menu

3. In the context menu, click **Select...**

Gateway displays the *Select Connected Users* dialog box.

Use the fields of the *Select Connected Users* dialog box to create selection criteria for the display of Connections. You can use one or more of the fields described in the following table.

Field	Use
Connected user	Enter the identification number of the Connected User to display.
First connection time	
Start	You can set either starting time or starting date criteria to filter results. You can also combine the starting date and starting time as criteria. You can combine starting time and/or starting date criteria with ending time and/or ending date criteria. To set a starting date as selection criteria: 1. Click the down arrow on the starting date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the start month and start date of your choice. To set a starting time as selection criteria: 1. Click the check box next to the starting time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

Field	Use
End	You can set either ending time or ending date criteria, or combine the two. You can combine ending time and/or ending date criteria with starting time and/or starting date criteria. To set an ending date as selection criteria: 1. Click the down arrow on the ending date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the end month and end date of your choice. To set an ending time as selection criteria: 1. Click the check box next to the ending time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

4. When you have set the criteria that you want to use to filter the display results, click **OK**.

Gateway applies your criteria to regenerate the display results.

Refreshing the Connected Users display

To refresh the display of Connected Users to take into account the most recent network events, do one of the following:

- In the left pane of the GUI main window, click  to expand the nodes **Transfer Management > Monitoring**. Then right-click **Connected Users**. In the displayed context menu, select **Refresh**.
- In the left pane of the GUI main window, click  to expand the nodes **Transfer Management > Monitoring**. Click **Connected Users**. Then press **F5**.

[Monitoring Connected Users \(command line\)](#)

[Monitoring Transfers](#)

Automating maintenance operations

[About Automating maintenance operations](#)

[Setting up automatic Log file archiving](#)

[Setting up automatic Statistic file archiving](#)

About Automating maintenance operations

You can use automatic maintenance to enable Gateway to run without interruption while archiving production data in Log or Statistics files. You use date/time periodic cycle and volumetric criteria to program the frequency of automatic maintenance tasks.

Setting up automatic Log file archiving

About automatic Log file archiving

You can configure automatic archiving of the log file depending on one of the following criteria that you define in the configuration menu:

- The date/time periodic cycle
- The maximum log file size in Kbytes, defined via the configuration parameter **Threshold size** (`auto_arc_bytes`)

To create an archive, Gateway uses three files:

- File in use
- First archive file
- Second archive file

To archive a log file, Gateway:

1. Renames the file in use to either archive file one or archive file two.
2. Creates and opens a new, empty log file.
3. Schedules the batch job defined by the configuration parameter `log_arc_cmdline`. This parameter calls the script `arclog.sh` (UNIX) or `arclog.bat` (Windows). As the first argument to this batch job, it passes the system pathname of the archived file. If this parameter is not set, the `ExitLogArchived` exit routine (synchronous routine) is called instead.

How to set up automatic Log file archiving

To set up automatic archiving of Log files:

1. In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.

2. Select **Gateway > Database > Archive**.
3. Complete the [Log archive parameters](#):
 - Select **Active**
 - Complete **Start date** and **Period**, or complete **Threshold size**If you specify both the **Start date/Period** and the **Threshold size**, the **Start date/Period** is ignored and the **Threshold size** is used.
4. Click **OK**.

Setting up automatic Statistics file archiving

The statistics file follows the same rules as the log file.

The configuration parameter that schedules the batch job is `stat_arc_cmdline` rather than `log_arc_cmdline`. This parameter calls the script `arcstat.sh` (UNIX) or `arcstat.bat` (Windows).

The exit routine used is `ExitStatArchived` instead of `ExitLogArchived`.

Axway Gateway: Managing Transfers

Viewing Mailbox contents (command line)

The Mailbox does not have its own set of online commands.

To view the Mailbox via online commands, use the [peldsp select trans](#) command to display details of Transfer Requests.

Axway Gateway: Managing Transfers

Viewing Mailbox contents (command line)

The Mailbox does not have its own set of online commands.

To view the Mailbox via online commands, use the [peldsp select trans](#) command to display details of Transfer Requests.

Axway Gateway: Managing Transfers

Managing the Mailbox (command line)

[About Mailbox management commands](#)

[Initializing the Mailbox](#)

[Optimizing the Mailbox](#)

[Managing the Mailbox](#)

[Backing up the Mailbox \(mirroring\)](#)

About Mailbox management commands

This topic contains information to help you manage the Mailbox in your daily operations. It includes descriptions of the processes and commands that are useful to operate and protect your Mailbox.

Gateway Mailbox management commands are designed for use by experienced users. While these commands are powerful tools, they can also be quite complex.

Initializing the Mailbox

You do not have to initialize this version of Gateway. The Installer executes the `gateinit` command for you. The command still exists in this version of Gateway for reasons of compatibility. This command, among other things, creates the Mailbox file. Subsequently, when you start Gateway, the software either automatically creates the Mailbox, or re-opens it, as required.

Alternatively, to initialize the Mailbox you can use the [pelmon init base](#) command.

Optimizing the Mailbox

Gateway includes the *Key File Management Copy* (`kfmcp`) tool. This tool enables you to [maintain and optimize the Gateway Mailbox files](#) (`xfer.dat` and `xferb.dat`). They are located in `<Gateway installation directory>/run_time/data`. These files contain status information for current file Transfer Requests.

Managing the Mailbox

After you initialize the Mailbox for the first time and then optimize the Mailbox, you can use the `pelmon` command to perform Mailbox management operations:

Command	Operation
pelmon init base	Initialize Gateway and the Mailbox
pelmon restart	Restart Gateway and activate links to the Mailbox
pelmon check base	Check the integrity of Gateway and the Mailbox
pelmon recover base	Recover a corrupted Mailbox
pelmon change pass phrase	Change the Mailbox database ciphering key

Command	Operation
<code>pelmon rebuild vfd</code>	Repair the Virtual File Directory
<code>pelmon reset</code>	Reset the status prompted by the <code>gatestatus</code> command to the default value

Deprecated pelmon subcommands

Do not use the following `pelmon` subcommands that were used in previous versions of Gateway. Only use the replacement commands:

- `pelmon start`
Replaced with the [`gatestart`](#) command.
- `pelmon stop`
Replaced with the [`gatestop`](#) command.
- `pelmon status`
Replaced with the [`gatestatus`](#) command.

pelmon init_base

Syntax	<code>pelmon init_base [-force (-f)] [-cipher_passphrase_dkfile (-cpdkf)] [-cipher_passphrase_encfile (-cpencf)]</code>
Description	Use this command to initialize Gateway, and activate links to the Mailbox. You can use this command to restart Gateway after a system crash that caused integrity problems on both Mailboxes. If the Mailbox file does not already exist, this command creates it. If the Mailbox file already exists, this command truncates the file, deleting all transfers from the Mailbox and resetting the transfer identifier to 1. If the Gateway Mirror option is activated, this command initializes both Mailboxes (master and mirror).
Alternative command	<code>gateinit</code>

Parameters

`-force (-f)` : Use this parameter to force Mailbox initialization when there is no integrity problem on the Mailbox. Additionally, this parameter removes all VFD Items without modifying the directory organization.

`-cipher_passphrase_dkfile (-cpdkf)` : Enter a path to a derived key used to encrypt the protection password. Maximum: 256 alphanumeric characters.

Use this parameter if AES encoding protects Gateway files. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file.

If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

`-cipher_passphrase_encfile (-cpencf)` : Enter a path to an encrypted password file. Use this parameter if encoding protects the Mailbox and its associated object database files. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file. If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

Example

Before initializing the Mailbox, stop Gateway.

Enter the command:

pelmon init_base

Gateway returns:

```
Testing Axway Gateway status: stopped
Checking software license key: xxxx xxxx xxxx
xxxx xxxx
Expiration date :none
Max network connections :unlimited
      Max site :unlimited
      Protocol options :PSIT FTP
      Misc product options :'Secured
      Mailbox'
Checking monitor option versus license:
Mirror option enable
Checking monitor context:
      Monitor stopped normally
Initializing master mailbox: done
Initializing mirror mailbox: done
```

pelmon restart

Syntax	pelmon restart [-trace_level (-1)] [-cipher_passphrase_dkfile (-cpdkf)] [-cipher_passphrase_encfile (-cpencf)] [-wait]
Description	Use this command to restart Gateway, and by extension, to restart the tracking functions of the Mailbox. Use this command on Gateway when it is running. This command does not clean the current contents of the Mailbox.

Parameters

`-trace_level (-l)` : Starts the Mailbox trace in binary format set to the specified level. Enter one of the following values:

- 0 - not activated
Use this parameter value for debugging purposes only, not for production
- 1
- 2
- 3
- 4 maximum trace information

`-cipher_passphrase_dkfile (-cpdkf)` : Enter a path to a derived key used to encrypt the protection password. Maximum: 256 alphanumeric characters.

Use this parameter if encoding protects the Mailbox and its associated object database files. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file.

If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

`-cipher_passphrase_encfile (-cpencf)` : Enter a path to an encrypted password file. Use this parameter if encoding protects the Mailbox and its associated object database files. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file. If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

`-wait` : Use this parameter if you want to check that Gateway starts up correctly.

`-no_xsr` : Use this parameter to start Gateway without Secure Relay.

`-auto_check (-auto)` : Y (Yes) or N (No) Enter Yes to enable automatic checking functions on restart. `auto_check` corrects certain file content problems, for example it confirms and corrects VFD file structures.

pelmon check_base

Syntax	<code>pelmon check_base [-recover_auto (-ra)] [-cipher_passphrase_dkfile (-cpdkf)] [-cipher_passphrase_encfile (-cpencf)]</code>
Description	Use this command to check Mailbox integrity. If the Gateway Mirror option is activated, the command checks both Mailboxes. This command is mandatory if: • Gateway stopped abnormally • a problem occurred when you started Gateway Stop Gateway before you check the Mailbox.
Parameters	<code>-recover_auto (-ra)</code>: automatically recovers a corrupted file (master or mirror Mailbox), without using any other command. <code>-cipher_passphrase_dkfile (-cpdkf)</code>: Enter a path to a derived key used to encrypt the protection password. Maximum: 256 alphanumeric characters. Use this parameter if Gateway files are encrypted. If you specify a path, Gateway automatically tries to get the password phrase required to open the Mailbox file from the specified user file. If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase. <code>-cipher_passphrase_encfile (-cpencf)</code>: Enter a path to an encrypted password file. Use this parameter if encoding protects the Mailbox and its associated object database files. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file. If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

Example

Enter the command:

```
pelmon check_base -recover_auto
```

Gateway returns:

Testing Axway Gateway status: stopped

Checking software license key : xxxx xxxx xxxx
xxxx xxxx

Expiration date :none

Max network connections:unlimited

Max site :unlimited

Protocol options :PEL FTP

Misc product options : 'Secured
Mailbox'

Checking monitor option versus license:

Mirror option enable

Checking monitor context:

Monitor stopped normally

Last checking ended normally

Checking base integrity:

Master mailbox : corrupted

Mirror mailbox : OK

Recovering of master mailbox: Done

Automatic recovering done.

pelmon recover_base

Syntax

```
pelmon recover_base [-force (-f)] [-how (-h)]  
[-cipher_passphrase_dkfile (-cpdkf)]
```

Description

Use this command to recover a corrupted Mailbox.
Stop Gateway before you execute this command.

Parameters

`-force (-f)` : Forces the recovery of a Mailbox that is not detected as corrupted.

`-how (-h)` : Enables you to manually select the Mailbox (master or mirror) for recovery.

Select one of the following values:

- `auto` - (default) automatically determines which Mailbox to recover
 - `master` - recovers master Mailbox
 - `mirror` - recovers mirror Mailbox
-

`-cipher_passphrase_dkfile (-cpdkf)` : Enter a path to a derived key used to encrypt the protection password. Maximum: 256 alphanumeric characters.

Use this parameter if Gateway files are encrypted. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file.

If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

`-cipher_passphrase_encfile (-cpencf)` : Enter a path to an encrypted password file. Use this parameter if encoding protects the Mailbox and its associated object database files. If you provide a path, Gateway automatically tries to access the password phrase required to open the Mailbox file from the specified user file. If you do not specify a value for this parameter, Gateway dynamically prompts you for the passphrase.

Example

Before initializing the Mailbox, stop Gateway.

Enter the command:

```
pelmon recover_base
```

Gateway returns:

```
Testing Axway Gateway status: stopped
Checking software license key: xxxx xxxx xxxx
xxxx xxxx
Expiration date :none
Max network connections :unlimited
Max site :unlimited
Protocol options :PSIT FTP
Misc product options :'Secured Mailbox'
Checking monitor option versus license:
    Mirror option enable
Checking monitor context:
    Monitor stopped normally
    Last checking ended normally
Checking base integrity:
    Master mailbox : corrupted
    Mirror mailbox : OK
Recovering of master mailbox: Done
```

pelmon change_pass_phrase

Syntax

```
pelmon change_pass_phrase [-old_cipher_
passphrase_dkfile (-ocpdkf)] [-old_cipher_
passphrase_encfile (-ocpencf)] [-new_cipher_
passphrase_dkfile (-ncpdkf)]
```

Description

Use this command to change the key used to encrypt the Gateway database files and to perform the encryption. The key is derived from the passphrase that you enter using this command.

For security purposes, you should regularly change the key.

If you later want to unencrypt the database files, specify the current passphrase in `-ocpdkf` and **do not** specify a new passphrase in `-ncpdkf`.

Prerequisites

Before you change the passphrase:

1. In the configuration menu, specify the [Database protection option](#).
2. [Reload the profile](#).
3. Stop Gateway.
4. Back up all the database files (all `.dat` files present in `<Gateway installation directory>/run_time/data`).

Parameters

`-old_cipher_passphrase_dkfile (-ocpdkf)`: Enter the path to the file that contains the derived key for encrypting old passphrase for the key used to encrypt the file.

If you do not specify a value, Gateway will prompt you for the old password. Maximum: 80 alphanumeric characters.

`-old_cipher_passphrase_encfile (-ocpencf)`: Enter the path to the file that contains the old encrypted passphrase for the key used to encrypt the file.

If you do not specify a value, Gateway will prompt you for the old password. Maximum: 80 alphanumeric characters.

`-new_cipher_passphrase_dkfile (-ncpdkf)`: Enter the path to the file that contains the derived key for encrypting new passphrase for the key used to encrypt the file.

If you do not specify a value, Gateway will prompt you for the new password.

If the database files are already encrypted and you do not specify a value, the command will unencrypt the files.

Maximum: 80 alphanumeric characters.

`-new_cipher_passphrase_encfile (-ncpencf)`: Enter the path to the file that contains the encrypted new passphrase for the key used to encrypt the file.

If you do not specify a value, Gateway will prompt you for the new password.

If the database files are already encrypted and you do not specify a value, the command will unencrypt the files.

Maximum: 80 alphanumeric characters.

Result

Gateway encrypts the database files using the newly-generated key and creates new files with the extension `.new`. The files that Gateway encrypts depend on the value specified in the **Database protection option** in the configuration menu.

If the operation is successful:

Gateway renames the previous files with the extension `.old` and the new ones with the extension `.dat`. Gateway then uses the new files.

Important: Back up the old files and remove them from the Gateway database directory. If you do not do this, the next **pelmon change_pass_phrase** command that you enter will not be processed. This is to prevent the old files being overwritten.

If the operation fails:

If an error occurs, the `change_pass_phrase` operation is immediately aborted and the files are not renamed.

In this case you can do one of the following:

- Correct the problem, then restart the `change_pass_phrase` operation
- Delete the `.new` files and restore the old files with the previous encryption parameters

Example

Enter the command:

```
pelmon change_pass_phrase
```

Gateway returns:

```
Enter your OLD pass-phrase
```

```
Enter your NEW pass-phrase
```

```
Re-enter your NEW pass-phrase
```

pelmon rebuild_vfd**Syntax**

```
pelmon rebuild_vfd
```

Description	Use this command to repair the Virtual File Directory (VFD) from the backup. Stop Gateway before you rebuild the VFD. To stop Gateway use the gatestop command.
Parameters	None
Example	Enter the command: pelmon rebuild_vfd Gateway returns: Testing Gateway status : stopped. Recovering VFD database... Importing VFD database from text file "d:\work_area\gw6114head\install_dir\run_time\arc\vfdexport.txt"...

pelmon reset

Syntax	pelmon reset
Description	Use this command to reset the status prompted by the gatestatus command to the default value.
Parameters	None
Example	Enter the command: pelmon reset Gateway returns: Printing Gateway status : running.

Backing up the Mailbox (mirroring)**About mirroring**

The Mailbox is the file transfer database. It is one of the most important files in Gateway. If you lose this file, you lose all the monitored information about Gateway. Gateway mirroring is an optional feature that requires no specific hardware except a second physical disk device. Mirroring protects the Gateway Mailbox from a software or hardware crash. Mirroring is submitted to a software license key.

When you activate the [Mirror option](#) in the configuration menu, Gateway maintains two Mailbox files instead of a single file:

- Master Mailbox
- Mirror Mailbox

Each operation that you perform on the master Mailbox is duplicated on the mirror Mailbox, except for the read operation.

Note: The master Mailbox file is the same as the Mailbox file used when you do not activate the mirror option.

Locate the mirror Mailbox file on a different physical disk from that of the master Mailbox file, to protect the Mailbox from a hardware crash. The disk that supports the mirror Mailbox file should have the same availability as the disk that stores the master Mailbox. When you install Gateway, you specify the location of the mirror Mailbox.

Error detection

When you try to recover a Mailbox error, determine the nature of the error, as well as which Mailbox contains the error (master or mirror).

Three files are used to maintain Mailbox integrity:

- Master Mailbox file - checked for integrity
- Mirror Mailbox file - checked for integrity
- Context file - indicates whether a crash occurred.

The *context file* (typically a file named `supctx.dat` in the `run_time/data` directory) is a text file that contains:

- Monitor run-time information
- Two fields for managing database integrity:
 - `sup_state = 0` (Monitor is stopped), `1` (Monitor is running or was stopped abnormally)
 - `chk_state = 0` (Error found in database), `1` (Database integrity is granted)

Caution: You can browse the context file for information. However, you must **never modify** field values with an editor. If you edit the file, you will damage Mailbox integrity.

Since errors may have many different origins and many different results, Gateway always checks the integrity of both Mailboxes before startup. It performs this integrity check via the `pelmon check_base` command. This command stores any errors it detects in the Gateway context file. You must correct these errors before you can start Gateway.

When you start Gateway, it writes its running state in the context file. If the system crashes, Gateway can then detect that it did not stop normally. To restart Gateway, check Mailbox integrity manually.

General guidelines for using mirroring

Stop Gateway before you perform any Mailbox maintenance operations.

When an integrity problem occurs, perform all the possible checks (using the [pelmon](#) command, or platform-specific tools) before attempting any recovery.

Important: Recovery is an irreversible operation. Backup both Mailbox files before attempting a recovery.

[Maintaining the Mailbox \(kfmcp command\)](#)

[About gatestatus messages](#)

[Configuration: Gateway parameters](#)

Axway Gateway: Managing Transfers

Maintaining the Mailbox (kfmcp command)

[KFM-based data files](#)

[kfmcp command](#)

[kfmcp command usage](#)

[Gateway KFM-based files](#)

[Resizing the Mailbox](#)

KFM-based data files

Some Gateway data files use a proprietary structure that offers the following access features:

- Fast record retrieval by key value
- Fast insertion/deletion
- Concurrent (reader/writer) access
- Record locking
- Integrity checking
- File space pre-allocation
- Free space reuse

The software that implements this file structure and the associated access method is called *KFM (Key File Management)* software.

The metadata that allows fast retrieval by key value is based on the hashcoding technique.

kfmcp command

The `kfmcp` utility supplied with the product enables you to manage KFM-based files.

Each KFM-based data file has the following set of attributes that define its structure:

Attribute	Meaning

<code>-version</code>	A number that informs KFM about the version of the file structure. Only version numbers 11 and 12 are used in the current version of Gateway. Most of the KFM-based files use version 11. KFM version 12 provides minor optimizations over version 11.
<code>-maxr</code>	Corresponds to the <i>modulo</i> of the hashcode. This is related to the maximum number of records without hash collisions. The maximum supported value is 262 144 (2^{18}). The actual number of records in a KFM file is not limited to this value. An average value of no more than 4 hash collisions should not significantly affect performance for direct accesses by key value.
<code>-alsz</code>	Initial file allocation table. The maximum value supported is 16 Gb. Any value greater than this limit is truncated to 16Gb.
<code>-alsz2</code>	The size in bytes used for secondary allocations. Each time KFM allocates more space for a file, it uses this value for the size of the allocation. This value must be distinguished with the initial file space allocation that can be achieved using the <code>kfmcp</code> command. The maximum value supported is 256Mb. Any value greater than this limit is truncated to 256Mb.
<code>-hhint,</code> <code>hp1, hp2</code>	Hints used for hashcode optimization.

Additionally, some other hidden attributes are used for ciphered files.

kfmcp command usage

You can use the `kfmcp` command for various tasks:

usage 1

```
kfmcp -t [infile1 infile2 ...]
```

Counts the total number of records of the files and prints the result.

usage 2

```
kfmcp -M maxr alsz_kb alsz2_kb [infile1 infile2 ...] outfile
```

Creates a new file envelope in `outfile` and copies all the records from `infile`s to `outfile`.

`maxr` = number of hash classes:

- Rounded down to the greatest power of 2 $\leq \text{maxr}$
- Maximum value $262\,144 = 2^{18}$

`alsz_kb`: primary (initial) file allocation in KB

`alsz2_kb`: secondary file space allocation in KB

Other attributes (`version`, `hhint`, `hp0`, `hp1`, cipher-related ones) are extracted from `infile1` and applied to `outfile`.

usage 3

```
kfmcp -X version maxr alsz_b alsz2_b [infile1 infile2 ...]  
outfile
```

Same as option `-M` except that:

- The version of `outfile` can be changed
- Allocation quantities are in bytes instead of kilobytes (KB) and limited to 2 Gigabytes.

usage 4

```
kfmcp -S [infile1 infile2 ...]
```

Prints attributes of `infile`s to the standard output. For example:

```
kfmcp -S xfer.dat
```

```
file=xfer.dat
```

```
version=11
```

```
maxr=262144
```

```
alsz2=100000
```

```
hhint=1
```

```
hp1=0
```

```
hp2=4
```

```
records=6
```

usage 5

```
kfmcp -d [infile1 infile2 ...]
```

Dumps the internal structure of the file to the standard output. Can be used for low-level debugging.

usage 6

```
kfmcp -R [infile1 infile2 ...] [outfile]
```

Creates a new file envelope in `outfile` and copies the records from `infile`s ... to `outfile`, while validating each input record. Invalid records will be skipped. This command can be used as a means to "repair" a mailbox file. If only one file is specified in the parameter list, it will be considered an `infile` and the command will verify its consistency. Only a partial check is performed, enough to create a problem-free copy of the `infile`s, but not to detect all possible consistency errors.

Gateway KFM-based files

On its first startup, Gateway creates the following KFM-based files:

File Name	KFM version	maxr	alsz2_b	alsz_b
admin.dat	11	32 768	500 000	500 000
agenda.dat	11	32 768	500 000	500 000
cert.dat	11	32 768	500 000	500 000
cgate.dat	11	32 768	500 000	500 000
cgategroup.dat	11	32 768	500 000	500 000
dlist.dat	11	32 768	500 000	500 000
key.dat	11	32 768	500 000	500 000
model.dat	11	32 768	500 000	500 000
nprof.dat	11	32 768	500 000	500 000
pmindex.dat	12	1024	100 000	100 000
profile.dat	11	32 768	500 000	500 000
proxy.dat	11	32 768	500 000	5 000 000
rules.dat	11	32 768	500 000	5 000 000
sgateb.dat	12	262 144	5 000 000	5 000 000
sgate.dat	12	262 144	5 000 000	5 000 000
sgategroupb.dat	12	16 384	5 000 000	5 000 000
sgategroup.dat	12	16 384	5 000 000	5 000 000
sprof.dat	11	32 768	500 000	500 000
sshprof.dat	11	32 768	500 000	500 000
trade.dat	11	32 768	500 000	500 000
user.dat	11	32 768	500 000	500 000
vsite03.dat	11	262 144	2 000 000	2 000 000
xferb.dat	11	262 144	5 000 000	5 000 000
xfer.dat	11	262 144	5 000 000	5 000 000

Mailbox files

Most of the Gateway KFM-based files contain configuration objects, such as Sites, CGates, Applications, Decision Rules, and so on. The `xfer.dat` and `xferb.dat` files contain transfer metadata items that are accessed very extensively by Gateway. These two files are collectively known as the *Transfer Mailbox* or simply the *Mailbox* in Gateway terminology.

Resizing the Mailbox

Although Gateway can allocate KFM file space when needed, allocating the whole Mailbox disk space needed for a given production workload can improve performance and protects against errors due to file allocation failures.

You can use the following guidelines to estimate the maximum size of the two Mailbox files:

- The file `xfer.dat` contains 1 fixed-size record for each Transfer Request. The size of each record is about 11 KB.
- The file `xferb.dat` also contains fixed size records. Each record is 3.5 KB. The use of an `xferb.dat` record for a Transfer Request depends on the protocol. For example, PESIT and PEL protocols do not need a record in `xferb.dat`. However, all EDI protocols require such a record for each transfer. As a general rule, used in the examples below, you can assume that the number of records in both `xfer.dat` and `xferb.dat` are the same.
- For a large number of records, the space of KFM files that is not used for data records (mostly hash technique related structure) is very small (less than 5% of the total file space).

Given these simple rules, assuming that each record requires 12 KB in `xfer.dat` and 4 KB in `xferb.dat`, we can estimate the disk space for these files.

Use case for 100 000 Transfer Requests

The preferred `kfmcp` command to use to resize the Mailbox is `kfmcp -M ...`.

File space for `xfer.dat`: $100\,000 \times 12\text{ KB} = 1\,200\,000\text{ KB}$

File space for `xferb.dat`: $100\,000 \times 4\text{ KB} = 400\,000\text{ KB}$

These values correspond to the `alsz_kb` argument of the `kfmcp -M` command.

The secondary allocation `alsz2_kb` is only useful if new space has to be allocated on need. It is not used if the full needed space is pre-allocated. Nevertheless, it is not recommended to specify a very small value for `alsz2_kb`.

In the following examples, we use the greater value between 5000 KB and 1% of the total file space, with an absolute maximum of 50 000 KB.

Example 1

```
kfmcp -M 131072 1200000 12000 xfer.dat xfer_new.dat
```

This example uses 131 072 hash classes for 100 000 transfers (reasonable).

Example 2

```
kfmcp -M 262144 1200000 12000 xfer.dat xfer_new.dat
```

This example uses twice the number of needed classes (rather high).

Example 3

```
kfmcp M 0 1200000 12000 xfer.dat xfer_new.dat
```

This example provides a value of 0 for `maxr` which has the effect of reusing the value from `xfer.dat` which is 262 144.

Given the low overhead of KFM space due to hashcoding related data, we recommend that you do not change the `maxr` value (262 144), even in situations where this value may seem too large in theory.

The following tables provide suitable values for `alsz_kb` and `alsz2_kb` depending on the number of Transfer Requests.

Values to use for *xfer.dat*

Number of transfers	alsz_kb	alsz2_kb
100 000	1 200 000	12 000
200 000	2 400 000	24 000
300 000	3 600 000	36 000
500 000	6 000 000	50 000
1 000 000	12 000 000	50 000

Values to use for *xferb.dat*

Number of transfers	alsz_kb	alsz2_kb
100 000	400 000	5 000
200 000	800 000	8 000
300 000	1 200 000	12 000
500 000	2 000 000	20 000
1 000 000	4 000 000	40 000

Use case for 300 000 Transfer Requests on a UNIX system

```
kfmcp -M 0 3600000 36000 xfer.dat xfer_new.dat
```



```

kfmcp -M 0 1200000 12000 xferb.dat xferb_new.dat

mv xfer.dat xfer_old.dat

mv xfer_new.dat xfer.dat

mv xferb.dat xferb_old.dat

mv xferb_new.dat xferb.dat

```

Axway Gateway: Managing Transfers

Monitoring Connections (command line)

[peldsp select connection](#)

[peldsp status connection](#)

[peldsp display connection](#)

[pelctl kill connection](#)

Displaying list of Connections: *peldsp select_connection*

Syntax	<code>peldsp select_connection [(-ci) (-ca) (-cb) (-s) (-m)]</code>
Description	Use this command to display the list of Connections that match the parameters that you specify. The Connections may be in a transfer phase (Transferring) or not (Still). To display the list of all Connections, enter the command with no parameter. Alternatively, specify the connection via the Connection identifier parameter. Use the optional parameters to further refine the search.
Parameters	<code>-connection_id (-ci)</code> : Enter an integer that uniquely identifies the connection.
	<code>-connected_after (-ca)</code> : Enter a time and date in the format <code>YYMMDD HHMMSS</code> .
	<code>-connected_before (-cb)</code> : Enter a time and date in the format <code>YYMMDD HHMMSS</code> .
	<code>-state (-s)</code> : Enter one of the following values: • S = Still (no activity) • T = Transferring
	<code>-mode (-m)</code> : Enter one of the following values: • I = Initiator • R = Responder

Example	<pre>peldsp select_connection -ca '070803 150000'</pre> This command selects the connections that occurred after 3 p.m. (150000) on 3 August 2007 (070803).

Displaying Connection status: *peldsp status_connection*

Syntax	<code>peldsp status_connection [(-ci) (-ca) (-cb) (-s) (-m)]</code>																
Description	Use this command to display status information for a list of active Connections that match the parameters that you specify. Each element of this list comprises a Connection identifier followed by status attributes. The Connections are either in a transfer phase (Transferring) or not (Still). To list all Connections, use the command with no parameters. Alternatively, refine the search using optional parameters.																
Parameters	<code>-connection_id (-ci)</code>: Enter an integer that uniquely identifies the connection. <code>-connected_after (-ca)</code>: Enter a time and date in the format <code>YYMMDD HHMMSS</code>. <code>-connected_before (-cb)</code>: Enter a time and date in the format <code>YYMMDD HHMMSS</code>. <code>-state (-s)</code>: Enter one of the following values: S = Still (no activity) T = Transferring <code>-mode (-m)</code>: Enter one of the following values: I = Initiator R = Responder																
Example	<pre>peldsp status_connection -m I</pre> This command selects Initiator mode connections. <table><tr><td>Ident</td><td>Sta</td><td>Address</td><td>Connected At</td></tr><tr><td>Cuser</td><td>ECount</td><td>ASpeed</td><td>XBytes</td></tr><tr><td>5</td><td>S</td><td>188.221.3.95/4096</td><td>070803 160124</td></tr><tr><td>RSK23</td><td>0</td><td>2154364</td><td>4308728</td></tr></table>	Ident	Sta	Address	Connected At	Cuser	ECount	ASpeed	XBytes	5	S	188.221.3.95/4096	070803 160124	RSK23	0	2154364	4308728
Ident	Sta	Address	Connected At														
Cuser	ECount	ASpeed	XBytes														
5	S	188.221.3.95/4096	070803 160124														
RSK23	0	2154364	4308728														

Displaying Connection attributes: *peldsp display_connection*

Syntax	<code>peldsp display_connection [(-ci)]</code>
Description	Use this command to display the supervision-oriented attributes of a Connection. The screen displays attributes for the Connection.

Parameters	<code>-connection_id (-ci)</code> : Enter an integer that uniquely identifies the connection.
Example	<pre> pelDSP display_connection -ci 5 c_connection_id = '5' c_connection_state = 'S' c_remote_address = '188.221.3.95/4096' c_date_connected = '070803 160124' c_protocol = 'FTP' c_mode = 'I' c_connected_user = 'RSK23' c_current_transfer_ident = '25468' c_transfer_date = '070803 163024' c_downloads_count = '3' c_uploads_count = '0' c_errors_count = '0' c_average_speed = '4308728' c_x_bytes = '2154364' </pre> For details of return parameters, refer to Connections: Parameters list.

Aborting a Connection: pelctl kill_connection

Syntax	<code>pelctl kill_connection [(-ci)]</code>
Description	Use this command to abort a specific Connection.
Parameters	<code>-connection_id (-ci)</code> : Enter an integer that uniquely identifies the connection.
Example	First, check the status with <code>pelDSP select_connection</code>: <pre>pelDSP select_connection -ca '070803 150000'</pre> Then abort the required connection: <pre>pelctl kill_connection -ci 5</pre> Finally, recheck the status with <code>pelDSP select_connection</code>: <pre>pelDSP select_connection -ca '070803 150000'</pre>

[Connections: Parameters list](#)

[Monitoring Transfers](#)

Monitoring Connected Users (command line)

[peldsp select connected user](#)

[peldsp status connected user](#)

[peldsp display connected user](#)

[pelctl disconnect user](#)

[pelctl reset connected user](#)

Displaying list of Connected Users: *peldsp select_connected_user*

Syntax	peldsp select_connected_user [(-cui) (-ca) (-cb)]
Description	Use this command to display a list of the active Connected Users (via their identifiers) that match the parameters that you specify. To list all Connected Users, enter the command with no parameter. Alternatively, identify the Connected User to select using the <code>-connected_user_id</code> parameter. Use the optional parameters to further refine the search.
Parameters	<code>-connected_user_id (-cui)</code> : Enter an integer that uniquely identifies the Connected User. Max 31 alphanumeric characters.
	<code>-connected_after (-ca)</code> : Enter a time and date in the format YYMMDD HHMMSS.
	<code>-connected_before (-cb)</code> : Enter a time and date In the format YYMMDD HHMMSS.
Example	<pre>peldsp select_connected_user -ca '070803 150000'</pre> <pre>RSK23 WNT002</pre> This command selects Connected Users whose first connection occurred after 3 p.m. (150000) on 3 August 2007 (070803).

Displaying Connected User status: *peldsp status_connected_user*

Syntax	peldsp status_connected_user [(-cui) (-ca) (-cb)]
Description	Use this command to display status information for the active Connected Users that match the parameters that you specify. Each element of this list comprises a Connected User identifier followed by status attributes. To list all Connected Users, enter the command with no parameter. Alternatively, refine the search using optional parameters.

Parameters	<code>-connected_user_id (-cui)</code> : Enter an integer that uniquely identifies the connected user. Max 31 alphanumeric characters.
	<code>-connected_after (-ca)</code> : Enter a time and date in the format <code>YYMMDD HHMMSS</code> .
	<code>-connected_before (-cb)</code> : Enter a time and date In the format <code>YYMMDD HHMMSS</code> .
Example	peldsp status_connected_user <pre> Ident Last Address Connected At Max ACount DCount Ucount ECount ASpeed XBytes RSK23 188.221.3.95/4096 070803 160124 10 124 102 22 0 2154364 4308728 WNT02 193.220.17.5/4096 070803 121107 10 14 8 6 0 1432864 2865728 </pre>

Displaying Connected User attributes: *peldsp display_connected_user*

Syntax	peldsp display_connected_user [(-cui)]
Description	Use this command to display the attributes of a specific Connected User. The screen displays attributes for the Connected User.
Parameters	<code>-connected_user_id (-cui)</code> : Enter an integer that uniquely identifies the connected user. Max 31 alphanumeric characters.
Example	peldsp display_connected_user -cui RSK23 <pre> c_connected_user_id = 'RSK23' c_last_remote_address = '188.221.3.95/4096' c_last_connection_date = '070803 160124' c_conn_max = '10' c_connection_date = '070803 163024' c_downloads_count = '3' c_uploads_count = '0' c_errors_count = '0' c_average_speed = '4308728' c_x_bytes = '2154364' </pre> For details of return parameters, refer to Connected Users: Parameters list.

Disconnecting a Connected User: *pelctl disconnect_user*

Syntax	pelctl disconnect_user [(-cui)]
Description	Use this command to disconnect a specific Connected User.

Parameters	<code>-connected_user_id (-cui)</code> : Enter an integer that uniquely identifies the Connected User. Max 31 alphanumeric characters.
Example	<pre> peldsp select_connected_user RSK23 WNT002 pelctl disconnect_user -cui WNT002 peldsp select_connected_user RSK23 </pre>

Resetting a Connected User: `pelctl reset_connected_user`

Syntax	<code>pelctl reset_connected_user [(-cui)]</code>
Description	Use this command to reset the Connected User quota to zero. This command is particularly useful when you are configuring the FTP <code>command_sensitivity</code> parameter. When a Connected User exceeds the command sensitivity quota, Gateway interrupts all current connections and forbids any further connections for this User.
Parameters	<code>-connected_user_id (-cui)</code> : Enter an integer that uniquely identifies the Connected User. Max 31 alphanumeric characters.
Example	<pre> peldsp select_connected_user RSK23 WNT002 pelctl reset_connected_user -cui WNT002 </pre>

[Connected Users: Parameters list](#)

[Monitoring Transfers](#)

Axway Gateway: Managing Transfers

Managing Statistics files (command line)

[About Statistics files](#)

[Managing Statistics files](#)

[pelctl archive stat](#)

[pelctl suspend stat](#)

[pelctl resume stat](#)

[pelctl archive conn stat](#)

```
pelctl suspend conn stat
```

```
pelctl resume conn stat
```

```
pelctl archive conn user stat
```

```
pelctl suspend conn user stat
```

```
pelctl resume conn user stat
```

About Statistics files

A **Statistics file** (or STAT file) is a formatted text file that contains records of terminated transfers.

The STAT file, named `stat.dat`, is located in:

```
<Gateway installation directory>/run_time/data/
```

Two other Statistics files were added, primarily for debugging use: `conn_stat.dat` and `conn_user_stat.dat`. These files store information about connections and connected users, each time a transfer ends.

Prerequisites for Statistics file generation

Statistics activation

To enable Gateway to generate STAT file records, activate the [Statistics option](#) in the configuration menu. When this option is activated, Gateway generates a record in the STAT file each time a transfer is terminated.

To enable Connection and Connected User Statistics, run:

```
peluconf set -s monitor conn_statistics_option yes
```

Statistics data dictionary file

The format of the records deposited to the Statistics file is determined by the contents of the **Statistics data dictionary file**. The Statistics data dictionary file is a formatted text file located in `<Gateway installation directory>/run_time/data/`. The name of the file is `statdict.dat`. This file contains a list of items separated by a comma. Each item is delimited by quotation marks.

The first item of a line must be the **STAT record format number**. It is possible to create a `statdict.dat` file with two or more lines, each beginning with a unique STAT record format number and followed by a set of attribute identifiers. You can then configure the Internal user exit `exituser/exitstat.c` to return a format number according to your own criteria. For example, for MSG type transfers you may choose to apply a specific STAT record format by programming `exitstat` to return a specific STAT record format number on MSG transfers. Note that format

numbers 3 and 4 are used to format connections and connected user statistics. You can change these format numbers from `exituser/existat.c` (`ExitConnStatMessage` and `ExitCousStatMessage`). Refer to [Internal user exits](#).

The other items in each line are the identifiers of transfer attributes for which Gateway generates statistics at the end of each transfer.

The default Statistics data dictionary record file comprises a single line. It resembles the following:

```
"1", "x_direction", "x_mode", "x_create_date", "x_state", "x_last_end_
reason", "x_x_bytes", "x_protocol", "x_originator", "x_destination", "x_dir_
path", "x_file_component"
```

where:

- "1" is the STAT record format number
- "x_direction" is the identifier of the end-of-transfer *direction* attribute whose actual values are stored in the generated statistics record
- "x_mode" is the identifier of the end-of-transfer *transfer mode* attribute whose actual values are stored in the generated statistics record
- and so on, for each successive identifier

Customizing the Statistics data dictionary file

You can configure the `statdict.dat` file so that it is useful for the examination of your transfer operations. To configure this file, open the file with a text editor and add or remove transfer attribute identifiers. On installation, this file contains a certain number of attribute identifiers as displayed above.

You can add any valid attribute that is generated during a transfer. For a list of possible attributes to use in the `statdict.dat` file, use the `peldsp display_trans` command to list the attributes of a typical transfer.

Additionally, you can add lines to the file, each line beginning with a unique STAT record format number and containing a set of attribute identifiers that together describe a statistics record format.

Statistics file generation

For each terminated transfer, Gateway:

- Replaces the record format parameters, defined in the Statistics data dictionary (`statdict.dat`), with their actual values
- Writes the record in the STAT file (`stat.dat`)

A resulting default statistics data file record in `stat.dat` may look something like this:

```
"1", "S", "I", "970430 100520", "E", "0", "12345", "FTP", "SITE_ORG",
"SITE_DEST", "/directory", "file"
```


Managing Statistics files

Use the following `pelctl` commands to perform Statistics file management operations.

`pelctl archive_stat`: Archive the Statistics file

Syntax	<code>pelctl archive_stat</code>
Description	This command copies the content of the STAT (statistics) file into an archive file, and then deletes the current STAT file. After the operation, log inputs are recorded in a new empty STAT file. The archive storage directory and file name are platform-dependent. For example, on UNIX platforms, the archive file name is the original file name with the suffix containing the timestamp of the archive request. (The archived log will have the format: <code>stat.dat_YYYYMMDD_HHMMSS</code>). In the Gateway configuration menu, you can specify the duration and/or the number of Kbytes whose threshold automatically triggers STAT file archiving. Recommendation: Archive the STAT file regularly to avoid the file becoming too large.
Parameters	None
Example	For UNIX platforms: Following an archive command, <code>pelctl archive_stat</code> executed on February 12, 2014 at 15 h 33 min. 22 sec, the archive is created, with file name: <pre><Gateway installation directory><Gateway installation directory>/run_time/data/stat.dat_20140212_153322</pre> The STAT file is a text file that you can view using a text editor. The lines of message text written to the log file record event occurrences.

`pelctl suspend_stat`: Suspend Statistics recording

Syntax	<code>pelctl suspend_stat</code>
Description	This command suspends Statistics recording.
Parameters	None
Example	Enter: <pre>pelctl suspend_stat</pre> Gateway suspends the writing of records to the Statistics file.

pelctl resume_stat: Resume recording in the Statistics file

Syntax	<code>pelctl resume_stat</code>
Description	This command resumes Statistics recording suspended by a <code>pelctl suspend_stat</code> command.
Parameters	None
Example	Enter: <code>pelctl resume_stat</code> Gateway resumes the writing of records to the Statistics file.

pelctl archive_conn_stat: Archive the Connection Statistics file

Syntax	<code>pelctl archive_conn_stat</code>
Description	This command copies the content of the Connection statistics file into an archive file, and then deletes the current Connection statistics file. After the operation, log inputs are recorded in a new empty connection statistics file. The archive storage directory and filename are platform-dependent. For example, on UNIX platforms, the archive filename is the original filename with the suffix containing the timestamp of the archive request. (The archived log will have the format: <code>conn_stat.dat_YYYYMMDD_HHMMSS</code>). In the Gateway configuration menu, you can specify the duration and/or the number of Kbytes whose threshold automatically triggers STAT file archiving. Recommendation: Archive the Connection Statistics file regularly to avoid the file becoming too large.
Parameters	None
Example	For UNIX platforms: Following an archive command, <code>pelctl archive_conn_stat</code> executed on February 12, 2014 at 15 h 33 min. 22 sec, the archive is created, with the filename: <pre><Gateway installation directory><Gateway installation directory>/run_time/data/conn_stat.dat_20140212_153322</pre> The Connection Statistics file is a text file that you can view using a text editor. The lines of message text written to the log file record event occurrences.

pelctl suspend_conn_stat: Suspend Connection Statistics recording

Syntax	<code>pelctl suspend_conn_stat</code>

Description	This command suspends Connection Statistics recording.
Parameters	None
Example	Enter: <pre>pelctl suspend_conn_stat</pre> Gateway suspends the writing of records to the Connection Statistics file.

pelctl resume_conn_stat: Resume recording in the Connection Statistics file

Syntax	<code>pelctl resume_conn_stat</code>
Description	This command resumes Statistics recording suspended by a <code>pelctl suspend_conn_stat</code> command.
Parameters	None
Example	Enter: <pre>pelctl resume_conn_stat</pre> Gateway resumes the writing of records to the Connection Statistics file.

pelctl archive_conn_user_stat: Archive the Connected User Statistics file

Syntax	<code>pelctl archive_conn_user_stat</code>
Description	This command copies the content of the Connected User statistics file into an archive file, and then deletes the current Connected User statistics file. After the operation, log inputs are recorded in a new empty connected user statistics file. The archive storage directory and file name are platform-dependent. For example, on UNIX platforms, the archive file name is the original file name with the suffix containing the timestamp of the archive request. (The archived log will have the format: <code>conn_user_stat.dat_YYYYMMDD_HHMMSS</code>). In the Gateway configuration menu, you can specify the duration and/or the number of Kbytes whose threshold automatically triggers STAT file archiving. Recommendation: Archive the Connected User Statistics file regularly to avoid the file becoming too large.
Parameters	None

Example	For UNIX platforms: Following an archive command, <code>pelctl archive_conn_user_stat</code> executed on February 12, 2014 at 15 h 33 min. 22 sec, the archive is created, with the filename: <pre><Gateway installation directory><Gateway installation directory>/run_time/data/conn_user_stat.dat_20140212_ 153322</pre> The Connected User Statistics file is a text file that you can view using a text editor. The lines of message text written to the log file record event occurrences.

pelctl suspend_conn_user_stat: Suspend Connection Statistics recording

Syntax	<code>pelctl suspend_conn_user_stat</code>
Description	This command suspends Connection Statistics recording
Parameters	None
Example	Enter: <pre>pelctl suspend_conn_user_stat</pre> Gateway suspends the writing of records to the Connected User Statistics file.

pelctl resume_conn_user_stat: Resume recording in the Connection Statistics file

Syntax	<code>pelctl resume_conn_user_stat</code>
Description	This command resumes Statistics recording suspended by a <code>pelctl suspend_conn_user_stat</code> command.
Parameters	None
Example	Enter: <pre>pelctl resume_conn_user_stat</pre> Gateway resumes the writing of records to the Connected User Statistics file.

Axway Gateway: Managing Transfers

Connections: Parameters List

Connections: Command parameters

The following table lists all Connection object **command parameters** in alphabetical order. The corresponding field name in the Gateway GUI is shown in *italics*.

Parameter	Meaning	Possible values
-connected_ after (-ca) Connection time/ Start	Date and time <i>after</i> which the Connection started.	Format: YYMMDD HHMMSS
-connected_ before (-cb) Connection time/ End	Date and time <i>before</i> which the Connection started.	Format: YYMMDD HHMMSS
-connection_ id (-ci) Connection ID	Connection identifier.	Integer that exclusively identifies the Connection.
-mode (-m) Mode	Connection mode of Gateway for the current Connection.	• I = Initiator • R = Responder
-state (-s) State	Connection state.	• T = Transferring • S = Still

Connections: Return parameters

The following table lists all Connection object **returned values** in alphabetical order. The corresponding field name in the Gateway GUI is shown in *italics*.

Parameter	Meaning
average_speed Speed	Average transfer speed for this Connection. Only significant transfers (more than one second) are taken into account.
connected_user Connected user	Connected User responsible for the Connection.
connection_id ID	Connection identifier.
connection_state State	Connection state: • T = Transferring • S = Still
current_transfer_ident	If any, the identifier of an active transfer using this Connection.
date_connected	Date and time the Connection started. Format: YYMMDD HHMMSS
downloads_count Downloads	Number of downloaded files during this Connection.

Parameter	Meaning
errors_count	Number of transfer errors during this Connection.
mode	Connection mode of Gateway for the current Connection: • I = Initiator • R = Responder
protocol	Protocol name.
remote_address	Outgoing Connection network address.
transfer_date	Date and time the transfer started. Format: <i>YYMMDD HHMMSS</i>
uploads_count Uploads	Number of uploaded files during this Connection.
x_bytes Data	Cumulated transferred data volume for this Connection.

[Monitoring Connections \(command line\)](#)

Axway Gateway: Managing Transfers

Connected Users: Parameters List

Connected Users: command parameters

The following table lists all Connected User object **command parameters** in alphabetical order. The corresponding field name in the Gateway GUI is shown in *italics*.

Parameter	Meaning	Possible values
-connected_ after (-ca) First connection time/ Start	Date and time <i>after</i> which the Connection started.	Format: <i>YYMMDD HHMMSS</i>
-connected_ before (-cb) First connection time/ End	Date and time <i>before</i> which the Connection started.	Format: <i>YYMMDD HHMMSS</i>

Parameter	Meaning	Possible values
-connected_ user_id (- cui) Connected user	Connected User identifier.	Integer that exclusively identifies the Connected User. Max 31 alphanumeric characters.

Connected Users: Return parameters

The following table lists all Connected User object **returned values** in alphabetical order. The corresponding field name in the Gateway GUI is shown in *italics*.

Parameter	Meaning
active_ connections Connections count	Number of currently active (transferring data) Connections.
average_speed Speed	Average transfer speed for this Connected User. Only significant transfers (more than one second) are taken into account.
command_ credit_used Command credit used	FTP (defense against hammer attacks). Percentage of command credit used.
conn_max Max connections	Maximum number of Connections allowed for this Connected User.
connected_ user_id Connected user name	Connected User responsible for the Connection.
download_ requests_used Download requests used	FTP (defense against hammer attacks). Number of download requests used within the user quota.
download_ size_used Download size used	FTP (defense against hammer attacks). Percentage of upload data size used within the user quota.

Parameter	Meaning
downloads_ count Downloads count	Number of downloaded files for this Connected User.
errors_ count Errors count	Number of transfer errors for this Connected User.
last_ connection_ date Last connection date	Date and time the last Connection started. Format: <i>YYMMDD HHMMSS</i>
last_remote_ address Last remote address	Outgoing network address of the last Connection.
upload_ requests_ used Upload requests used	FTP (defense against hammer attacks). Number of upload requests used within the user quota.
upload_size_ used Upload size used	FTP (defense against hammer attacks). Percentage of upload data size used within the user quota.
uploads_ count Uploads count	Number of uploaded files for this Connected User.
x_ bytes Data count	Cumulated transferred data volume for this Connected User.

[Monitoring Connected Users \(command line\)](#)

Axway Gateway: Managing Transfers

Log files

About Log files

Gateway stores traces of all monitoring, protocol, and network events in a **Log** file. This file provides you with detailed transfer tracking information.

The Log file contains a trace of all significant events relating to transfer monitor activities and to the transfers performed. Messages contained in the Log file are time-stamped. New messages are added to the end of the file as they are generated. This enables you to view messages in chronological order, inspecting processing events step by step.

You can display the entire contents of the Log file, or use filtering criteria to view a selected subset of Log messages.

When you start Gateway, the log file is either automatically created, or reopened if it already exists. By default, Gateway creates the first Log file at:

```
<Gateway installation directory>/run_time/data/log.dat
```

Once Gateway begins logging to the Log file, it continues until one of the following events occurs:

- Gateway stops
- You suspend logging
- You archive the current Log file

When you archive a Log file, Gateway creates an archive file from the current `log.dat` file by adding the archiving timestamp.

Gateway names the archived log file `log.dat_20040212_153322`, where `20040212_153322` represents the date and time when the archiving was triggered.

Working with Log files

You can work with the Log:

- In the GUI
- Via the Command Line Interface
- By navigating to the log directory on your host machine, and reading the Log file in a text editor (the log is an editable text file).

[Working with Logs](#)

[Working with Logs \(command line\)](#)

[Overview: Monitoring Axway Gateway](#)

[Monitoring Transfers](#)

Axway Gateway: Managing Transfers

Log files

About Log files

Gateway stores traces of all monitoring, protocol, and network events in a **Log** file. This file provides you with detailed transfer tracking information.

The Log file contains a trace of all significant events relating to transfer monitor activities and to the transfers performed. Messages contained in the Log file are time-stamped. New messages are added to the end of the file as they are generated. This enables you to view messages in chronological order, inspecting processing events step by step.

You can display the entire contents of the Log file, or use filtering criteria to view a selected subset of Log messages.

When you start Gateway, the log file is either automatically created, or reopened if it already exists. By default, Gateway creates the first Log file at:

```
<Gateway installation directory>/run_time/data/log.dat
```

Once Gateway begins logging to the Log file, it continues until one of the following events occurs:

- Gateway stops
- You suspend logging
- You archive the current Log file

When you archive a Log file, Gateway creates an archive file from the current `log.dat` file by adding the archiving timestamp.

Gateway names the archived log file `log.dat_20040212_153322`, where `20040212_153322` represents the date and time when the archiving was triggered.

Working with Log files

You can work with the Log:

- In the GUI
- Via the Command Line Interface
- By navigating to the log directory on your host machine, and reading the Log file in a text editor (the log is an editable text file).

[Working with Logs](#)

[Working with Logs \(command line\)](#)

[Overview: Monitoring Axway Gateway](#)

[Monitoring Transfers](#)

Secure Relay Master Agent messages (XSR MA)

This following is a list of Secure Relay Master Agent termination log messages:

100	XMA E %1
<i>Type</i>	Error
<i>Meaning</i>	Error message from Secure Relay Master Agent related to a Security Termination.
<i>Parameters</i>	1 Message from Secure Relay Master Agent regarding the reason for the error message.
<i>Action</i>	Investigate Passport Certificates related to Error message.
101	XMA I %1
<i>Type</i>	INFO
<i>Meaning</i>	INFO message from Secure Relay Master Agent related to successful Security Termination. These messages contain the Certificate Chains according to which the received certificate was validated.
<i>Parameters</i>	1 Message from Secure Relay Master Agent containing the Certificate Chains.
<i>Action</i>	None.

Axway Gateway: Managing Transfers

Working with Logs

[Viewing and managing Logs](#)

[Viewing archived logs](#)

[Viewing Secure Relay Master Agent Logs](#)

[Suspending Log recording](#)

[Archiving the current log](#)

[Filtering the Log message display](#)

[Refreshing the log display](#)

Viewing and managing Logs

To view a complete list of all log messages contained in the current Log file:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Click the **Log** node.

Gateway displays all logged messages in a table in the display pane (right pane). Each row of the table represents a logged message. The table columns contain the following information for each message.

Column heading	Contents	Data format
Status	Message identifier, indicating the Gateway component that generated the message and the message type	Icon and text
Date	Date that the message was logged	dd:MM:yy
Time	Time that the message was logged	HH:mm:ss
Message	Message text strings detailing the logged event	Text

Customizing the display

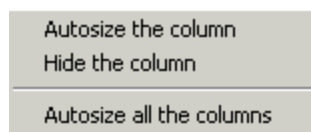
By default, Gateway displays the logged messages with each of the above columns visible and with the messages displayed from top to bottom in the order in which they were generated.

To change the order of the display, click one of the column headers cells.

- Click the Status header cell once to rearrange the display in ascending alphabetical order of Status column identifiers. Click a second time to display in descending order.
- Click the Date or Time header cells to rearrange the display in descending chronological order. Click a second time to return to ascending chronological order.
- Click the Message header cell to rearrange the display in ascending alphabetical order of Message text strings. Gateway lists non-alphabetic characters (for example, the quotation mark) before alphabetical characters. Click a second time to display in descending alphabetic order of message text strings.

To change the size of displayed columns, right-click any column.

Gateway displays the following context menu.



Select the command:

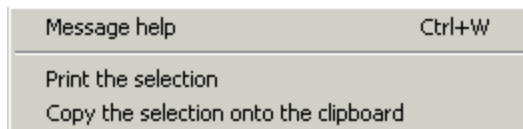
- **Autosize the column**, to adjust the selected column size to the column contents
- **Hide the column**, to remove the entire column from view
- **Autosize all the columns**, to adjust all columns to the size of their contents

Managing displayed messages

In the Log display pane, Gateway provides two additional context menus that you can use to handle the displayed messages.

Status column context menu

Right-click the *Status* column of any message in the display pane. Gateway displays the following context menu.

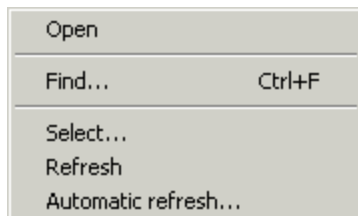


Select the command:

- **Message help**, to open a window displaying detailed help information related to the message type
- **Print the selection**, to print the message
- **Copy the selection onto the clipboard**, to record the selected message to the clipboard memory for recopying to a text processing or mail tool

Date/Time/Message context menu

Right-click *Date*, *Time* or *Message* column of any message in the display pane. Gateway displays the following context menu.



Select the command:

- **Open**, to open a new window containing a list of the displayed messages
- **Find...**, to open a search dialog box, enabling you to search the log message list for specific character strings
- **Select...**, to open the Select Log Messages dialog box
Use the fields of the Select Log Messages dialog box to create selection criteria for the display of Log messages. See [Filtering the Log message display](#) on how to use these fields.

Gateway applies your criteria to regenerate the display results.

- **Refresh**, to refresh the display of Log messages, taking into account the most recent network events.
- **Automatic refresh...**, to open a dialog box that enables you to set an automatic refresh cycle time interval. You can set the refresh interval in units of 10 seconds (10, 20, 30 seconds, etc.)

You can use one or more of the fields described in the following table:

Field	Use
Maximum number	Enter the maximum number of message results you want to view
Date and start time	You can select either a starting date or starting time criteria, or combine the two. You can combine starting time and/or starting date criteria with ending time and/or ending date criteria. To set a starting date as selection criteria: 1. Click the down arrow on the starting date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the start month and start date of your choice. To set a starting time as selection criteria: 1. Click the check box next to the starting time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

Field	Use
Date and end time	You can set either an ending date or ending time criteria, or combine the two. You can combine ending time and/or ending date criteria with starting time and/or starting date criteria. To set an ending date as selection criteria: 1. Click the down arrow on the ending date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the end month and end date of your choice. To set an ending time as selection criteria: 1. Click the check box next to the ending time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

Field	Use
Sending process	To select a sending process as display selection criteria: 1. Click the down arrow located in the Sending process field. Gateway displays a drop-down list: ◦ CHK ◦ EDI ◦ EXT ◦ FPSIT ◦ FTP ◦ HTTP ◦ JCT ◦ NDSP ◦ NET ◦ NOT ◦ ODT ◦ PCNX ◦ PEL ◦ POP ◦ SECS ◦ SMTP ◦ SNA ◦ SR ◦ SUP ◦ TDRV ◦ TPM (PassPort) ◦ TRADE ◦ TSD ◦ USR ◦ VFD 2. Select the sending process you want to use as display selection criteria.
Text extract	Enter all or part of a text common to all the messages you want to display.

When you have set the criteria that you want to use to filter the display results, click **OK**.

Gateway applies your criteria to regenerate the display results.

- **Refresh**, to refresh the display of Log messages, taking into account the most recent network events
- **Automatic refresh...**, to open a dialog box that enables you to set an automatic refresh cycle time interval. You can set the refresh interval in units of 10 seconds (10, 20, 30 seconds, etc.)

Viewing Archived Logs

To view a complete list of all archived log files:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Click the **Archived Log** node.

Gateway displays all archived log files in a table in the display pane (right pane). Each row of the table represents a log file. The table columns contain the following information for each message:

Column heading	Contents	Data format
File Name	Archived Log file name	Icon and text
Date	Date when was the last writing into file	dd.MM.yy
Time	Date when was the last writing into file	HH:mm:ss

3. Double-click the archived file name to view it. A sub-tree item is created containing the file name for each archived log the user loads.

Note: Pay attention to how many archived logs you are loading.

Viewing Secure Relay Master Agent Logs

To view the [log messages for Secure Relay Master Agent related to security termination](#):

1. In the left pane of the UI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Click the **Secure Relay MA Log** node.

Gateway displays only the Secure Relay Master Agent logged messages related to security termination in a table in the display pane (right pane). Each row of the table represents a logged message. The table column descriptions are the same as for Log.

Security termination for both TLS and SSH requires Passport. If the security termination is not possible because Passport returns an error to the Secure Relay Router Agent, then the Secure Relay Router Agent sends a message to the Secure Relay Master Agent, which is logged in Gateway. If the connection to Passport cannot be established then this message is also sent to Gateway.

There are two types of messages which you will be able to see in the Secure Relay MA Log: Security termination failure and Full Certificate path for successful security termination. However, this depends on the Secure Relay Master Agent log configuration. To control the Secure Relay Master Agent configuration you must access the `log.properties` file from the `<install_dir>/run_time/xsr` directory. Locate `log4j.category.com.axway.xsr.log.securityTermination=ERROR, XSRTERM_LOGFILE`, and set the level you need for logging:

- **ERROR** – Logs for security termination failure
- **DEBUG** – Logs for both security termination failure and successful security termination (contains the full certificate path)

The filter criteria that can be applied to a Secure Relay MA Log are similar to a log, except for the Sending process which is missing from the filter selection. You can use the same filter that you use on the Log and Archived log, but when applying a filter on Secure Relay MA Log, the Sending process, if set, will be ignored.

If there are several Gateways connected to the same Secure Relay Router Agent, then the error message is sent only to the Gateway that opened the port on which security termination failed.

Note You might encounter a "File Not Found" error when selecting a Secure Relay MA Log if Secure Relay is not activated.

Limitations

If the connection with Secure Relay Master Agent is lost then Secure Relay does not store any messages. As a result, these messages will be lost.

If a connection is closed by a peer before the Secure Relay Router Agent notifies Gateway, or if a connection is dropped by the Secure Relay Router Agent as a result of an internal error, this will not be logged.

Suspending Log recording

To suspend the recording of messages to the Log:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Right-click the **Log** node.

Gateway displays a context menu.

3. From the context menu select **Suspend recording**.

Archiving the current Log

To archive the current Log file:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Right-click the **Log** node.

Gateway displays a context menu.

3. From the context menu select **Archive...**

Filtering the Log message display

To select filter criteria for the display of current log messages:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Right-click the **Log** node.

Gateway displays a context menu.

3. From the Log context menu select **Select...**

Gateway displays the *Select Log Messages* dialog box.

Use the fields of the *Select Log Messages* dialog box to create selection criteria for the display of Log messages. You can use one or more of the fields described in the following table.

Field	Use
Maximum number	Enter the maximum number of message results you want to view.

Field	Use
Date and start time	You can select either starting date or starting time criteria, or combine the two. You can combine starting time and/or starting date criteria with ending time and/or ending date criteria. To set a starting date as selection criteria: 1. Click the down arrow on the starting date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the start month and start date of your choice. To set a starting time as selection criteria: 1. Click the check box next to the starting time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

Field	Use
Date and end time	You can set either ending date or ending time criteria, or combine the two. You can combine ending time and/or ending date criteria with starting time and/or starting date criteria. To set an ending date as selection criteria: 1. Click the down arrow on the ending date field displaying the current date. Gateway displays the calendar selection tool. 2. Use the calendar selection tool to navigate to the end month and end date of your choice. To set an ending time as selection criteria: 1. Click the check box next to the ending time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated. 2. Click to select the time unit you want to modify: hour, minute or second. Gateway highlights the time unit. 3. Type a valid number or use the up and down arrows to set the time unit. 4. Select and set the other time units (hour, minute, second) as required.

Field	Use
Sending process	To select a sending process as display selection criteria: - Click the down arrow located in the Sending process field. Gateway displays a drop-down list: • CHK • EDI • EXT • FPSIT • FTP • HTTP • JCT • NDSP • NET • NOT • ODT • PCNX • PEL • POP • SECS • SMTP • SNA • SR • SUP • TDRV • TPM (PassPort) • TRADE • TSD • USR • VFD - Select the sending process you want to use as display selection criteria.
Text extract	Enter all or part of a text common to all the messages you want to display.

4. To create a new filter:
 - a. Click **Add New Filter**, to create a blank template for the new filter.
 - b. Enter a name for the Filter.
 - c. Choose whether the filter is available only for this server, or whether it is global. (This option cannot be changed later).
 - d. Configure Selection criteria.
 - e. Add a text comment for this filter. (optional)
5. To delete a filter:
 - a. Select the filter.
 - b. Click **Delete Filter**.
6. To modify a filter:
 - a. Select the filter.
 - b. Configure Selection criteria.
 - c. Click **Save and apply**.

Refreshing the Log display

To refresh the display of Log messages, taking into account the most recent network events:

1. In the left pane of the GUI main window, click  to expand the nodes:

Transfer Management > Monitoring

2. Right-click the **Log** node.

Gateway displays a context menu.

3. From the Log context menu select **Refresh....**

Gateway regenerates the display, adding any new event messages.

Axway Gateway: Managing Transfers

Working with Logs (command line)

[`pelctl archive log`](#)

[`pelctl suspend log`](#)

[`pelctl resume log`](#)

pelctl archive_log: Archive the Log

Syntax	<code>pelctl archive_log</code>

Description	This command copies the content of the Log file into an archive file, and then deletes the current log file. After the operation, the log inputs are recorded in a new empty log file. The archive storage directory and filename are platform-dependent. For example, on UNIX platforms, the archive filename is the original filename with the suffix containing the timestamp of the archive request. (The archived log will have the format: <code>log.dat_YYYYMMDD_HHMMSS</code>). When configuring Gateway, you can specify the duration and/or the number of Kbytes whose threshold automatically triggers Log file archiving.
Parameters	None
Example	Example archive filename on UNIX: <pre><Gateway installation directory>/run_time/data/ log.dat_20040212_153322"</pre> The Log file is a text file that you can view using a text editor. The lines of message text in the log file are based on event occurrences. Recommendation: Archive the log file regularly to avoid the file becoming too large.

pelctl suspend_log: Suspend message recording to the Log

Syntax	<code>pelctl suspend_log</code>
Description	This command suspends Log recording.
Parameters	None

pelctl resume_log: Resume message recording to the Log

Syntax	<code>pelctl resume_log</code>
Description	This command resumes Log recording suspended by a previous <code>pelctl suspend_log</code> command.
Parameters	None

Working with Logs (command line)

[`pelctl archive log`](#)

[`pelctl suspend log`](#)

[`pelctl resume log`](#)

pelctl archive_log: Archive the Log

Syntax	<code>pelctl archive_log</code>
Description	This command copies the content of the Log file into an archive file, and then deletes the current log file. After the operation, the log inputs are recorded in a new empty log file. The archive storage directory and filename are platform-dependent. For example, on UNIX platforms, the archive filename is the original filename with the suffix containing the timestamp of the archive request. (The archived log will have the format: <code>log.dat_YYYYMMDD_HHMMSS</code>). When configuring Gateway, you can specify the duration and/or the number of Kbytes whose threshold automatically triggers Log file archiving.
Parameters	None
Example	Example archive filename on UNIX: <pre><Gateway installation directory>/run_time/data/ log.dat_20040212_153322"</pre> The Log file is a text file that you can view using a text editor. The lines of message text in the log file are based on event occurrences. Recommendation: Archive the log file regularly to avoid the file becoming too large.

pelctl suspend_log: Suspend message recording to the Log

Syntax	<code>pelctl suspend_log</code>
Description	This command suspends Log recording.
Parameters	None

pelctl resume_log: Resume message recording to the Log

Syntax	<code>pelctl resume_log</code>
Description	This command resumes Log recording suspended by a previous <code>pelctl suspend_log</code> command.
Parameters	None

Axway Gateway: Transfer Examples

Transfer examples

Refer to the following topics for examples of file transfer procedures:

- [Executing a basic PeSIT transfer](#)
- [Executing a basic FTP transfer](#)
- [Executing an SFTP transfer](#)
- [Executing an FTP transfer using TLS and certificates](#)
- [Executing an EDIINT AS2 transfer](#)

These examples were designed for users who do not already have samples installed on their Gateway.

Axway Gateway: Transfer Examples

Executing a basic PeSIT transfer

[About this example](#)

[Prerequisites](#)

[Executing the Transfer Request](#)

[Transfer results](#)

About this example

This topic describes a basic configuration that enables you to execute a Transfer Request between two computers, both hosting Gateways. The purpose of this topic is to provide you with the minimum set of parameters you need to accomplish a transfer. In the context of your daily operations you can add functions, and adjust this example configuration to adapt it to your specific requirements.

In this exercise, we use the Gateway GUI to transfer a `.txt` formatted file from **machine A** (the Initiator/Sender), to **machine B** (the Responder/Receiver) via the PeSIT E protocol over TCP/IP.

We use a minimal set of configuration values. This enables us to demonstrate basic transfer requirements and to facilitate problem solving in the event of transfer failure.

Prerequisites

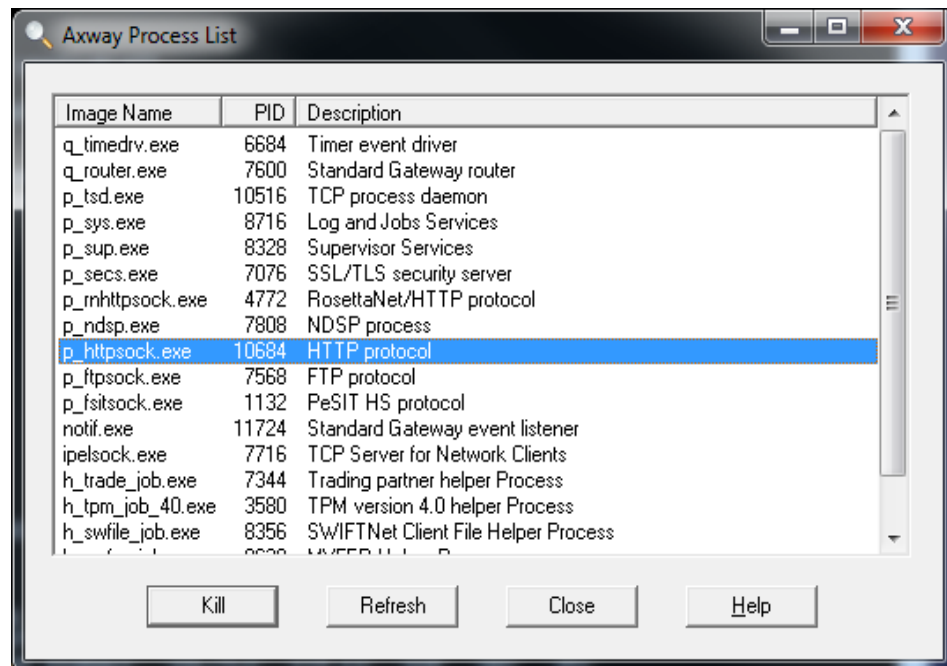
Installation prerequisites

- Gateway must be correctly installed and running on **machine A** and **machine B**.
- The two machines must be linked by a TCP/IP connection.
- You require a Windows environment to support the Navigator (GUI).

Hint: For Gateways hosted on Windows platforms, to confirm that Gateway is operating correctly, go to the Windows **Start** menu and select **Programs > Axway Software > Axway [ProjectName]> Gateway > Process List**

The application displays the *Process List* window containing a list of all of the currently running Gateway processes.

Example Process List display:



The list of processes running on your machine may vary depending on your protocol configuration.

Configuration prerequisites

- The sending **machine A** and receiving **machine B** must both support the PeSIT E protocol for the transfer.
- [Activate the PeSIT E protocol](#) on both machines.

CGateGroup and CGate prerequisites

A CGate provides security control for incoming connections. During the connection phase of a transfer, the CGate object provides the information that determines whether or not Gateway allows a given remote machine to connect.

In our transfer example, only the Responder/Receiver **machine B** requires a CGate object for access negotiation. The Initiator/Sender **machine A** is not concerned with an incoming connection attempt.

On **machine B**, first create a CGateGroup:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup...**

Gateway displays the *New CGateGroup* window.

3. Enter values in *only* the following fields of the window:
 - **Name:** Enter **PESITTEST**
 - **State:** Select **Enabled** from the drop-down list
 - **Protocol:** Enter **PESIT E**
 - **Called SAP:** Enter * (star character)
(If you know the port that is locally configured for the PeSIT protocol, you can enter it in this field. The default port is **6330**.)
4. Click **OK**. For this exercise is not necessary to enter any other CGateGroup information.
Gateway creates the group and adds it to the directory tree as a subdirectory of the CGate folder.

On **machine B** in the CGateGroup you just created, create a CGate object for the **machine A** -> **machine B** transfer connection:

1. Right-click the **PESITTEST** CGateGroup, and select **New > CGate** from the context menu to display the *New CGate* window.
2. Enter values in *only* the following fields of the window:
 - **Name:** Enter **PESITALL**
 - **State:** Select **Enabled** from the drop-down list

To keep the connection negotiation simple, do not enter any values in the **ID** or **Password** fields of this tab for this exercise.

3. Click **OK**.

Gateway creates the CGate and adds it to the directory tree as a subdirectory of the PESITTEST folder.

Site prerequisites

File transfers occur between two machines. In a transfer using Axway MFT, each participating machine is referred to as a *Site*. Each of the two Sites in our transfer must recognize the other partner Site to ensure that they can agree on the parameters of the exchange. Gateway uses Site objects to identify Sites. A Site object contains the definitions that enable one site to communicate with another.

Machine A: Defining the Remote Site

In our Transfer Request, machine A will send a file to another machine (Remote Site).

We identify the characteristics of the remote machine (**machine B**) to Gateway on **machine A** by creating a *Remote Site object*.

On **machine A**:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click **Remote Site**, then from the context menu that appears, select **New**.
Gateway displays the *New Remote Site* window.
3. In the **General** tab of the *New Remote Site* window, enter values in the following fields:
 - **Alias**: Enter the HostName of **machine B**. If, for example, **machine B** has HostName machineB, enter **machineB**
 - **Protocol**: From the drop-down list select **PeSITE E**.
When you select this protocol, Gateway displays two additional tabs in the *New Remote Site* window.
 - **Protocol identifier**: Use the same protocol identifier as you use in the default Local Site definition on **machine B**. In our example our Protocol identifier is **machineB** (in this case it is identical to the HostName).

To confirm the protocol identifier of **machine B**:

- In the left pane of the GUI main window, click  to expand the nodes: **Partner Management > Sites > Local Site**
 - Right-click **Local Site**.
 - In the display pane (right pane), right-click the name of the Local Site. Select **View...** from the context menu that appears. Gateway opens the *Local Site view* window.
 - Note the name in the **Protocol Identifier** field.
 - **Group name**: Gateway enters the default group name **GDEFAULT**. Do not change this value.
 - **Role frame**: In this frame, click to select **Responder/Receiver** role for **machine B**.
 - **Initialization mode**: In this frame, click to select **Responding Site** and **Trace**.
4. In the **Options** tab of the *New Remote Site* window, accept all default values.

5. In the **Network address** tab of the *New Remote Site* window:
 - Select the **Main address** radio button.
 - Right-click the **Network type** field, and select **TCP** from the drop-down list.
When you select **TCP**, Gateway displays IP address, port number and Proxy fields in the tab.
 - Enter the IP address of the Remote Site machine.
 - Enter the TCP port for transfer access. For PeSIT transfers, the default port is **6330**.
6. In the **PeSIT** tab of the *New Remote Site* window, accept all default values. *Do not* enter any ID or passwords in provided fields.
7. In the **Net security** tab of the *New Remote Site* window, accept the default value, **None**.
8. Click **OK** to accept the parameter values.
Gateway creates the machine B Remote Site object.

Machine B: Defining the Remote Site

In our Transfer Request, machine B will receive a file from another machine (Remote Site).

We identify the characteristics of the remote machine (**machine A**) to Gateway on **machine B** by creating a Remote Site object.

On **machine B**:

1. In the left pane of the GUI main window, click  to expand the nodes: **Partner Management > Sites**
2. Right-click **Remote Site**. From the context menu that appears, select **New**
Gateway displays the *New Remote Site* window.
3. In the **General** tab of the *New Remote Site* window, enter values in the following fields:
 - **Alias**: Enter the HostName of **machine A**. For example, if **machine A** has as HostName *machineA*, enter **machineA**
 - **Protocol**: Select **PeSITE E** from the drop-down list.
When you select this protocol, Gateway displays two additional tabs in the *New Remote Site* window.
 - **Protocol identifier**: Use the same protocol identifier as you use in the default Local Site definition on **machine A**. In our example our Protocol identifier is **machineA** (in this case it is identical to the HostName).

To confirm the protocol identifier of **machine A**:

- On **machine A**, in the left pane of the GUI main window, click  to expand the nodes: **Partner Management > Sites > Local Site**
- Right-click **Local Site**.

- In the display pane (right pane), right-click the name of the Local Site. Select **View...** from the context menu that appears. Gateway opens the *Local Site* view window.
 - Note the name in the **Protocol Identifier** field.
 - **Group name:** Gateway enters the default group name **GDEFAULT**. Do not change this value.
 - **Role frame:** In this frame, click to select **Initiator/Sender** role for **machine A**.
 - **Initialization mode:** In this frame, click to select **Initiating Site** and **Trace**.
4. In the **Options** tab of the *New Remote Site* window, accept all default values.
 5. In the **Network address** tab of the *New Remote Site* window:
 - Select the **Main address** radio button.
 - Right-click the **Network type** field, and select **TCP** from the drop-down list.
When you select TCP, Gateway displays IP address, port number and Proxy fields in the tab.
 - Enter the IP address of the Remote Site machine.
 - Enter the TCP port for transfer access. For PeSIT transfers, the default port is **6330**.
 6. In the **PeSIT** tab of the *New Remote Site* window, accept all default values. *Do not* enter any ID or passwords in provided fields.
 7. In the **Net security** tab of the *New Remote Site* window, accept the default value, **None**.
 8. Click **OK** to accept the parameter values.
Gateway creates the **machine A** Remote Site object.

Executing the Transfer Request

We want to transfer the file `Test.txt` from **machine A** to **machine B**. For this example we assume that the file `Test.txt` is located in `C:\temp`.

To transfer the file:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click **Transfers**, and then select **New > Transfer Request** in the context menu.
Gateway displays the *New Transfer Request* window.
3. In the **General** tab of the *New Transfer Request* window, enter values in the following fields:
 - **Type:** From the drop-down list, select the request type **Trans** (Transfer Request).
 - **Mode:** From the drop-down list, select the mode in which the local site operates, **Initiator**. You are initiating the transfer with your Local Site in the client (initiator) role.
 - **Direction:** From the drop-down list, select the direction of your transfer, **Send**.

- **Application:** From the drop-down list, select an application to define file system attributes for your file transfer. Gateway provides the default application *DEFAULT_A* for PeSIT transfers of files in ASCII format. Select **DEFAULT_A** from the drop-down list.
 - **Sites/Destination Alias:** In the Destination Alias field enter the alias of the destination machine, **machineA**, exactly as you entered it when you created your Remote Site object.
4. In the **Attributes** tab of the *New Transfer Request* window, enter the name of the file you want to transfer and the path to that file. For this example, enter: `C:\temp\Test.txt`
Leave the remaining fields of this tab empty for this example.
 5. For the remaining tabs, accept all default values. The default values of the remaining fields are either empty (no value) or *undefined*.
 6. Click **OK** to validate the values and initiate the Transfer Request.
Gateway begins processing of the transfer, and attributes a number to the transfer which it displays in a pop-up window.



Transfer results

When you submit the Transfer Request, Gateway analyzes the information you have provided for the transfer in the Transfer Request and in the associated Site and Application objects. If the network and application conditions are correct, the transfer is executed and a copy of the file is deposited on **machine B**.

Using the Mailbox

Machine A and **machine B** create and record a record of each processed Transfer Request in their respective Mailboxes.

To view the record of a Transfer Requests contained in the Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** sub-node.

In the right pane, Gateway displays all Transfer Request records currently contained in the Mailbox. Each row represents one Transfer Request.

3. In the left pane, double-click the line that represents the Transfer Request you want to review.

Gateway displays the *Transfer Request information* window. Use the tabs of this window to view the characteristics of the Transfer Request.

Using the Log file

By default, Gateway generates traces of network activities that it records to your log file. To view the contents of the log file, in the left pane of the GUI main window, select **Transfer Management > Monitoring > Log**. Gateway displays the current log file contents in the right pane.

To display a help file that provides information about a specific log entry, double-click any line of the log file in the right pane. Gateway opens the online help, displaying information relating to the line you selected.

If you require supplementary trace information concerning CGate processes, you can activate CGate logging. In this example, CGate logging can be useful to monitor the handling of connection access to **machine B**.

On **machine B**, proceed as follows to enable the recording of CGate event information to the log file:

- In the left pane of the Gateway GUI, right-click the icon that represents your Local Axway Gateway and select **Configure**.
- Select **Gateway**.
- Click the **Log levels...** button.
- In **CGate**, select **Full**.
- Click **OK** to confirm the new log level settings.
Gateway immediately takes the new settings into account.

[About PeSIT Hors SIT](#)

[Configuring protocols](#)

[Viewing and managing Mailbox contents](#)

[Working with Logs](#)

Axway Gateway: Transfer Examples

Executing a basic FTP transfer

[About this example](#)

[Prerequisites](#)

[Configuring the client/sender \(machine A\)](#)

[Configuring the server/receiver \(machine B\)](#)

[Executing the FTP transfer to server](#)

[Transfer results](#)

About this example

This topic describes the minimal configuration needed to enable you to execute an FTP Transfer, without certificates or security, between two machines, both hosting Gateways.

To send a file via FTP, both the client and the server need to be configured to recognize one another. In addition, the transfer protocol FTP and its most basic parameters must be defined. The following example uses the minimal number of parameters possible to achieve an FTP transfer. You can use this example as a basis for a more complex transfer.

In this example we use the Gateway GUI to transfer a `.txt` file from **machine A**, the client/sender, to **machine B**, the server/receiver.

Prerequisites

- Gateway must be correctly installed and running on both **machine A** and **machine B**.
- The two machines must be linked by a TCP/IP connection.
- The FTP protocol must be activated on both machines (refer to [Configuring protocols](#)).
- Open a Windows environment to use the GUI.

Configuring the client/sender (machine A)

On **machine A** you must configure a Gateway Remote Site object for this FTP transfer.

Remote Site

On **machine A**, define the **machine B** values in the *Remote Site* window as follows:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
 2. Right-click **Remote Site**, then from the context menu that appears, select **New**.
Gateway displays the *New Remote Site* window.
 3. Complete the fields of the *New Remote Site* window. You do not need to complete fields that are not listed.
- In the [General](#) tab:
 - **Alias:** Enter the Remote Site alias. You can enter up to 31 alphanumeric characters.
 - **Protocol:** From the drop-down list, select **FTP**.
 - **Group name:** Enter a group name for this Site. Keep in mind that the Site must belong to the same group as the Application used for a given Transfer. By default, this value is set to **GDEFAULT**.

- **Role:** Click to select all as enabled.
- **Initialization mode:** Click to select all as enabled.
 - In the [Options](#) tab:
 - **Maximum data block size (in bytes):** Enter **4000**.
 - **Max. no. of simultaneous connections to this Site:** Enter **4**.
 - **Max. no. of simultaneous connections from this Site:** Enter **4**.
 - **Max. no. of connection retries before time limit:** Enter **300**.
 - **Time before first retry (in seconds):** Enter **100 or higher**.
 - **Maximum interval between retries (in seconds):** Enter **1800**.
 - In the [Network address](#) tab:
 - **Communication address:** Select **Main address**.
 - **Network type:** From the drop-down list, select **TCP/IP**.
 - In the Right pane, TCP specific main address tab:
 - **IP address or machine name:** Enter the IP address as <my. domain.name>
 - **Port number:** Enter 6370.
 - In the [FTP/HTTP](#) tab:
 - **ID/Password to send** (*Mandatory*): Enter the identifier and password to send during the connection phase. You can enter up to 25 alphanumeric characters in the **ID** field and 15 alphanumeric characters in the **password** field.
In client mode, Gateway uses the values that you set in these fields for the login phase on a remote server.
 - **ID/Password to check:** Enter the identifier and password used to check partner identity. The identifier corresponds to the protocol identifier.
- 4. After completing all Remote Site tabs, click **OK** to confirm.

Configuring the server/receiver (machine B)

In our example FTP Transfer, **machine B** is the destination and receives a file from **machine A**, the source for the transfer.

On **machine B**, configure the following Gateway objects to enable **machine B**, the server/receiver, to receive a file via an FTP transfer from **machine A**:

- [VFD](#)
- [CGateGroup](#)
- [CGate](#)
- Network Profile. See the Security Guide > [Managing Network profiles](#) for further information (requires login)

VFD

Create a new VFD object on **machine B** as follows:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **VFD** sub-node, then select **New** from the context menu.

Gateway opens the *New Virtual File Directory* window.

3. Complete the fields as follows:
 - **PathName:** Enter **/FTP** if this is not already displayed. When you create a new directory, Gateway automatically completes the path name field with the path of the parent directory. The absolute path of the root directory is **/**.
 - Repeat to create another directory called **/MyCustomers**. The result is **/FTP/MyCustomers**.
 - **Name:** Enter the name of the directory, for example **Customer1**.
4. Click **OK** to confirm.

Later in this example, you will transfer the selected file to **/FTP/MyCustomers/Customer1**.

CGateGroup

Create a new root CGateGroup object for the **machine B** configuration as follows:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup**

Gateway displays the *New CGateGroup* window.

3. In the **General** tab, complete the fields as follows:
 - **Name:** Enter **GROUP FTP** as the unique name for the CGateGroup.
 - **Parent CGateGroup:** Enter **GFTP**.
 - **State:** From the drop-down list select **Enabled**.
 - **Protocol:** From the drop-down list, select **FTP**.
 - **Called address:** Enter ***** as the called TCP/IP address for this CGateGroup.
 - **Called SAP:** Enter ***** as the TCP/IP local connection port.
Alternatively, you can enter the local SAP defined in the Gateway configuration menu. By default, this value is 6370.
 - **Template Site:** From the drop-down list, select **TFTP** as the Template Site that you want Gateway to use to create a dynamic Site.
4. In the **Output parameters** tab, complete the fields as follows:
 - **Root directory:** Enter **/FTP/MyCustomers**.
 - **Home directory:** You do not need to complete this field.

The **Output parameters** tab is identical for both CGateGroup and CGate objects. The fields on the **Output parameters** tab define parameters for the current connection.

5. In the **VFD Rights** tab, complete the fields as follows:
 - **Directory path:** Enter `/` as the Directory path. Regardless of the VFD rights defined for the current CGate, the connected user only has access to the VFD directory tree below the root directory.
 - **File pattern:** Enter `*` in this field. This field further refines the VFD rights to specific file types.
 - **Sub-directory inheritance:** Click to select this box. This indicates that all sub-directories inherit the parameters that you set here.
 - Click to select all of the **Directory rights** boxes: **List**, **Create**, and **Remove**.
 - Click to select all of the **File rights** boxes: **Read**, **Delete**, and **Write**.
6. Click **Add** to confirm.
7. In the **Calling Address** tab, complete the fields as follows:
 - **Calling address:** Enter the wildcard character `*`.
 - **Access allowed:** Click to select. This authorizes access from the calling address.
8. Click **Add** to confirm.

You have created the new CGateGroup object.

CGate

The CGate object is an identification filter used internally in server mode. CGates determine whether or not a client has the right to connect to Gateway and exchange files with it. Create a CGate object for this example transfer as follows:

1. Right-click to select the parent **CGateGroup** folder that you have just created.
2. Select **New** then **CGate** from the context menu to display the *CGate* window.
3. Complete the CGate **General** tab as follows:
 - **Name:** Enter **CG_FTP1**, as a unique identifier for the new object.
 - **Parent CGateGroup:** If it is not already displayed, enter **GFTP**, the name of the parent CGateGroup with which the new object is associated.
 - Complete the **HTTP / FTP / SFTP / PeSIT pre-connection Identification/Checking** fields as follows:

Login user name: Enter the user [login name of machine A](#), the client/sender.

Login password: Enter the password associated with the user login field. Remember that the password is case sensitive.

4. Complete the CGate **Output** tab as follows:
 - **Root directory:** The root directory for the connected user should be displayed automatically.
 - **Home directory:** In this example, enter **/FTP/MyCustomers/Customer1** as the VFD directory on which the first LIST command is processed following protocol connection.
5. Click **OK** to confirm.

Network Profile

A Network Profile associates an incoming network connection request with a Security Profile. The connection does not use Security if you have not selected TLS or SSH.

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Right-click the **Network Profile** sub-node, and from the context menu that appears, select **New**
Gateway displays the *New Network Profile* window.
3. In the *New Network Profile* window, enter the following values:
 - **Network profile:** Enter **NP_FTP**.
 - **Origin address/port:** Enter *.*.*.* as the address or the listening port for **machine A**.
 - **Destination address/port:** Enter *.*.*.* as the address or the listening port for **machine B**.
 - **Network type:** Select **TCP/IP**.
 - **Network security:** Select **None**.
4. Click **OK** to confirm.

Executing the FTP transfer to server

From **machine A** initiate the transfer:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **Transfers** sub-node, and then select **New > Transfer Request** from the context menu.
Gateway displays the *New Transfer Request* window.

3. Complete the Transfer Request [General](#) tab.
 - **Type:** From the drop-down list, select the Transfer type: **TRANS**.
 - **Mode:** From the drop-down list, select the Transfer connection mode **Initiator**.
 - **Direction:** From the drop-down list, select the Transfer direction **Send**.
 - **Application:** Select **FTP_A** as the Application from the drop-down list.
 - **Permanent:** This option allows you to set Transfer availability to permanent. From the drop-down list, select **No**.
 - **Originator alias:** Enter the Local Site alias that is the source of the file transfer. In this example the alias for **machine A**.
 - **Destination alias:** Enter the Site alias that is the file transfer destination. In this example the alias for **machine B**.
4. Complete the Transfer Request [Attributes](#) tab.
 - **File component:** Enter the local file name (without the file path) of the physical file to send.
5. After completing these tabs, click **OK** to confirm.

Machine A contacts **machine B** and begins the file transfer procedure.

Transfer results

After completing the procedure [Executing the FTP transfer to server](#), refresh your screen and check that the transfer was successfully completed. If all of the entered network and application conditions are correct, the transfer is executed and a copy of the file is put on **machine B**.

After a transfer, remember to refresh the screen in both the Mailbox and the Log file.

Using the Mailbox

Machine A and **machine B** create and record a record of each processed Transfer Request in their respective Mailboxes.

To view the record of a Transfer Requests in the Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** sub-node.

In the right pane, Gateway displays all Transfer Request records currently contained in the Mailbox. Each row represents one Transfer Request.

3. In the left pane, double-click the line that represents the Transfer Request you want to review.

Gateway displays the *Transfer Request information* window. Use the tabs of this window to view the characteristics of the Transfer Request.

Using the Log file

By default, Gateway records network activity in a log file. To view the log file, in the left pane of the GUI main window click **Transfer Management > Monitoring > Log**. Gateway displays the current log file contents in the right pane.

To display a help file that provides information about a specific log entry, double-click any line of the log file in the right pane. Gateway opens the online help, displaying information relating to the line you selected.

If you need more trace information about CGate processes, you can activate CGate logging. In this example, CGate logging can be useful to monitor the connection with **machine B**.

On **machine B**, proceed as follows to enable the recording of CGate event information to the log file:

- In the left pane of the Gateway GUI, right-click the icon that represents your Local Gateway and select **Configure**.
- Select **Gateway**.
- Click the **Log levels** button.
- In **CGate**, select **Full**.
- Click **OK** to confirm the new log level settings.
Gateway immediately takes the new settings into account.

[About FTP](#)

[Configuring protocols](#)

[Viewing and managing the Mailbox](#)

[Working with Logs](#)

Axway Gateway: Transfer Examples

Executing an SFTP transfer

[About this example](#)

[Prerequisites](#)

[Configuring the SFTP client/sender \(machine A\)](#)

[Configuring the SFTP server/receiver \(machine B\)](#)

[Executing the SFTP transfer](#)

[Transfer results](#)

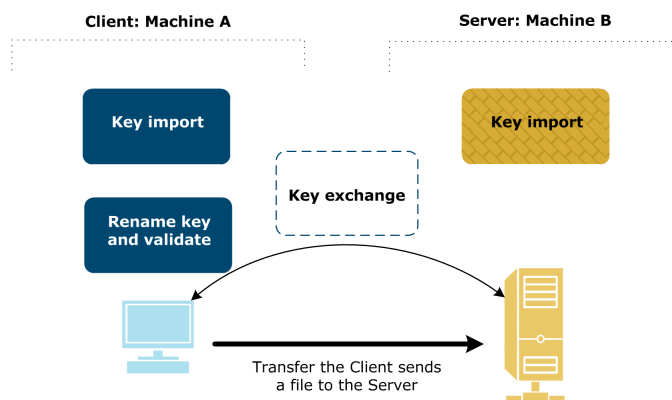
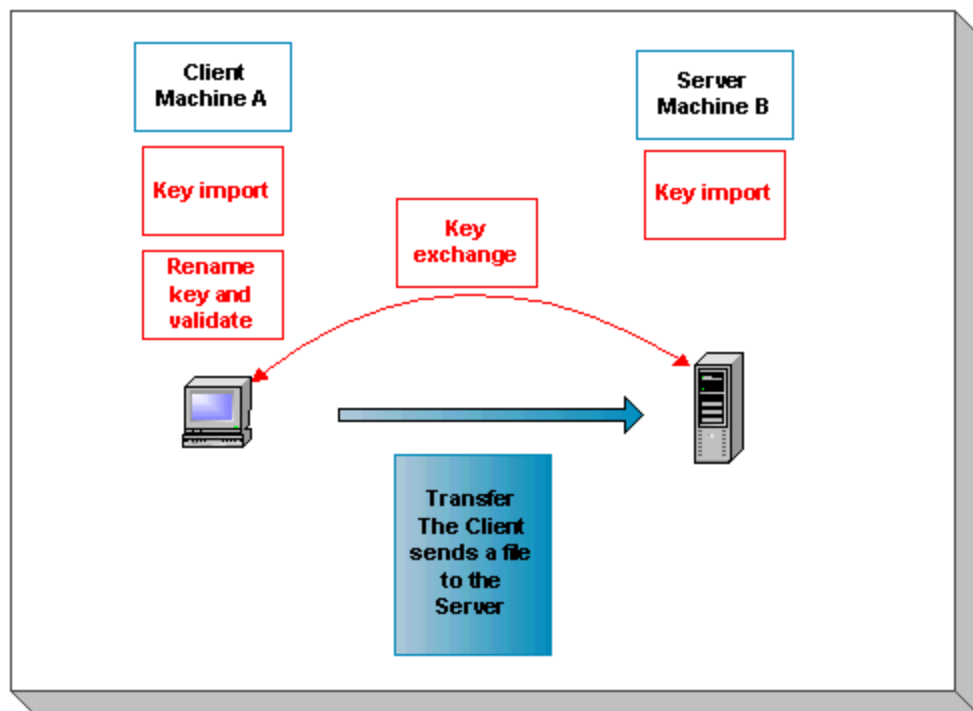
[Validating keys](#)

About this example

This topic describes the minimal configuration needed to enable you to execute an SFTP transfer, that means using security, between two machines, both hosting Gateways.

To send a file via SFTP, both the client and the server need to be configured to recognize one another. In addition, the transfer protocol SFTP and its most basic parameters must be defined. The following example uses the minimal number of parameters possible to achieve an SFTP transfer. You can use this example as a basis for a more complex transfer.

In this example we use the Gateway GUI to transfer a `.txt` file from **machine A**, the client/sender, to **machine B**, the server/receiver.



Prerequisites

- Gateway must be correctly installed and running on both **machine A** and **machine B**.
- The two machines must also be linked by a TCP/IP connection.
- The sending **machine A** and receiving **machine B** must both support the SFTP protocol for the transfer. Additionally, the TCP option must be activated for SFTP.
 - In the configuration menu of both machines, activate the SFTP protocol over TCP. Refer to [Configuring protocols](#).
 - After modifying the configuration, remember to stop and restart Gateway so that the modifications are taken into account.
- Both machines must have the appropriate keys, client or server, for the security portion of the SFTP transfer to be realized. If you have not already done so, import both a Public key and a Private key for the server, and a Private key for the client before setting the SSH parameters.
- This example is executed via a GUI supported by a Windows environment.

Configuring the SFTP client/sender (machine A)

For this SFTP transfer, you need to configure the following objects on **machine A**.

- Keys. See *Security Guide* > [Certificates and keys](#) for further information (requires login)
- SSH. See *Security Guide* > [SSH](#) for further information.
- [Remote Site](#)

Importing keys

On installation, Gateway A provides a set of keys and certificates. You can import additional keys using the GUI.

To import a user key:

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Right-click the **Key** sub-node.
Gateway displays a context menu.
3. Select **Import...** from the context menu.
Gateway displays the *Key import* window.
4. Complete the *Key import* window as follows.
 - **Name:** Enter **SSH_PRV** as the name for the key.
 - **File containing the key:** Enter the name of the file containing the key, or use the Browse button to select the key file.

- **File format:** Select **PEM**.
- **Key Type:** Select **PRIVATE**.
- **Algorithm:** Select **RSA**.

5. Click **OK** to confirm and close the *Key import* window.

SSH Profile window

Use the **SSH Profile** tab to define the SSH profile for **machine A**.

If you are performing this procedure for the first time, you must import a key before completing the SSH Profile window. Validate the key before the starting the SFTP transfer portion of this example.

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management > Security Profile
2. Right-click **SSH Profile** and select **New** from the context menu.
Gateway displays the *New SSH Profile* window.
3. In the *New SSH Profile* window, complete the fields in the **General** tab as follows.
 - **Name:** Enter a unique name for the SSH Profile. In this example use **SSH_CLIENT**.
 - **Profile Type:** From the drop down list, select **CLIENT**.
 - **Key exchange algorithms:** Select **DH_GROUP1_SHA1**.
 - **Encryption algorithms:** Select **AES128_CBC** and **3DES_CBC**.
 - **Mac Algorithm:** Select **HMAC_SHA1**.
 - **Public key:** Select **SSH_RSA**
 - **Client Authentication:** Select **Password or Public Key**.

A pop-up window lets you know that key authentication parameters are added when you modify the algorithms described above. Click **OK** to confirm and close the window.

Hint: The four algorithm buttons must have a value defined to operate in SFTP. Typically the strongest algorithm is listed first. Be sure that for each button there is at least one algorithm in common for both the server and the client.

4. In the *New SSH Profile* window, complete the fields in the **Authentication** tab as recommended below.
 - **Private key alias:** Select **CLIENT_PRV**.
 - **Trust hosted public key:** Select to activate.
 - **Automatic import of partner keys:** Select to activate.
5. Click **OK** to save the new SSH profile and close the window.

Remote Site

On **machine A**, define the **machine B** values in the Remote Site window. Complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node. From the context menu that appears, select **New**
Gateway displays the *New Remote Site* window.
3. Complete the tabs of the *New Remote Site* window as follows. You do not need to complete fields that are not listed.
4. In the **General** tab:
 - **Alias:** Enter the Remote Site alias. You can enter up to 31 alphanumeric characters.
 - **Protocol:** From the drop-down list, select **SFTP**.
 - **Role:** Click to select all **Role** and **Initialization modes**.
5. In the **Network address** tab:
 - **Communication address:** Select **Main address**.
 - **Network type:** From the drop-down list, select **TCP/IP**.
 - In the right pane:
IP address or machine name: Enter the IP address as <my.domain.name>
Port number: For this example enter a port number that is greater than 6320, usually 6321 for SFTP.
6. In the **SFTP** tab:
 - **User name:** Enter **CLIENT_SFTP**.
 - **SSH Profile:** Select **SSH_CLIENT** which was defined earlier in the *SSH profile* window.
 - **Newline convention:** Select **Windows**.

In client mode, Gateway uses the values that you set in these fields for the login phase on a remote server.
7. Click **OK** to confirm and close the window.

Note: If you are performing this procedure for the first time, you must import a User certificate before beginning the server **Machine B** set-up. See the *Security Guide* > [Importing a Certificate](#) for further information (requires login).

Configuring the SFTP server/receiver (machine B)

In this example, **machine B** is the destination and receives a file from **machine A**, the source for the transfer. You need to configure the following objects on **machine B** (the server/receiver) to enable it to receive a file via an SFTP transfer from **machine A**.

- Keys. See the *Security Guide* > [Managing keys](#) for further information (requires login)
- [VFD](#)
- SSH Profile. See *Security Guide* > [SSH Protocol](#) for further information (requires login)
- Network Profile. See *Security Guide* > [Network profiles](#) for further information.
- [CGateGroup](#)
- [CGate](#)

Importing keys

A set of keys and certificates is delivered with Gateway.

Import both a public and private key for the server, using the following procedure:

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Right-click the **Key** sub-node.
Gateway displays a context menu.
3. Select **Import...** from the context menu.
Gateway displays the *Key import* window.
4. To import a public key for the server, complete the *Key Import window* as follows:
 - **Name:** Enter **SVR_PUB** as the name for the key.
 - **File containing the key:** Enter the name of the file containing the key, or use the Browse button to select the key file.
 - **File format:** Select **PEM**.
 - **Key Type:** Select **PUBLIC**.
 - **Algorithm:** Select **RSA**.
5. Click **OK** to confirm and close the *Import key* window.
6. Repeat the *Key Import* process to import a private key as follows. Access the *Import key* window.
 - **Name:** Enter **SVR_PRV** as the name for the key.
 - **File containing the key:** Enter the name of the file containing the key, or use the Browse button to select the key file.

- **File format:** Select **PEM**.
 - **Key Type:** Select **PRV**.
 - **Algorithm:** Select **RSA**.
7. Click **OK** to confirm and close the *Import key* window.

VFD

On **machine B**, create a new VFD object:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **VFD** sub-node, and then select **New** from the context menu.
Gateway opens the *New Virtual File Directory* window.
3. Complete the following field:
 - **PathName:** Enter **/SFTP**. When you create a new directory, Gateway automatically completes the path name field with the path of the parent directory. The absolute path of the root directory is **"/**.
4. Select the newly created VFD folder SFTP and right-click **New**
5. The window path name displays **/SFTP**. Enter the directory name, for example **ClientSFTP**.
6. Click **OK** to confirm and close the window. The new SFTP folder appears under the VFD node, and the ClientSFTP sub-folder is displayed below.

SSH Profile

On **machine B**, define the SSH profile for the server.

If you are performing this procedure for the first time, check that you have imported the server keys before completing the SSH Profile window.

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management > Security Profile
2. Right-click **SSH Profile** and select **New** from the context menu.
Gateway displays the *New SSH Profile* window.
3. Complete the fields in the **General** tab of the *New SSH Profile* window as indicated below.
 - **Name:** Enter a unique name for the SSH Profile, the same name that you used in the previous Network tab. In this example use **SSH_SRV**.
 - **Profile Type:** From the drop down list, select the **SERVER**.
 - **Key exchange algorithms:** Select **DH_GROUP1_SHA1**.
 - **Encryption algorithms:** Select **AES128_CBC** and **3DES_CBC**.

- **Mac Algorithm:** Select **HMAC_SHA1**.
- **Public key:** Select **SSH_RSA**.
- **Client Authentication:** Select **Password or Public Key**.

Hint: For each of the 4 algorithm buttons at least one value must be defined for to operate in SFTP. Typically the strongest algorithm is listed first. Be sure for each button there is at least one algorithm in common for both the server and the client.

4. Complete the fields in the **Authentication** tab as indicated below.
 - **Private key alias:** Select **SRV_PRV**.
5. Click **OK** to confirm your selections and close the *New SSH Profile* window.
6. Gateway displays the new SSH Profile in the right pane work area.

Network Profile

A Network Profile associates an incoming network connection request with a Security Profile. The connection does not use Security if you have not selected TLS or SSH in the configuration menu.

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Right-click the **Network Profile** sub-node, and from the context menu that appears, select **New**
Gateway displays the *New Network Profile* window.
3. In the *New Network Profile* window, enter the following values:
 - **Network profile:** Enter a name for the profile.
 - **Origin address/port:** Enter *.*.*.*/*
 - **Destination address/port:** Enter *.*.*.*/* as the listening port for the **machine B**.
 - **Network type:** Select **TCP/IP**.
 - **Network security:** Select **SSH**.
 - **Security profile:** Select **SSH_SRV**, as defined in the SSH profile.
4. Click **OK** to confirm and close the *New Network Profile* window.

CGateGroup

Create a new root CGateGroup object as follows for the **machine B** configuration.

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup...**
Gateway displays the *New CGateGroup* window.

3. In the *New CGateGroup* window, complete the **General** tab as follows.
 - **Name:** Enter **GROUP_SFTP** as the unique name for the CGateGroup.
 - **State:** From the drop-down list select the state **Enabled**.
 - **Protocol:** From the drop-down list, select **SFTP**.
 - **Called address:** Enter ***** as the called TCP/IP address for this CGateGroup.
 - **Called SAP:** Enter ***** as the TCP/IP local connection port.
Alternatively, you can enter the default port, 6710.
 - **Template Site:** From the drop-down list, select **TSFTP** as the Template Site that you want Gateway to use to create a dynamic Site.
4. Click the **Output Parameters** tab and enter the following values.
 - **Root directory:** Enter **/SFTP/**
 - **Home directory:** Leave blank.

The **Output Parameters** tab is identical for both CGateGroup and CGate objects. The fields on this tab define parameters for the current connection.

5. Click the **VFD Rights** tab and enter the following values.
 - **Directory path:** Enter **/** as the Directory path. Regardless of the VFD rights defined for the current CGate, the connected user only has access to the VFD directory tree below the root directory.
 - **File pattern:** Enter ***** in this field. This field further refines the VFD rights to specific file types.
 - **Sub-directory inheritance:** Click to select this box. This indicates that all sub-directories inherit the parameters that you set here.
 - Click to select all of the **Directory rights** boxes: List, Create, and Remove.
 - Click to select all of the **File rights** boxes: **Read**, **Delete**, and **Write**.
6. Click **Add** to confirm.
7. Click the **Calling Address** tab and enter the following values.
 - **Calling address:** Enter the wildcard character *****.
 - **Access allowed:** Click to select. This authorizes access from the calling address.
8. Click **Add**, then click **OK** to confirm. You have created a new CGateGroup object.

CGate

The CGate object is an identification filter used internally in server mode. CGates determine whether a client has the right to connect to Gateway and exchange files with it or not. Create a CGate object for this example transfer as follows.

1. Right-click to select the parent **CGateGroup** folder that you just created.
2. Select **New** then **CGate** from the context menu to display the *CGate* window.

3. Complete the **General** tab as follows.
 - **Name:** Enter **CG_SFTP**, as a unique identifier for the new object.
 - Complete the **HTTP / FTP / SFTP / PeSIT pre-connection Identification/Checking** fields as follows:
 - **Login user name:** Enter the user login name of the **machine A (Client1)** in this example) the client/sender.
 - **Login password:** Enter the password associated with the user login field. Remember that the password is case sensitive.
 - In the **PHSE** pane enter ***** in all fields.
4. Complete the **Output** tab as follows.
 - **Root directory:** In this example use **/SFTP**, the VFD root directory for the connected user, is automatically displayed. Alternatively, use the **Browse** button to make your selection.
 - **Home directory:** In this example enter **/SFTP/ClientSFTP** as the VFD directory.
5. Complete the **VFD Rights** tab as follows.
 - **File pattern:** Enter ***** for the file pattern.
 - **General information:** Click to select **Sub directory inheritance**.
 - **Directory rights:** Click to select all.
 - Click **Add** to confirm the selections in the **VFD** tab.
6. Click **OK** to confirm and close the *CGate* window. The server, machine B, is configured and ready to execute an SFTP transfer.

Executing the SFTP transfer

From **machine A** initiate the transfer:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **Transfers** sub-node, and then select **New > Transfer Request** from the context menu.

Gateway displays the *New Transfer Request* window.

3. In the *New Transfer Request* window, complete the Transfer Request [General tab](#):
 - **Type:** Select the Transfer type **TRANS**.
 - **Mode:** Select the Transfer connection mode **Initiator**.
 - **Direction:** Select the Transfer direction **Send**.
 - **Application:** Select **DEFAULT_B**.
 - **Destination alias:** Enter the Site alias that is the file transfer destination. In this example the alias for **machine B (SSH server)**.

4. Complete the Transfer Request [Attributes tab](#).
 - **File component:** Enter the local file name (without the file path) of the physical file to send.
5. Click **OK** to confirm. **Machine A** contacts **machine B** and begins the file transfer procedure.

Note: If you are using certificates that are not already validated by both machine A and machine B, the SFTP transfer cannot execute until keys are exchanged between the two machines, and then validated, as described in the topic Certificate validation.

Transfer results

After completing the *Executing an SFTP transfer* procedure, refresh your screen and check that the transfer was successfully completed. If all of the entered network and application conditions are correct, the transfer is executed and a copy of the file is put on **machine B**.

After a transfer, remember to refresh the screen in both the Mailbox and the Log file.

Using the Mailbox

Machine A and **machine B** create and record a record of each processed Transfer Request in their respective Mailboxes.

To view the record of a Transfer Requests in the Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** sub-node.

In the right pane, Gateway displays all Transfer Request records currently contained in the Mailbox. Each row represents one Transfer Request.

3. In the left pane, double-click the line that represents the Transfer Request you want to review.

Gateway displays the *Transfer Request information* window. Use the tabs of this window to view the characteristics of the Transfer Request.

Refer to [Viewing and managing Mailbox contents](#).

Using the Log file

By default, Gateway generates traces of network activities that it records to your log file. To view the log file, in the left pane of the GUI main window click **Transfer Management > Monitoring > Log**. Gateway displays the current log file contents in the right pane.

To display a help file that provides information about a specific log entry, double-click any line of the log file in the right pane. Gateway opens the online help, displaying information relating to the line you selected.

For more advanced debugging you can activate [SFTP specific traces](#).

If you need more trace information about CGate processes, you can activate CGate logging. In this example, CGate logging can be useful to monitor the connection with **machine B**.

On **machine B**, proceed as follows to enable the recording of CGate event information to the log file:

- In the left pane of the Gateway GUI, right-click the icon that represents your Local Gateway and select **Configure**.
- Select **Gateway**.
- Click the **Log levels...** button.
- In **CGate**, select **Full**.
- Click **OK** to confirm the new log level settings.
Gateway immediately takes the new settings into account.

Refer to [Working with Logs](#).

Validating keys

Before a transfer can occur, you must manually verify any keys that are exchanged by either machine involved in the transfer. After verifying the keys, repeat the SFTP transfer procedure.

For a given machine, either the machine A or B in the example, keys are displayed in the right pane window of the GUI. A yellow dot next to the key name identifies a key that has not been validated.

To validate a key:

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Click the **Key** sub-node to select it.
Gateway displays a list of all available keys in the display (right) pane.
3. Click the key name to select the key that you want to validate.
4. Right-click and select **Enable**. The yellow certificate tag becomes green. We recommend that you rename the key at this time.
5. When you have validated keys for both machines A and B, repeat the [SFTP transfer procedure](#).

[About SFTP](#)

[Configuring protocols](#)

[Viewing and managing the Mailbox](#)

[Working with Logs](#)

Axway Gateway: Transfer Examples

Executing an FTP transfer using TLS and certificates

[About this example](#)

[Prerequisites](#)

[Configuring the client/sender \(machine A\)](#)

[Configuring the server/receiver \(machine B\)](#)

[Executing the FTP transfer](#)

[Transfer results](#)

[Validating certificates](#)

About this example

This topic describes the basic configuration that enables you to execute an FTP transfer between two Axway Gateway hosted machines using certificates.

To send a file via FTP, both the client and the server need to be configured to recognize one another. In addition, the transfer protocol FTP and its most basic parameters must be defined in the Gateway configuration menu.

The following example uses the minimal number of parameters possible to achieve an FTP transfer. You can use this example as a basis for a more complex transfer.

In this example we use the Gateway GUI to transfer a `.txt` file from **machine A**, the client/sender, to **machine B**, the server/receiver.

Prerequisites

- Gateway must be correctly installed and running on both **machine A** and **machine B**.
- The two machines must also be linked by a TCP/IP connection.
- Open a windows environment to support the GUI.
- Configure Gateway to support FTP and to require a secure transfer that uses certificates. Refer to [Configuring protocols](#).

Configuring the client/sender (machine A)

For this FTP transfer, configure the following two objects on **machine A**.

- [Remote Site](#)
- TLS Security Profile. See the [Security Guide](#) for further information (requires login)

Remote Site

On **machine A**, define values for remote **machine B** in the Remote Site window. Complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node. From the context menu that appears, select **New**.
Gateway displays the *New Remote Site* window.
3. In the *New Remote Site* window, complete the tabs as follows. You do not need to complete fields that are not listed.
4. In the **General** tab:
 - **Alias:** Enter the Remote Site alias. You can enter up to 31 alphanumeric characters.
 - **Protocol:** From the drop-down list, select **FTP**.
 - **Group name:** Enter a group name for this Site. Keep in mind that the Site must belong to the same group as the Application used for a given Transfer. By default, this value is set to **GDEFAULT**.
 - **Role:** Click to select all as enabled.
 - **Initialization mode:** Click to select all as enabled.
5. In the **Options** tab:
 - **Max. no. of simultaneous connections to this Site:** Enter **100**.
 - **Max. no. of simultaneous connections from this Site:** Enter **100**.
6. In the **Network address** tab:
 - **Communication address:** Select **Main address**.
 - **Network type:** From the drop-down list, select **TCP/IP**.
 - In the Right pane, TCP specific main address tab:
 - **IP address or machine name:** Enter the IP address as <my.domain.name>
 - **Port number:** For this example enter 6370.
7. In the **FTP/HTTP** tab:
 - **ID/Password to send (mandatory):** Enter the identifier and password to send during the connection phase. You can enter up to 25 alphanumeric characters in the ID field and 15 alphanumeric characters in the password field.

In client mode, Gateway uses the values that you set in these fields for the login phase on a remote server.

8. In the [Net Security](#) tab:
 - **Network security:** From the drop-down list, select **TLS**. The following **TLS security** fields are displayed.
 - **Security profile for outgoing connection:** Enter the name of the outgoing TLS Security Profile such as: **SP_FTP_OUT**.
9. When you have completed all required fields, click **OK** to confirm.

TLS Security Profile

Once you have completed the Remote Site set-up, use the **TLS Profile** tab to define the TLS Profile for **machine A**.

If you are performing this procedure for the first time, you must import a User certificate before completing the TLS Profile window. See *Security Guide* > [Importing a Certificate](#) for further information.

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Security Profile
2. Right-click the **TLS Profile** sub-node and select **New** from the context menu.
 Gateway displays the *New TLS Profile* window.
3. In the *New TLS Profile* window, complete the **General** tab fields as follows:
 - **Name:** Enter a unique name for the TLS Profile, the same name that you used in the previous Network tab. In this example use **SP_FTP_OUT**.
 - **Profile Type:** From the drop-down list, select the **CLIENT**.
 - **Client authentication:** Select **TLS_AUT_OPTIONAL**, the most flexible of the authentications, from the drop-down list.
 - **Accepted SSL versions:** Select **TLSV1**.
 - **Accepted cipher suites:** We recommend, for this example, that you select **RSA_WITH_AES_128_CBC_SHA**.
 - **Automatic import of partner certificate:** Click to select and enable this function.
 - **User certificates:** The Select single certificate box should be active. If not, click to select.
 - Enter the **certificate name**. Use the  icon to browse the certificate list if you do not know the name of the certificate. See *Security Guide* > *Managing TLS* > [Managing TLS Security Profiles](#) for further information (requires login).

Note: In this example, you have to validate the Root certificate validation before the FTP can successfully occur. Refer to [Validating certificates](#).
4. In the **Detail** tab of the TLS Profile window leave all default settings.
 You have completed the **machine A**, client/sender, configuration.
5. Click **OK** to confirm your selections.

Configuring the server/receiver (machine B)

In our example FTP Transfer, **machine B** is the destination and receives a file from **machine A**, the source for the transfer. Configure the following objects to enable **machine B**, the server/receiver, to receive a file via an FTP transfer from **machine A**. Define the following for machine B:

- [VFD](#)
- [CGateGroup](#)
- [CGate](#)
- Network Profile. See *Security guide* > [Managing Network Profiles](#) for further information (requires login).
- TLS Profile. See *Security guide* > [SSL and TLS protocols](#) for further information.

VFD

On **machine B**, create a new VFD object:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **VFD** sub-node, and then select **New** from the context menu.
Gateway opens the *New Virtual File Directory* window.
3. Complete the fields of the *New Virtual File Directory* window as follows:
 - **PathName**: Enter **/FTP** if this is not already displayed. When you create a new directory, Gateway automatically completes the path name field with the path of the parent directory. The absolute path of the root directory is **/**. Repeat and create the directory **/MyNewCustomers**.
 - **Name**: Enter the directory name, for example **Customer1**.
4. Click **OK** to confirm.

CGateGroup

On **machine B**, create a new root CGateGroup object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup...**
Gateway displays the *New CGateGroup* window.
3. In the *New CGateGroup* window, complete the **General** tab as follows:
 - **Name**: Enter **GROUP FTP** as the unique name for the CGateGroup.
 - **State**: From the drop-down list, select the state **Enabled**.
 - **Protocol**: From the drop-down list, select **FTP**.

- **Called address:** Enter * as the called TCP/IP address for this CGateGroup.
 - **Called SAP:** Enter * as the TCP/IP local connection port.
(Alternatively, you can enter the default port, 6370)
 - **Template Site:** From the drop-down list, select **TFTP** as the Template Site that you want Gateway to use to create a dynamic Site.
4. Click the **Output Parameters** tab and complete the fields as follows:
- **Root directory:** Enter **/FTP/MyNewCustomers**.
 - **Home directory:** Leave blank.
- The **Output parameters** tab is identical for both CGateGroup and CGate objects. The fields on the **Output parameters** tab define parameters for the current connection.
5. Click the **VFD Rights** tab and complete the fields as follows:
- **Directory path:** Enter / as the Directory path. Regardless of the VFD rights defined for the current CGate, the connected user only has access to the VFD directory tree below the root directory.
 - **File pattern:** Enter * in this field. This field further refines the VFD rights to specific file types.
 - **Sub-directory inheritance:** Click to select this box. This indicates that all sub-directories inherit the parameters that you set here.
 - Click to select all of the **Directory rights** boxes: List, Create, and Remove.
 - Click to select all of the **File rights** boxes: **Read**, **Delete**, and **Write**.
6. Click **Add** to confirm.
7. Click the **Calling Address** tab and complete the fields as follows:
- **Calling address:** Enter the wildcard character *.
 - **Access allowed:** Click to select. This authorizes access from the calling address.
8. Click **Add**, then click **OK** to confirm.
You have created a new CGateGroup object.

CGate

The CGate object is an identification filter used internally in server mode. CGates determine whether a client has the right to connect to Gateway and exchange files with it or not. Create a CGate object for this example transfer:

1. Right-click to select the parent **CGateGroup** folder that you have just created.
2. Select **New** then **CGate** from the context menu to display the *CGate* window.
3. Complete the CGate **General** tab as follows:
 - **Name:** Enter **CG_FTP1**, as a unique identifier for the new object.
 - Complete the **HTTP / FTP / SFTP / PeSIT pre-connection Identification/Checking** fields as follows:

Login user name: Enter the user [login name of the machine A](#) (**Customer1** in this example) the client/sender.

Login password: Enter the password associated with the user login field. Remember that the password is case sensitive.

4. Complete the CGate **Output** tab as follows:
 - **Root directory:** In this example use **/FTP/MyCustomers**, the VFD root directory for the connected user, is automatically displayed.
 - **Home directory:** In this example enter **/FTP/MyCustomers/Customer1** as the VFD directory on which the first LIST command is processed following protocol connection.
5. Click **OK** to confirm.

Network Profile

A Network Profile associates an incoming network connection request with a Security Profile. The connection does not use Security if you have not selected TLS or SSH. Create a Network Profile:

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Right-click the **Network Profile** node, and from the context menu that appears, select **New**
Gateway displays the *New Network Profile* window.
3. In the *New Network Profile* window, enter the following values:
 - **Network profile:** Enter **NP_FTP**.
 - **Origin address/port:** Enter ***.*.*.*** as the address and ***** as the listening port for the **machine A**.
 - **Destination address/port:** Enter ***.*.*.*** as the address and ***** as the listening port for the **machine B**.
Alternatively, enter the default port 6370
 - **Network type:** Select **TCP/IP**.
 - **Network security:** Select **TLS**.
 - **Security profile:** Select **SP_FTP_IN**.
4. Click **OK** to confirm.

TLS Profile window

Once you have completed the Remote Site set-up, use the **TLS Profile** tab to define the TLS Profile for **machine B**.

If you are performing this procedure for the first time, you must import a Root certificate before completing the TLS Profile window. See the *Security Guide* > [Importing a Certificate](#) for further information (requires login).

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Security Profile
2. Right-click the **TLS Profile** sub-node, and then select **New** from the context menu.
Gateway displays the *New TLS Profile* window.
3. Complete the fields in the **General** tab of the *New TLS Profile* window as follows:
 - **Name:** Enter a unique name for the TLS Profile, the same name that you used in the previous Network tab. In this example use **SP_FTP_IN**.
 - **Profile Type:** From the drop down list, select the **SERVER**.
 - **Client authentication:** Select **TLS_AUT_OPTIONAL**, the most flexible of the authentications, from the drop-down list.
 - **Accepted SSL versions:** Select **TLSV1**.
 - **Accepted cipher suites:** For this example, select **RSA_WITH_AES_128_CBC_SHA**.
 - **Automatic import:** Click to select and enable this function.
 - **User certificates:** The **Select single certificate** box should be active. If not, click to select. Note that when you activate the User certificate, you may make certificate validation necessary before the FTP can successfully occur.
4. Enter the certificate name. Use the  icon to browse the certificate list if you do not know the name of the certificate. See the *Security Guide > [Managing Certificates](#)* for further information.
5. In the **Detail** tab of the *TLS Profile* window, leave all default settings.
You have completed the **machine B**, server/receiver, configuration.
6. Click **OK** to confirm your selections.
Note: In this example, you have to validate the User certificate before the FTP transfer can successfully occur. Refer to [Validating certificates](#).

Executing the FTP transfer

From **machine A** initiate the transfer:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **Transfers** sub-node, and then select **New > Transfer Request** from the context menu.
Gateway displays the *New Transfer Request* window.
3. Complete the Transfer Request [General tab](#).
 - **Type:** From the drop-down list, select the Transfer type **TRANS**.
 - **Mode:** From the drop-down list, select the Transfer connection mode **Initiator**.
 - **Direction:** From the drop-down list, select the Transfer direction **Send**.

- **Application:** Select **FTP_A** as the Application from the drop-down list.
 - **Destination alias:** Enter the Site alias that is the file transfer destination. In this example the alias for **machine B**.
4. Complete the Transfer Request [Attributes tab](#).
 - **File component:** Enter the local file name (without the file path) of the physical file to send.
 5. Click **OK** to confirm after completing these tabs.
Machine A contacts **machine B** and begins the file transfer procedure.

Note: If you are using certificates that are not already validated by both machine A and machine B, the FTP transfer cannot execute until certificates are exchanged between the two machines, and then validated, as described in [Validating certificates](#).

Transfer results

After completing the procedure [Executing the FTP transfer](#), refresh your screen and check that the transfer was successfully completed. If all of the entered network and application parameters are correct, Gateway executes the transfer and deposits the file on **machine B**.

After a transfer, remember to refresh the screen in both the Mailbox and the Log file.

Using the Mailbox

Machine A and **machine B** create and record a record of each processed Transfer Request in their respective Mailboxes.

To view the record of a Transfer Requests in the Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** node.

In the right pane, Gateway displays all Transfer Request records currently contained in the Mailbox. Each row represents one Transfer Request.

3. In the left pane, double-click the line that represents the Transfer Request you want to review.

Gateway displays the *Transfer Request information* window. Use the tabs of this window to view the characteristics of the Transfer Request.

Refer to [Viewing and managing Mailbox contents](#).

Using the Log file

By default, Gateway generates traces of network activities that it records to your log file. To view the log file, in the left pane of the GUI main window click **Log**. Gateway displays the current log file contents in the right pane.

To display a help file that provides information about a specific log entry, double-click any line of the log file in the right pane. Gateway opens the online help, displaying information relating to the line you selected.

If you need more trace information about CGate processes, you can activate CGate logging. In this example, CGate logging can be useful to monitor the connection with **machine B**.

On **machine B**, proceed as follows to enable the recording of CGate event information to the log file:

- In the left pane of the Gateway GUI, right-click the icon that represents your Local Gateway and select **Configure**.
- Select **Gateway**.
- Click the **Log levels...** button.
- In **CGate**, select **Full**.
- Click **OK** to confirm the new log level settings.
Gateway immediately takes the new settings into account.

Refer to [Working with Logs](#).

Validating certificates

Before a transfer can occur, you must manually verify any certificates that are exchanged by either machine involved in the transfer. After verifying the certificates, repeat the FTP transfer procedure.

For a given machine, either the machine A or B in the example, certificates are displayed in the right pane window of the GUI. A yellow dot identifies new certificates that have not been validated.

Before you can successfully execute the example FTP transfer, you must validate the certificates received by both machine A and B.

See the *Security Guide* > [Managing certificates](#) for further information (requires login).

To validate a certificate:

1. In the left pane of the GUI main window, click  to expand the nodes:
Security Management > Transfer Security Management
2. Click the **Certificate** sub-node to select it.
Gateway displays a list of all available certificates in the display (right) pane.
3. Click the name of the certificate you want to validate.
4. Right-click the name of the certificate you want to validate, and then select **Enable** from the context menu. The yellow certificate tag becomes green.
5. When you have validated certificates for both machine A and machine B, repeat the procedure [Executing the FTP transfer](#).

[About FTP](#)

[Configuring protocols](#)

[Viewing and managing the Mailbox](#)

Executing an EDIINT AS2 transfer

[About this example](#)

[Before you execute this example \(AS2 transfer prerequisites\)](#)

[Configuring the EDIINT *distributor* machine](#)

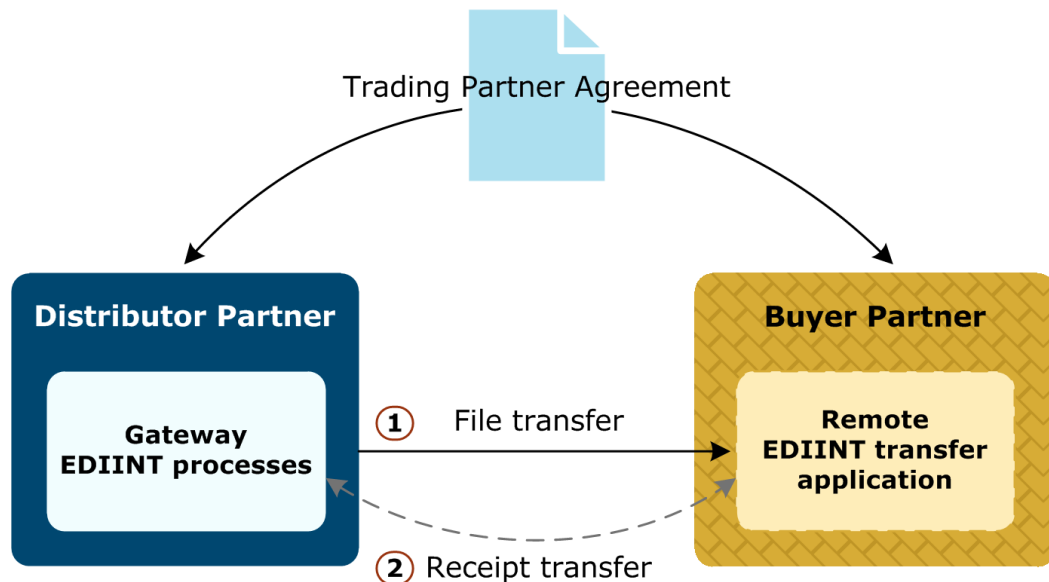
[Submitting an AS2 Transfer Request](#)

[After the transfer](#)

About this example

This topic provides an example for transferring files via the EDIINT AS2 protocol based on an example Trading Partner Agreement. In the context of your daily operations you can adapt this example to your specific transfer requirements.

In this example we use the Gateway GUI to transfer a `.txt` file from the local machine of a commercial *distributor* entity to the remote machine of a *buyer* entity. The receiving remote *buyer* entity responds to the file transfer by returning a receipt.



Transfer overview

To initiate a file transfer towards a business partner using EDIINT AS2:

- [Establish a Trading Partner Agreement](#) with the remote business partner with whom you want to exchange the file
- Confirm that the EDIINT AS2 processes are activated in the Gateway configuration menu. Refer to [Configuring Protocols](#).
- [Configure the sender Axway Gateway to recognize the remote receiving partner](#) by creating the appropriate Site object
- [Create the Trading Partner objects](#) that describe both the local and remote partner EDIINT security characteristics
- [Set the authentication method](#)
- [Create the local file directories](#) for the deposit of files transferred via HTTP
- [Create the CGateGroup and CGate objects](#) to control remote access to your local machine
- [Submit an AS2-type Transfer Request](#)

The example in this topic uses a set of parameters to achieve an EDIINT AS2 transfer based on an example Trading Partner Agreement. You can use this example as a model for executing a transfer based on a similar type of agreement.

Establishing the Trading Partner Agreement

In any EDIINT exchange, the exchange partners need to have prior agreement on certain exchange parameters. This agreement is formalized in a Trading Partner Agreement.

For this example, the Trading Partner Agreement establishes that the following parameters are to be used between the buyer and distributor partners:

- **AS version:**
 - AS2 over HTTP
- **EDIINT partner names:**
 - Distributor
 - Buyer
- **Certificates:**
 - The partners exchange the certificates they will use for signing and encryption. Because they have selected dual certificate use (separate certificates for signing and encryption), each partner will possess four certificates. Each certificate includes the associated public key. Each partner will possess the private keys for two of these certificates.
- **Security services:**
 - signature: yes
 - encryption: yes
 - compression: yes
- **Algorithms:**
 - The partners indicate any algorithm restrictions.

- **Receipts**
 - required: yes
 - signed: yes
 - synchronous: yes
 - protocol: HTTP

Before you execute this example

- Install and start Gateway on your local machine
- Confirm that you are linked to your business partner by a TCP/IP connection
- Open a Windows environment to support the GUI
- Configure Gateway to activate support for the EDIINT protocol AS2 format. Refer to [Configuring protocols](#).
- Confirm that you and your exchange partner own the necessary security certificates and keys and agree on the processes associated with exchanges

Configuring the EDIINT *distributor* machine

For this AS2 transfer example, your local machine is the *distributor* machine. The *distributor* machine initiates sending of the file you want to transfer, and accepts the return emission of the EDIINT receipt. To enable these operations you configure the following Gateway objects on the distributor machine:

- [Remote Site](#)
- [Remote Trading Partner](#)
- [Local Trading Partner](#)
- [VFDs](#)
- [CGateGroup](#)
- [CGate](#)

Additionally you need to [change the default authentication method](#) in the configuration menu.

Remote Site object

In Gateway, on the distributor machine, define values for the machine of the *buyer* partner that is the target of the file transfer. Use the Remote Site object.

To create a Remote Site object that describes the *buyer* partner of the EDIINT AS2 transfer, complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites

2. Right-click the **Remote Site** sub-node. From the context menu that appears, select **New**. Gateway displays the *New Remote Site window*.
3. Complete the **Remote Site** tabs as follows. You do not need to complete fields that are not listed.
 - In the **General** tab:
 - **Alias:** Enter *ALIAS_BUYER*
 - **Protocol:** From the drop-down list, select **AS2**
 - **Protocol identifier:** Enter *BUYER*
 - **Role:** Click all check boxes to select all roles
 - **Initialization mode:** Click all check boxes to select all initialization modes
 - In the **FTP/HTTP** tab:
 - **acknowledgment option:** Select *By monitor*
 - In the **Network address** tab:
 - **Communication address:** Select **Main address**.
 - **Network type:** From the drop-down list, select **TCP/IP**.
 - In the Right pane, TCP specific main address tab:
 - **IP address or machine name:** For this example we use the imaginary address `<ADDRESS_HOST_CLIENT>`
 - **Port number:** For this example we use port **8080**.
4. Click **OK** to confirm and close the window.

Remote Trading Partner object

The Remote Trading Partner object provides Gateway with information about the EDIINT compression, signature, encryption and encapsulation operations to be used for the transfer.

On your local *distributor* machine, in Gateway, define values for the remote *buyer* Trading Partner.

To create a Remote Trading Partner object for the EDIINT AS2 transfer, complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node. From the context menu that appears, select **New > AS2 partner**. Gateway displays the *Trading Partner window*.
3. Complete the following fields of the *Trading Partner window*.
 - In the *Identification criteria* section:
 - **Local Partner:** Confirm that this check box is *not* selected
 - **Name:** Enter a name for the remote partner, *BUYER*
 - **Format:** **AS2** is displayed

- **Alias:** Enter an alias for the remote partner, *ALIAS_BUYER*
- In the *Certificate/Key* section:
 - **Use dual certificates:** Click to select dual certificate use
 - **Signing certificate: Type:** From the drop-down list, select *SEC_ALIAS*
 - **Signing certificate: Name:** Enter the name of the local certificate you are using to check the signature the remote partner: For example *Buyer_cert_sign*
 - **Encryption certificate: Type:** From the drop-down list, select *SEC_ALIAS*
 - **Encryption certificate: Name:** Enter the name of the local certificate you are using to encrypt information sent to the remote partner. For this example, enter *Buyer_cert_encrypt*
 - **Signing algo:** From the drop-down list, select *SHA1withRSA*

For details about the certificates and associated keys necessary for EDIINT transfers, refer to [About EDIINT - Certificates](#).

- In the *Requested Services* section:
 - **Encrypt file:** Click to select
 - **Sign file:** Click to select
 - **Compress file:** Click to select
 - **Request receipt:** From the drop-down list, select *Signed*
 - **Synchronize receipt:** Click to select
Refer to [About EDIINT - Synchronous and asynchronous receipt processes](#).
 - **Receipt URL/email:** Enter the address at which you want to receive the return receipt. For example: *http://ADDRESS_HOST_DISTRIBUTOR/client/transfer?user=buyer&password=buyer*
- In the *Parameters for file reception* section:
 - **Force signed:** Click to select
 - **Force encrypted:** Click to select
 - **Force receipt requested:** From the drop-down list, select *Signed*
 - **Force receipt URL/email:** Enter the address where you want to receive the return receipt. For example: *http://ADDRESS_HOST_DISTRIBUTOR/client/transfer?user=buyer&password=buyer*

4. Click **OK** to confirm and close the window.

Local Trading Partner object

The Local Trading Partner object contains information about the local EDIINT compression, signature, encryption and encapsulation operations that are used for the transfer. It also enables you to specify how to handle receipts returned by the remote *buyer* partner.

To create a Local Trading Partner object for the EDIINT AS2 transfer, complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node. From the context menu that appears, select **New > AS2 partner**.

Gateway displays the *Trading Partner* window.

3. Complete the following fields of the *Trading Partner* window.
 - In the *Identification criteria* section:
 - **Local Partner:** Click to select
 - **Name:** Enter a name for the remote partner, *DISTRIBUTOR*
 - **Format:** **AS2** is displayed
 - **Alias:** Enter an alias for the remote partner, *ALIAS_DISTRIBUTOR*
 - In the *Certificate/Key* section:
 - **Use dual certificates:** Click to select
 - **Signing certificate: Type:** From the drop-down list, select *SEC_ALIAS*
 - **Signing certificate: Name:** Enter the certificate name, for this example *DistributorSignCert*
Note: You must possess the private key for this certificate.
 - **Signing certificate: Key password:** Enter the password for the certificate key, for this example *DistribSnCert*
 - **Encryption certificate: Type:** From the drop-down list, select *SEC_ALIAS*
 - **Encryption certificate: Name:** Enter the certificate name, for this example *DistributorEnCert*
Note: You must possess the private key for this certificate.
 - **Encryption certificate: Key password:** Enter the password for the certificate key, for this example *DistribEnCert*
 - **Signing algo:** From the drop-down list, select *SHA1withRSA*

For details about the certificates and associated keys necessary for EDIINT transfers, refer to [About EDIINT - Certificates](#).

4. Click **OK** to confirm and close the window.

Setting the authentication method

The Gateway default authentication method for server identification is unsuitable for this example. In the configuration menu, reset the authentication method to *CGI user password*:

1. In the left pane of the GUI main window, right-click the Gateway server that you want to configure and select **Configure** from the context menu.
Gateway displays the *Configuration* window in the right pane.
2. Select **Connectivity > Internet protocols > EDIInt > AS2**.
3. Click **Options...**.
Gateway displays the *HTTP options* window.
4. In the **Server identification / Method** field, select *CGI user password*.
5. Click **OK** to confirm and close the *HTTP options* window.
6. Click **OK** to close the *Configuration* window.

VFD

To create a new VFD object for the reception of receipts from the *buyer*.

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **VFD** sub-node, then select **New** from the context menu.
Gateway opens the *New Virtual File Directory* window.
3. Complete the fields as follows:
 - **PathName**: Enter */http_as2/client/*
 - Repeat to create another directory called */receipts*. The result is */http_as2/client/receipts*.
4. Click **OK** to confirm.

CGateGroup object

The CGateGroup object enables you to control protocol characteristics of connection attempts to your local machine. To receive the return EDIINT receipt from the *buyer* entity, create a CGateGroup entity that enables *buyer* to connect to your machine.

To create a CGateGroup object for the EDIINT AS2 transfer, complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup**.
Gateway displays the *New CGateGroup* window.

3. Complete the following fields of the *New CGateGroup* window.
 - In the **General** tab:
 - **Name:** Enter the name for the CGateGroup, *GR_AS2*
 - **State:** From the drop-down list, select *Enabled*
 - **Protocol:** From the drop-down list, select *AS2*
 - **Called SAP:** Enter the universal value *"*"*
 - **Template Site:** From the drop-down list, select *AS2*
4. Click **OK** to validate your values and save the object.

CGate object

The CGate object enables you to impose password control on connection attempts to your local machine. It also enables you to define VFD access rights.

To receive the return EDIINT receipt from the *buyer* entity, create a CGate object that enables *buyer* to connect to your machine via a password, and access restricted local directories.

To create a CGate object for the EDIINT AS2 transfer, complete the fields in the tabs as follows. Do not modify fields that are not listed.

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click  to the left of CGateGroup to display the list of all existing CGateGroups.
3. From the list, right-click the CGateGroup name **GR_AS2** that you created in the [previous section](#). Gateway displays a context menu.
4. From the context menu, select **New CGate** to display the *New CGate* window. Complete the following fields of the *New CGate* window.
 - In the **General** tab:
 - **Name:** enter the name for the CGate, *as2_buyer*
 - **Parent CGateGroup:** enter the name of the CGateGroup that you created in the previous procedure, *GR_AS2*
 - **State:** From the drop-down list, select *Enabled*
 - **Login user name:** Enter *buyer*
 - **Login password:** Enter the universal value *buyer*
 - In the **Output parameters** tab:
 - **Root directory:** Enter */http_as2*
 - **Home directory:** enter */http_as2/buyer*
 - In the **VFD Rights** tab:
 - **VFD rights - directory path:** enter */http_as2:*:R:L:Y ; /http_as2/buyer:*:DRW:L:CR:Y ; /http_as2/buyer/receipts:*:DWR:L:RC:Y*
5. Click **OK** to validate your values and save the object.

Submitting an AS2 Transfer Request

When you have created your Gateway objects as described in the above sections, you are ready to execute the EDIINT transfer. To execute a transfer, submit a Transfer Request as follows.

From Gateway on your local machine (*distributor*) initiate the transfer:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Right-click the **Transfers** sub-node, and then select **New > Transfer Request** from the context menu.

Gateway displays the *New Transfer Request* window.

3. In the *New Transfer Request* window, complete the Transfer Request [General tab](#):
 - **Type:** Select the Transfer type **TRANS**.
 - **Mode:** Select the Transfer connection mode **Initiator**.
 - **Direction:** Select the Transfer direction **Send**.
 - **Destination alias:** Enter the Site alias that is the file transfer destination. For this example we enter the alias defined in the [Remote Site object](#) above, *ALIAS_BUYER*.
4. Complete the Transfer Request [Attributes tab](#).
 - **File component:** Enter the local file name (without the file path) of the physical file to send.
5. Complete the Transfer Request [Trading Partner tab](#).
 - **Destination partner:** Enter *Buyer*
 - **Originator partner:** Enter *Distributor*
 - **Format:** AS2 is displayed
 - **Is signing:** From the drop-down list, select *Yes*
 - **Is compressing:** From the drop-down list, select *No*
 - **Is encrypting:** From the drop-down list, select *Yes*
 - **Receipt request:** From the drop-down list, select *Signed*
 - **Synchronize receipt:** From the drop-down list, select *Yes*
 - **Receipt URL/email:** Enter the address. For example: `ADDRESS_HOST_DISTRIBUTOR/client/transfer?user=buyer&password=buyer`
6. Click **OK** to confirm.

The origination machine contacts the target machine and begins the file transfer procedure.

The machine that is the target of the transfer processes the transfer, and generates and returns a receipt to the originating machine.

Note: A subject can be set in the "email subject" field from the Transfer Request SMTP tab.

After the transfer

Using the Mailbox

For each AS2 transfer that you send, Gateway records two transfer records in the Mailbox:

- One transfer record containing details of the transfers to the remote partner
- One transfer record containing details of the reception of the receipt from the remote partner

To view the details contained in transfer records:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** node.
Gateway displays all currently stored Mailbox records in a table in the display pane. By default, Gateway displays a set of columns containing information about each transfer. The table columns contain the following information for each recorded Transfer (if the data exists for that transfer).
3. Right-click the corresponding record in the Mailbox to view details of a transfer.
Each record contains eight tabs displaying details of different aspects of the transfer process. The Trading Partners tab contains information pertaining particularly to AS2 transfers.

Using the Log file

By default, Gateway generates traces of network activities that it records to your log file. To view the log file, in the left pane of the GUI main window click **Transfer Management > Monitoring > Log**. Gateway displays the current log file contents in the right pane.

To display a help file that provides information about a specific log entry, double-click any line of the log file in the right pane. Gateway opens the online help, displaying information relating to the line you selected.

If you need more trace information about CGate processes, you can activate CGate logging. In this example, CGate logging can be useful to monitor the connection initiated by the remote machine.

Proceed as follows to enable the recording of CGate event information to the log file:

- In the left pane of the Gateway GUI, right-click the icon that represents your Local Axway Gateway and select **Configure**.
- Select **Gateway**.
- Click the **Log levels...** button.
- In **CGate**, select **Full**.
- Click **OK** to confirm the new log level settings.
Gateway immediately takes the new settings into account.

Managing partners

An exchange of files on a network involves at least two exchange partners: a sending partner and one or more receiving partners. The following topics describe the Gateway objects used to define these partners.

[Sites](#)

[Proxies](#)

[CGates and CGateGroups](#)

[Diffusion Lists](#)

[Trading Partners](#)

Axway Gateway: Managing Partners

Managing site objects

About Site objects

Overview

This topic introduces Site objects and explains the difference between static Sites (local and remote) and dynamic (template) Sites.

In Gateway, to define the parameters of a file transfer, you require a description of the partner for the file exchange. You specify the characteristics of partner sites in *Site objects*.

A Site object specifies either the destination or the origin of a file transfer. Each Site:

- Is specific to a given transfer protocol
- Can communicate across multiple types of network: TCP/IP, LU 6.2,...

A transfer begins with the physical connection of two Sites. One Site is the file Sender and one is the Receiver. In Gateway, the terms *Initiator* and *Responder* are used in place of the standard terms Client and Server. The Initiator Site requests the connection, and the Responder Site responds to the connection request. The Initiator and Responder can both be Sender and Receiver. During the connection phase, the Initiator sends its identification. If the Responder is an Axway MFT product, it checks the Initiator identification to see if they correspond to elements specified in its Remote Site attributes.

Static Sites

A static Site is a Site that is defined as an object and stored in the Gateway database. FTP, SFTP and HTTP server products, Axway Transfer products, Axway EndPoint, and Gateway itself are all examples of static Sites.

A static Site in Gateway is either:

- A **Local Site** - an exchange Site located on the Gateway system, or
- A **Remote Site** - a partner Site defined as a Site object on another machine

A Local Site is associated with a Remote Site via the Remote Site parameter: **Partner local alias** (-partner_loc_alias).

To define a Site, you specify the following parameters:

- Name
- Authorized mode: sender/receiver, Initiator/Responder
- Transfer protocol (PeSIT, FTP, HTTP, and so on)
- Communication type (TCP/IP, LU 6.2) and network address for main and backup paths

Dynamic Sites

Template Site definition

To exchange files in client mode, Gateway requires a complete description (including network parameters such as IP address and SAP) of the connection partner and the file exchange process.

When multiple client stations exist, the task of redefining a new Remote Site for each and every file exchange partner may become long and time-consuming. To overcome this, Gateway provides a dynamic Site creation feature that enables you to accelerate the Site definition process and exchange files with unrecognized client stations.

When you submit a Transfer Request for which no defined Remote Site exists, Gateway dynamically creates a Site that is based on one of the Template Sites provided with the product. A Template Site is like any Remote Site, but some of its protocol and network attributes remain empty.

Gateway automatically creates the following Template Sites when you install the product:

- TFTP (for generic access from FTP clients)
- THTTP (for generic access from HTTP clients)
- TFTP_HTTP (special bi-protocol Template Site for generic access from clients that use either HTTP or FTP)
- TSFTP (for generic access from SFTP clients)
- TODT (for generic access from OFTP clients)
- TPEL (for generic access from PEL clients)

- TPHSD (for generic access from PeSIT HS D clients)
- TPHSE (for generic access from PeSIT HS E clients, such as Axway EndPoint)
- TAS2 (for generic access from EDIINT AS2 clients)
- TAS3 (for generic access from EDIINT AS3 clients)
- TRN_HTTP (for generic access from RosettaNet HTTP clients)

Dynamic Site creation

You can associate each client workstation with a physical profile. At the time of connection, the client workstation inherits the attributes of the Template Site.

At the protocol connection level, Gateway analyzes the CGateGroup definition and checks whether a Template Site is specified. If the connection requires the use of a Template Site, Gateway:

- Dynamically creates a Remote Site from the Template Site definition
- Completes the Site definition with the received connection parameters

The lifetime of a dynamic Site lasts as long as a reference to the Site is contained in a Mailbox record of a transfer. When you delete all Mailbox records containing a reference to the dynamic Site, Gateway automatically removes the dynamic Site.

The dynamic Site alias is a concatenation of the following strings:

- "\$": indicates that the current Remote Site is a dynamic Site
- Protocol prefix: current connection protocol prefix which can take the following values:
 - **D**: PeSIT HS D connections
 - **E**: PeSIT HS E connections
 - **P**: PEL connections
 - **F**: FTP connections
 - **H**: HTTP connections
 - **T**: bi-protocol (HTTP & FTP) connections
 - **S**: SFTP connections
 - **O**: OFTP (Odette) connections
 - " _ " : separator character
 - "*connected client identifier*" protocol-dependent string:
 - PHSE, PEL, OFTP: client identifier sent by the connected client
 - HTTP, FTP and bi-protocol (HTTP & FTP): user login sent by the connected client

[Overview: Transfer Sites](#)

[Working with Local Sites](#)

[Working with Remote Sites](#)

[Address configuration for Remote Sites](#)

[Working with Local Sites \(command line\)](#)

[Working with Remote Sites \(command line\)](#)

[Site objects: Parameters List](#)

Axway Gateway: Managing Partners

Working with Local Sites

[Creating a Local Site object](#)

[Displaying a list of Local Site objects](#)

[Displaying selected Local Site objects](#)

[Viewing Local Site attributes](#)

[Modifying Local Site attributes](#)

[Deleting a Local Site](#)

Creating a Local Site object

Gateway always contains at least one Local Site object that it generates from the configuration parameters you entered during the installation. You cannot remove this default Local Site object. You can, however, modify its parameters.

You can create as many additional Local Sites as you require.

To create a new Local Site object:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click the **Local Site** sub-node, then select **New** from the context menu to display the *New Local Site* window.
3. In this window, define the characteristics of the Site object by entering values in the fields, as described in the table below.
4. After completing the fields, click **OK** to accept the values and close the window.

Gateway creates the new Local Site object and enters a new line representing the object in the display pane.

New Local Site window

Field	Description
General information	

Field	Description
Alias	Enter the Local Site alias. Maximum: 31 alphanumeric characters.
Protocol identifier	Enter the Site protocol name. Maximum: 25 alphanumeric characters. By default, this value is the same as the Site alias name.
OFTP	Select this option if the protocol type is OFTP (Odette).
Comments	<i>Optional</i> Enter a free-text description for the object. Maximum: 80 alphanumeric characters.
Default email address	<i>SMTP only</i> Enter the default email address (From) for the Local Site. Maximum: 128 characters. When Gateway is acting as an SMTP client, it must indicate the From address of the email sender. If you do not specify the From address in the Transfer Request, Gateway searches for the address in the Local Site associated with the Remote Site that sends the Request. A Remote Site and a Local Site are always associated with a Request that you send. Gateway identifies the Local Site either via the Remote Site parameter Partner local alias , or via the Local site name defined in the Gateway configuration menu.
Directory for received files	Enter the name of the directory used to store received files. Maximum: 127 alphanumeric characters.
User parameters	
Param 1	User parameter. Maximum: 40 alphanumeric characters.
Param 2	User parameter. Maximum: 40 alphanumeric characters.
Authentication certificate (OFTP R2.0 only)	
Type	Specify the type of authentication certificate to use: • SEC_ALIAS • Passport PS entity • Passport PS certificate
Name	Specify the authentication certificate to use.
Password	<i>Optional</i> Enter the connection password to send (from the Initiator) and the connection password to check (received from the Responder).

Displaying a list of Local Site objects

To display a list of all Local Site objects stored in the database:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Local Site** sub-node.

In the right pane, Gateway displays a list of all Local Sites.

Displaying selected Local Site objects

If your Gateway configuration includes multiple Local Site objects, you can limit the display by applying display selection criteria.

To limit the Local Site object display:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the Local Site sub-node.

Gateway displays a list of all Local Sites in the right pane.

3. Right-click the **Local Site** sub-node, then select **Select...** from the context menu.

Gateway displays the *Select Local Sites* window.

4. Enter either the name of a Local Site and/or a Local Site protocol identifier as selection criteria. Then click **OK**.

Gateway regenerates the right window display list, taking into account your selection criteria.

Viewing Local Site attributes

To view the attributes of a specific Local Site:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Local Site** sub-node.

Gateway displays a list of all Local Sites in the right pane.

3. In the right pane, right-click the name of the Local Site for which you want to view the Site attributes. Select **View...** from the context menu.

Gateway displays the *Local Sites (Read only)* window. This window provides you with a view of all attributes for the selected Local Site.

4. When you have finished viewing the information, click **Close** to quit the window.

Modifying Local Site attributes

To modify the attributes of a Local Site:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Local Site** sub-node.
Gateway displays a list of all Local Sites in the right pane.
3. In the right pane, right-click the name of the Local Site for which you want to modify the Site attributes. Select **Modify...** from the context menu.
Gateway displays the *Local Site* editing window. This window comprises the same fields as the [New Local Site](#) window above.
4. Enter new values as required.
5. After modifying the attributes, click **Close** to accept the new values and close the window.

Deleting a Local Site

To delete an existing Local Site:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Local Site** sub-node.
Gateway displays a list of all Local Sites in the right pane.
3. In the right pane, right-click the name of the Local Site you want to delete. Select **Delete...** from the context menu.
Gateway displays a confirmation dialog box.
4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
Alternatively, click **No** or **Cancel** to abandon the operation.
Gateway deletes the Local Site from the database and removes the entry from the display window.

[Site objects: Parameters List](#)

[About Site objects](#)

[Working with Local Sites \(command line\)](#)

Axway Gateway: Managing Partners

Remote Sites

[About Remote Sites](#)

[Creating a Remote Site](#)

[Displaying a list of Remote Sites](#)

[Displaying selected Remote Sites](#)

[Viewing Remote Site attributes](#)

[Modifying a Remote Site](#)

[Deleting a Remote Site](#)

About Remote Sites

When you install Gateway, it creates a set of Remote Sites adapted for use with different protocols, and stores them in your database. The following sections describe how to add your own Sites and how to manage all of the available Sites.

Creating a Remote Site

To create a new Remote Site object:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Right-click the **Remote Site** sub-node and select **New**

Gateway displays the *New Remote Site* window.

The set of fields and tabs displayed for Remote Site objects depends on the protocol that you select. The initial screen displays three tabs:

- **General**
- **Options**
- **Network address**

When you select the protocol on the **General** tab, the tabs displayed will change.

3. Complete the tabs of the *New Remote Site* window according to your requirements.

The following topics describe the fields in each tab and their possible values:

- [General tab](#)
- [Options tab](#)
- [Network address tab](#)
- Protocol-specific tabs:

- [PeSIT tab](#)
 - [OFTP tab](#) (Odette)
 - [FTP/HTTP tab](#)
 - [PEL tab](#)
 - [POP3 tab](#)
 - [SMTP tab](#)
 - [SFTP tab](#)
 - [SWIFTNet tab](#)
 - [JMS tab](#)
 - [X.400 tab](#)
 - [Net security tab](#)
4. After completing the fields, click **OK** to accept the values and close the window.

Displaying a list of Remote Sites

To display a list of all Remote Site objects stored in the database:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
In the right pane, Gateway displays a list of all Remote Sites.

Displaying selected Remote Sites

If your Gateway configuration includes multiple Remote Site objects, you can limit the display by applying display selection criteria.

To limit the Remote Site object display:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
Gateway displays a list of all Remote Sites in the right pane.
3. Right-click the **Remote Site** sub-node, then select **Select...** from the context menu.
Gateway displays the *Select Remote Sites* window.
4. In the *Select Remote Sites* window, enter one or more of the following Remote Site attributes as selection criteria:
 - Remote Site Alias
 - Protocol

- Network Type
- Network address

5. Click **OK**.

Gateway regenerates the right window display list, taking into account your selection criteria.

Viewing Remote Site attributes

To view the attributes of a specific Remote Site:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Remote Site** sub-node.

Gateway displays a list of all Remote Sites in the right pane.

3. In the right pane, right-click the name of the Remote Site for which you want to view the Site attributes. Select **View...** from the context menu.

Gateway displays the *Remote Sites (read only)* window. This window provides you with a view of all attributes for the selected Remote Site.

4. When you have finished viewing the information, click **Close** to quit the window.

Modifying a Remote Site

To modify the attributes of a Remote Site:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Remote Site** sub-node.

Gateway displays a list of all Remote Sites in the right pane.

3. In the right pane, right-click the name of the Remote Site for which you want to modify the Site attributes. Select **Modify...** from the context menu.

Gateway displays the *Remote Site* editing window. This window has the same tabs and fields as the [New Remote Site](#) window above.

4. Enter new values in the editing fields as required.
5. When you have finished modifying the attributes, click **OK** to accept the new values and close the window.

Deleting a Remote Site

To delete an existing Remote Site:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
Gateway displays a list of all Remote Sites in the right pane.
3. In the right pane, right-click the name of the Remote Site you want to delete. Select **Delete...** from the context menu.
Gateway displays a confirmation dialog box.
4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
Alternatively, click **No** or **Cancel** to abandon the operation.
Gateway deletes the Remote Site from the database and removes the entry from the display window.

Axway Gateway: Managing Partners

Address configuration for Remote Sites

This topic provides additional information about address configuration for Gateway partners. In the GUI, you enter this information in the **Network address** tab of the *New Remote Site* window. The parameters required depend on your network interface:

- [TCP/IP](#)
- [SNA LU 6.2](#)

Configuring Remote Sites for TCP/IP

To create a Remote Site that uses TCP/IP, you supply the remote machine name (if using a name server) or its IP address (such as 193.56.234.50), and the port number on which the remote file transfer protocol is listening.

For example, create a Remote Site using the following command:

```
peladm create_site -pr PHSE -a REMOTE_PHSE_TCP -pi RPT0  
                  -ct TCP -dest_address 193.56.234.50/22200
```

This command creates a destination Site with the following properties:

- Protocol = PeSIT E (-pr PHSE)
- Alias name = REMOTE_PHSE_TCP (-a REMOTE_PHSE_TCP)
- Protocol identifier = RPT0 (-pi RPT0)

To access this Site, Gateway:

- Uses TCP/IP network (-ct TCP)
- Calls the IP address 193.56.234.50 on port number 22200

The partner application must be running on a platform with this IP address and listening to this port number.

Configuring Remote Sites for RAS

For Windows platforms only.

In Gateway, each remote partner participating in a file transfer must be defined as a Remote Site object.

To define a Remote Site as an RAS server, you must complete the TCP/IP address field in Remote Site definition in the following format:

- **Network type** = TCP
- **Network address** = [destination] nnn.nnn.nnn.nnn
- **Port** = mmmm

where [destination] is a section name in the RASPPP.INI file and nnn.nnn.nnn.nnn is the called TCP/IP host address. For example:

- **Network type** = TCP
- **Network address** = [PARIS] 172.20.102.33
- **Port** = 2345

Configuring Remote Sites for SNA LU 6.2

To create a Remote Site accessible over SNA LU 6.2, set the **Network type** to **LU 6.2**.

The field Machine name is used to define the *CPI-C side information*. Note that in some SNA access methods, *CPI-C side information* is referred to as *CPI-C symbolic destinations*.

The field **LU: Local** contains the Logical Unit for the local partner.

For example, create a Remote Site using the following command:

```
peladm create_site -pr PHSE -a REMOTE_LU -pi RPX0  
                  -ct LU62 -dest_address TOMVS@IPASOPL1@0
```

This command creates a destination Site with the following properties:

- Protocol = PHSE (-pr PHSE)
- Alias name = REMOTE_LU (-a REMOTE_LU)
- Protocol identifier = RPX0 (-pi RPX0)

To access this Site, Gateway:

- Uses the SNA network (-ct LU62)
- Calls the address TOMVS@IPASOPL1@0 (-dest_address TOMVS@IPASOPL1@0)

You must define the corresponding and related SNA configuration parameters in the SNA configuration database of your system before you create the Gateway Sites that use these parameters.

For Windows users

Every partner involved in a file transfer must be defined in Gateway. To identify a Remote Site as the LU 6.2 partner, the Site definition must include the following parameters:

- Network type: LU 6.2
- Machine: xxxxxxxx

where xxxxxxxx is a CPI-C symbolic destination defined in the SNA configuration database.

Specify the *CPI-C symbolic destination* name in the **Destination** field in the Gateway Remote Site definition. It identifies the SNA resources to use to connect to the Site.

In addition, if you are using PEL protocol over SNA LU 6.2 with the remote partner, the acknowledgment window must be greater than 0. If you are using PeSIT HS protocol over SNA LU 6.2, the acknowledgment window must be 1.

Also refer to [Configuring protocols: SNA LU 6.2](#).

For UNIX users

On most UNIX systems, you can define CPI-C information profiles that group all SNA parameter values required to establish connections to the Remote Site.

Axway Gateway: Managing Partners

New Remote Site: General tab

Field	Description
Alias	Enter the Remote Site alias. Maximum: 31 alphanumeric characters. Note: When creating PassPort (TPM) Template Sites, the alias is limited to 23 alphanumeric characters.

Field	Description
Protocol	Select one of the following protocols: • PeSIT D • PeSIT E • OFTP – then select the protocol version: ◦ R1.2 ◦ R1.3 ◦ R1.4 ◦ R2.0 • FTP • HTTP • FTP/HTTP • POP3 • SMTP • SFTP • PEL – then select the protocol version: ◦ No = no value ◦ HV = High Value ◦ TR = Trailer ◦ HV/TR = High Value and Trailer ◦ CRC = Cyclic Redundancy Code • AS1_POP3 • AS1_SMTP • AS2 • AS3 • RosettaNet_HTTP • RosettaNet_SMTP • RosettaNet_POP3 • SWIFTNet • JMS • X.400 When you select a protocol, the set of available tabs changes dynamically. Note: If you are using SWIFTNet, refer to Configuring the Axway Gateway/SWIFTNet environment.

Field	Description
Protocol identifier	<i>Not for all protocols</i> Enter the Remote Site protocol name. Maximum: 25 alphanumeric characters.
Group name	Enter a group name for this Site. Note that the Site must belong to the same group as the Application used for a given Transfer. Default = GDEFAULT.
Transfer owner	Enter the Transfer owner name to assign to Transfer Requests created by Gateway. For example, when Gateway receives an incoming Transfer that must be routed, it automatically creates an outgoing Transfer Request. The user that you specify has all the usual Transfer access rights. Maximum: 31 alphanumeric characters.
Partner local alias	Enter the Local Site alias to use during the connection phase. Maximum: 31 alphanumeric characters. If you do not enter a value in this field, Gateway uses the default Local Site specified in the configuration menu.
Comments	<i>Optional</i> Enter a free-text description for the object. Maximum: 80 alphanumeric characters.
Directory for received files	Enter the name of the directory where the received files are stored. Maximum: 127 alphanumeric characters. If you also specify a received file directory in the Application object, the Application definition takes precedence.
Role	Select one or more of the following options: • Initiator/Sender • Initiator/Receiver • Responder/Sender (mandatory for POP3 protocol) • Responder/Receiver (mandatory for SMTP protocol)
Initialization mode	Initialization attributes for the Site. On startup, the Site functions in the modes that you specify here. Select one or more of the following options: • Initiating Site • Responding Site (mandatory for POP3 and SMTP) • Trace

[Working with Remote Sites](#)[Site objects: Parameters List](#)[About Site objects](#)

Axway Gateway: Managing Partners

New Remote Site: Options tab

Field	Description
Maximum data block size (in bytes)	Specify the maximum data block size in bytes. Integer: 128 - 32768. Default = 4000.
Maximum number of simultaneous connections to this Site	Specify the maximum number of connections to this Site. Integer: 0 - 999. Default = 1. For POP3 Sites, this value is always set to 1.
Maximum number of simultaneous connections from this Site	Specify the maximum number of connections from this Site. Integer: 0 - 999. Default = 1. For POP3 Sites, this value is always set to 0.
Maximum number of connection retries before time limit	Specify the maximum number of times the software tries to connect before stopping the Site. Integer: 0 - 999. Default = 10.
Time before first retry (in seconds)	Specify the minimum delay before the first connection retry. Integer: 1 - 9999. Default = 10. Each subsequent retry doubles the time that you enter here. For example, if you enter 10 seconds, the first retry occurs after 10 seconds, the second after 20 seconds, the third after 40 seconds, and so on until either: • The maximum interval between retries is reached; or • The maximum number of connection retries is reached

Field	Description
Maximum interval between retries (in seconds)	Specify the maximum interval between connection retries. The value that you enter here must be greater than or equal to the value entered in the Time before first retry field. Integer: 1 - 9999. Default = 1800. For example, if you enter 30 in this field, 10 in the Time before first retry field and 10 in the Maximum number of connection retries field, then the first retry occurs after 10 seconds, the second after 20 seconds and the third after 30 seconds since this is the maximum interval allowed. Each subsequent retry occurs after 30 seconds until the maximum number of retries is reached, in this case 10.
Sentinel: Transfer Filter	Use this option to send a record of Transfer state changes to Sentinel. Select one of the following: • All = sends all Transfer state changes to Sentinel • Summary = sends a summary of Transfer state changes. Summary Transfer states are: canceled, ended, acked, created, to begin • None = sends no Transfer state changes • Undefined = Gateway uses the value set in the configuration menu

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[About Axway Sentinel](#)

Axway Gateway: Managing Partners

New Remote Site: Network address tab

Field	Description
Communication address	You can define up to four communication addresses for the Remote Site. You must define the properties for the Main address first. To view an existing definition, or define the properties for a new address, select one of the communication paths used to connect to the Site: • Main address (default) • Alternative address n°1 • Alternative address n°2 • Alternative address n°3
Network type	Select one of the following: • TCP/IP • LU 6.2 • Empty (default) The fields displayed in the right pane depend on the type of network that you select in this field.
Initial communication path	This field displays the initial address used to connect to the Site. By default this field is set to Main address and is grayed out. However, if an error occurs on the Main address, you can select another communication address from the drop-down list.
Right pane - TCP/IP specific fields: For more information, refer to Address configuration for Remote Sites .	
IP address or machine name	Enter the IP address. Maximum: 63 alphanumeric characters.
Port number	Enter the port number.
Network Proxy	This field displays a list of Proxies whose protocol is either SOCKS4 or HTTPS. This field is available for all protocols. Select a Proxy from the drop-down list.
FTP/HTTP Proxy	<i>FTP and HTTP only</i> This field displays the list of available FTP or HTTP Proxies, depending on the type of site. Select a corresponding Proxy from the drop-down list.

Field	Description
Allowed Router Agents	Select the Secure Relay Router Agents that are allowed to connect to this Site.
Bypass XSR	If enabled: Bypass XSR for outgoing calls. Outgoing connections are made directly.
Right pane - LU 6.2 specific fields: For more information, refer to Address configuration for Remote Sites .	
Destination (mandatory)	Enter the CPI-C symbolic destination defined in the SNA configuration database. Maximum: 8 alphanumeric characters. The symbolic destination identifies the SNA resources to use to connect to the Remote Site. This field contains a definition of all the other parameters on this tab. Any values that you enter in the other fields on this tab will override the values in the CPI-C symbolic destination.
User identification	Enter the login user ID.
Password	Enter the login user password.
Type	Enter the conversation type. This version of the product supports <i>Basic Conversation</i> only.
Mode	Enter the LU 6.2 mode name to use. The Mode Name determines the session parameters to use.
LU Partner	Enter the Logical Unit for the Remote partner.
LU Local	Enter the Logical Unit for the Local partner.
Transaction	Enter the name of the CPI-C Transaction program that you want to invoke.
PIP	Enter the Program Initialization Parameters.

[Working with Remote Sites](#)

[Address configuration for Remote Sites](#)

[Site objects: Parameters List](#)

New Remote Site: PeSIT tab

This topic describes the fields on the following tab of the *New Remote Site* window:

- **PeSIT** (the fields displayed depend on whether the selected protocol is PeSIT HS D or E)

Field	Description
Pre-connection phase:	
ID to send to partner (field)	<i>Optional</i> Set the identifier and password to send to the partner Site in the pre-connection phase.
+ Set password to send to partner (button)	Maximum: 8 alphanumeric characters in each field.
ID to receive from partner (field)	<i>Optional</i> Enter the identifier and password to receive from the partner Site in the pre-connection phase. Gateway checks that the received partner ID and password correspond to the values that you enter here.
+ Set password to receive from partner (button)	
Connection phase:	
Set Client password to send / to receive (buttons)	<i>Optional</i> Open dialog boxes to enter the client password to send and to check during the connection phase.
Set Server password to send / to receive (buttons)	<i>Optional</i> Enter the server password to send and to check during the connection phase. • PeSIT HS D: enter a numeric value less than 65536 • PeSIT HS E: enter up to 8 alphanumeric characters
Restart allowed	Select this option to enable the restart function. The restart function allows you to restart an interrupted Transfer from a specified synchronization point. It is always the sender who initiates the restart, and the data receiver who decides from which synchronization point to restart the Transfer.

Field	Description
Resynchronization allowed	Select this option to enable the resynchronization function. If an incident occurs during data transfer, this option allows you to ask the partner to restart the Transfer from a previous synchronization point. The file remains open during this operation, which either user can initiate.
Synchronization acknowledgment window	The synchronization acknowledgment window allows you to: - Specify the number of synchronization points between two acknowledgment messages - Restart an interrupted Transfer from a specific synchronization point Integer: 0 - 4. Default = 4. 0 = acknowledgments not used. For example, if you enter 3 in this field and 100 in the Interval between synchronization points field, Gateway sends 100 KB three times and then waits for an acknowledgment.
Interval between synchronization points	Specify the interval in Kbytes between synchronization points, that is the number of KB transmitted before the sender Site sends another synchronization point to the receiver Site. Integer: -1 - 999. -1 = synchronization points not used.
acknowledgment option <i>PeSIT HS E only</i>	Select one of the following acknowledgment options: Undefined: acknowledgment behavior not known (default) Not to ack: no reply expected By monitor: Gateway automatically sends an acknowledgment request By user: waits for a user acknowledgment request

Field	Description
Negative acknowledgment option <i>PeSIT HS E only</i>	The PeSIT HS E protocol supports END-TO-END acknowledgment, that is, the final destination of a file transfer can send a message to the originator to acknowledge reception of the file. This is a positive acknowledgment. The Negative acknowledgment option allows the final destination to indicate that an error occurred during the Transfer. Select this option to indicate that this Remote Site supports the negative acknowledgment extension. If you do not select this option, all negative acknowledgment requests are denied.
Message	<i>Optional</i> Enter a free-format user message (PI 99). Maximum: 512 alphanumeric characters. <i>For PeSIT HS only:</i> You can also use this field to transmit specific values to Axway Transfer CFT during the connection phase.
Change direction support	Select this option to poll this Remote Site on Gateway startup and to enable the Change Direction functionality, that is, to invert the Initiator/Responder roles between the partner Sites.
Polling interval	Specify the time (in minutes) to wait before starting the next polling cycle. Format: <i>MM</i>. Default = 0.
Maximum connections	Specify the maximum number of concurrent active connections in <i>Change direction</i> mode. Note that <i>change direction</i> connections are counted separately from the normal Initiator/Responder connections with this Site. Integer: 0 - 999. Default = 0.
Activated	Select this option to activate the polling functionality. If you do not select this option, all fields in the Change direction support section are ignored.
Permanent connection <i>PeSIT HS only</i>	Select this option to establish permanent polled connections between server and client when Change direction support (the polling option) is configured.

Field	Description
Connection hold delay	Specify the time (in seconds) that the connection remains active without any transmission events.
Heartbeat rate	Specify the interval (in seconds) between Heartbeat messages emitted by the server to confirm the presence of a file-requestor client. Integer: 0 - (2 ³¹ - 1).

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[About PeSIT Hors SIT](#)

Axway Gateway: Managing Partners

New Remote Site: OFTP tab

Field	Description
Data windows	Enter the maximum number of buffers that you can send or receive before the Site puts a synchronization point into the Mailbox. Integer: 1 - 999. Default = 100.
Network credit	Enter the number of Data Exchange Buffers the sender can send before the end of the file or the credit value is reached. The sender must then wait for an acknowledgment (a CDT response) before sending more DATA commands. Integer: 0 - 99. Default = 4. 0 = no synchronization points used.
Set password to send/ to check	<i>Optional</i> Open dialog boxes to enter the connection password to send (from the Initiator) and the connection password to check (received from the Responder).
User message	<i>Optional</i> Enter a user message to transmit during the connection phase. Maximum: 254 alphanumeric characters.

Field	Description
Max. size for received files (KB)	Enter the maximum size allowed for received files. Specify the value in KB. Integer: 0 - 999 999 999. 0 = no limit.
Max number of sessions	Maximum number of sessions (IN + OUT) for each remote site. Please note that Gateway supports per-partner granularity for selecting the template when using PassPort PM. • If a remote site with alias = <code>SSIDCODE.SIOICD+SSIDCODE.SIOORG</code> exists in Gateway, this site is selected as the template site. • If not, the default configured template site is used (<code>TODT</code>). If the value is 0, then the max number of simultaneous sessions it is not directly checked. It will be indirectly limited by the site's in/out connection max settings ("Maximum number of simultaneous connections to/from this Site" from the Options tab.)
acknowledgment option	Select the acknowledgment option: • Undefined: acknowledgment behavior not known (default) • Not to ack: no reply expected • By monitor: Gateway automatically sends an acknowledgment request • By user: the user must send an acknowledgment request
EERP Style	An OFTP session has two acknowledgment modes (French and International). The difference between these two modes is that the destination and originator fields in End-to-End-Response (EERP) and File Protocol Data Unit (FPDU) are inverted. Select the acknowledgment style: • French • International

Field	Description
Application method	Select the method to use to retrieve the Application object: • User exit • File name • User message field
Restart allowed	Check this box to enable the restart function. The restart function allows you to restart interrupted Transfers.
Compression allowed	Check this box to enable file compression. OFTP allows horizontal data compression (the deletion of repeated characters in a record).
Special Logic Coding	Check this box to enable special logic coding. Special logic coding is an OFTP protocol feature used to detect and correct errors, in particular with less reliable asynchronous connections.
Authentication method	<i>OFTP R2.0 only</i> Specify the authentication method to use: • None: No authentication required • EncryptedData: AUCH is CMS EncryptedData • EnvelopedData: AUCH is CMS EnvelopedData Since this is not a negotiable feature, contact your partner to agree on the method used.
Authentication certificate: Type	<i>OFTP R2.0 only</i> Specify the type of authentication certificate to use: • SEC_ALIAS • Passport PS entity • Passport PS certificate
Authentication certificate: Name	<i>OFTP R2.0 only</i> Specify the authentication certificate to use.

New Remote Site: FTP, HTTP, FTP/HTTP tabs

This topic describes the fields on the following tabs of the *New Remote Site* window:

- **FTP/HTTP**
- **FTP**
- **HTTP**

Field	Description
HTTP/FTP: Protocol security:	
ID to Send (field) + Set password to send to partner (button)	<i>Optional</i> Enter the identifier and password to send during the connection phase. Maximum: 63 alphanumeric characters in the ID field and 127 alphanumeric characters in the password field. In client mode, Gateway uses the values that you set in these fields for the login phase on a remote server.
ID to Check (field) + Set password to receive from partner (button)	Enter the identifier and password used to check partner identity. The identifier corresponds to the protocol identifier. In server mode this field is mandatory. Gateway uses these values to check user/password values provided by the client.
HTTP CGI Parameters / FTP Specific command	Enter specific FTP <code>SITE</code> or <code>QUOTE</code> commands or HTTP CGI parameters to send before or after the Transfer.

Field	Description
Application method	<i>Responder mode only</i> Select the method to use to retrieve the Application object: • User exit: select this option if the Application name is not in the Remote Site definition, or when you need additional information to identify the Application name • User message field: the Application is located in the AIM field and is transmitted with the FTP SITE command • File name: the Application is named in the file name parameter Note that if the Application is not defined in the method that you select in this field, the Transfer Request is rejected.
Acknowledgement option	Select one of the following acknowledgement options: • Undefined: acknowledgement behavior not known (default) • Not to ack: no reply expected • By monitor: Gateway automatically sends an acknowledgement request • By user: waits for a user acknowledgement request
FTP:	
Disable Gateway specific behavior	Select this option to disable the use of FTP commands specific to Gateway.
Allow different IP addresses for control and data sessions	The remote IP address of a proxy control session may be different from the remote address of its data session. Select this option to disable the protection against FTP bounce attacks and allow different IP addresses for control and data sessions.

Field	Description
Restart allowed	Select this option to enable the restart function. This allows Gateway to restart Transfers interrupted by incidents. In BLOCK or COMPRESSED mode, synchronization points are sent with the data. When you restart a Transfer, a synchronization point is negotiated, and both partners can start managing the file at the position they kept. Gateway records these synchronization points in the Mailbox. This means that you cannot restart Transfers stored in real directories in BLOCK or COMPRESSED mode.
Restart type	Select one of the following restart modes: • Stream: The client sends the restart offset to the server and both partners initialize their file access depending on the restart offset. The Transfer can then begin again. The client determines the restart point depending on the size of the data already received: ◦ If downloading the file, the client knows the size already received ◦ If uploading the file, the client must use the FTP <code>SIZE</code> command to determine the size already received by the server • Block and Compressed: during data exchange, some restart information is sent in the data flow. The restart point can be a string of alphanumeric characters without a SPACE (" "). • All: enable Stream, Block and Compressed restart modes • None Note that you cannot set the Restart type to Block and Compressed if the Transfer mode is set to None.

Field	Description
Transfer mode	You can transfer data in stream, block, or compressed mode. • Stream mode: In stream mode, a new data connection is established for each Transfer. In this mode, the Site can send and receive different files simultaneously. Stream mode is always available, regardless of the value you select in this field. • Block and compressed modes: In these modes, the data connection may be kept open, depending on the FTP response code sent by the Responder after the end of each Transfer. Select one of the following Transfer modes: • (Empty) • None (Stream mode only) (default) • Compressed • Block • Block and compressed
Data structure	Select one of the following data structures: • File: no internal structure is defined and the file is considered to be a continuous sequence of data bytes • Record: the file is made up of sequential records • File and Record
Passive mode	In passive mode, the client always initiates the data session. Select one of the following passive mode options: • Responder and Initiator • Responder: if the Site is the responder, it must accept PASV commands • Initiator: if the Site is the client, it can use PASV commands • None (default): the Site does not accept PASV commands

Field	Description
Quotas: Use the quotas section to defend against hammer attacks. A hammer attack can occur when a user floods the server with requests until it can no longer respond to any other users.	
Maximum uploads B/min	Set the maximum number of Kbytes that you can upload to this Remote Site per minute.
Maximum uploads Requests/min	Set the maximum number of Transfer Requests that you can upload per minute.
Maximum uploads MB	Set the maximum file size in MB.
Maximum downloads KB/min	Set the maximum number of Kbytes of data that you can download from this Remote Site per minute.
Maximum downloads Requests/min	Set the maximum number of Transfer Requests that you can download per minute.
Command sensitivity	Move the sensitivity cursor to the level of sensitivity required. This parameter allows you to define the number of commands that you can pass during a session. Each command corresponds to a certain credit, depending on how costly the command is in terms of memory and system resources. The higher the sensitivity, the fewer commands you can pass.

Field	Description
FTP CS List fields	Select this option in Responder mode to define which columns to send to FTPCS clients in the LIST command. Click the FTP CS List fields button to display the list of fields that you can send to the FTPCS client. By default, all columns are listed. Regardless of the field order that you specify, fields are always sent in the same predefined order (as listed below). Check one or more of the following: • LOCID: local_ident • GTYPE: VFD type (real or virtual directory, mount point, real or virtual file) • RIGHTS: VFD rights • FILENAME: long name - <i>s<xfer_ident>.<filename></i> for a virtual file, otherwise <i><filename></i> • SFILENAME: short name - <i><filename></i> only • APPLI: Application name • SIZE: file size • DATE: date of last modification/ creation of transfer • TYPE: MVS transfer state • RSTATE: routing state (T: to route, R: routed, A: acked) • UMSG: user message • ORG: origin • DEST: destination • P1: parameter 1 • P2: parameter 2 • PERM: permanent transfer • DATA_CODE: B for binary, T for test If the item described in the list is not a virtual file, the only significant fields are GTYPE, RIGHTS, FILENAME, SFILENAME, DATE and SIZE. The Gateway FTP server completes these fields. LOCID and GTYPE are required fields and are always sent to the client.

Field	Description
HTTP client:	
Identification method	Select the identification method to use when transferring files: • Anonymous: sends neither user nor password • CGI: adds user and password to the CGI parameters • Web: standard HTTP identification method (adds user and password to HTTP headers) • Cookies: adds user and password to CGI parameters of the login request. Each subsequent Transfer Request does not send these parameters, but rather the cookies received with the login response. If you select this option, complete the Login request fields. • NTLM: use NTLM authentication (based on configured user and password)
Send transfer attributes	Check this box to send the following Transfer attributes with the file: • <code>appli</code> • <code>dest</code> • <code>org</code> • <code>xfer_ident</code> • <code>msg</code> These attributes are sent as CGI parameters.
User agent	Enter the ID of the client application used to contact the remote partner. This information is useful for certain logging and statistics processes. Maximum: 128 alphanumeric characters.
Login request (only available if you set Identification method to Cookies):	
Login path	Specify the URL path to the server CGI script that will identify Gateway and return the cookie. Specify the path to use with the login request. For example: <code>/cgi-bin/login.cgi</code>. This is not an absolute URL. Maximum: 256 alphanumeric characters.

Field	Description
HTTP method	Select the HTTP method to use with the login request: • GET • POST
Login CGI parameters	<i>Optional</i> Enter additional CGI parameters to add to the login requests. Maximum 80 alphanumeric characters in the format: <code>login_param1=value1&login_param2=value2</code>

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[About FTP](#)

[About HTTP](#)

Axway Gateway: Managing Partners

New Remote Site: PEL tab

Field	Description
Protocol security	
Set site password in initiator mode	Opens a dialog box to enter the password for the Remote Site in Initiator mode. Maximum: 8 alphanumeric characters.
Limits	
Number of blocks between acknowledgments	Enter the number of data blocks to send before waiting for an acknowledgment from the recipient. Default = 45.

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[About PEL](#)

Axway Gateway: Managing Partners

New Remote Site: POP3 tab

Gateway displays this tab if you select either of the following protocols in the [General tab](#):

- POP3
- AS1_POP3

Field	Description
Authentication method	Enables you to define the authentication mechanism (AUTH) to use when connecting to the POP3 server. Select one of the following values: • Auto: Gateway automatically selects an authentication method that the POP3 server supports (in the following order: APOP, CRAM, BASIC) • Basic: the use of USER and PASS in clear text (default) • CRAM (Challenge Response Authentication Method): user and password are transmitted in encrypted form, as specified in RFC 2195 • Apop (<user> <digest-MD5>): user id is transmitted in plain text, but the password is transmitted in MD5 digest, as specified in RFC 2449 Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Login identification (mandatory)	Enter the user login name to connect to the POP3 server account. Maximum: 24 characters.
Set Login password (mandatory)	Opens a dialog box to enter the user login password to connect to the POP3 server account. Maximum: 14 characters.

Field	Description
Application method	Select the method to use to retrieve the Application object: • User exit: select this option if the Application name is not in the Remote Site definition, or when you need additional information to identify the Application name • User message field: not supported • File name: not supported Note: If you do not define an Application method, the Transfer Request is rejected.
Extract file	Check this option to indicate that there is a file attachment to extract on reception. If you also set this value in the Application object, the Application definition overrides the value that you set here.
Polling options	
Polling interval (in minutes)	Specify the time to wait before starting the next polling cycle. Specify this value in minutes in the format <i>MM</i>. Default = 10.
Activated	Check this option to activate polling.
acknowledgment option	Select one of the following options: • Undefined • Not to ack • By monitor • By user
Mail disposition on error	Select one of the following options: • Undefined • Leave • Delete • Recover (default)
SMIME	<i>Only for POP3 (not for AS1_POP3)</i> Check this option to activate S/MIME security functions.

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[About SMTP/POP3](#)

Axway Gateway: Managing Partners

New Remote Site: SMTP tab

Field	Description
Alphanumeric X-Priority (SEPA Mail)	Check this option to indicate that the X-Priority header should take alphanumeric values (<i>HIGHEST, HIGH, NORMAL, LOW, LOWEST</i>). If left unchecked, the header will take numeric values (1 to 5).
Authentication method	Select the authentication method to use when connecting to the SMTP server: • None (default) • Plain • Login (provide user and password) • Anonymous
Login identification	Enter the user login name. Maximum: 24 alphanumeric characters.
Set Login password	Opens a dialog box to enter the user login password. Maximum: 14 alphanumeric characters.
Default email address	Enter the default destination email address (To) for the Remote Site. Maximum: 128 characters. The address that you enter in this field is only used if you do not define a destination email address in the Transfer Request.
Attach file	Check this option to indicate whether or not to attach a file by default. If you check this option, Gateway creates an email and attaches the file. Do not check this option if the file to transfer is already formatted as an email.

[Working with Remote Sites](#)[Site objects: Parameters List](#)[About SMTP/POP3](#)

Axway Gateway: Managing Partners

New Remote Site: SFTP tab

Field	Description
Parameters to connect to Remote Site	
User name (mandatory)	Enter the client user name to send to the Remote Site during the connection phase. This field serves to authenticate the client. Maximum: 25 alphanumeric characters.
SSH Profile	Enter the name of the SSH profile to use for the outgoing connection. This parameter is only used in client mode. The SSH Profile that you specify must be of the type CLIENT and must be enabled.
Set password to send to partner	Opens a dialog box to enter the password to send to the Remote Site during the connection phase. Maximum: 15 alphanumeric characters. The password is only required if the server requests the PASSWORD authentication method.
Private key alias	Select the private key alias to use to authenticate your Site to the remote partner. This field is mandatory if the server requests the PUBLIC KEY authentication method. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Certificate alias	Select the certificate alias to use to authenticate your Site to the partner. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Disable SFTPV6	- If checked, SFTPV6 will not be negotiated (server role - not advertised, client role - not selected) nor accepted (server role, when client is starting directly with SFTPV6, Gateway server will respond with SFTPV3). Only SFTPV3 will be used for this remote site. - If unchecked, SFTPV6 is preferred whenever available
Remote Site parameters to check	

Field	Description
User name (mandatory)	Enter the user identifier that corresponds to the protocol identifier. Maximum: 25 alphanumeric characters. In Responder mode, Gateway uses this value to check the user name that it receives from the client.
Set password	Opens a dialog box to enter the client password. Maximum: 14 characters. In Responder mode, Gateway uses this value to check the user password that it receives from the client.
Remote Site authentication (Initiator and Responder mode) The remote client uses the values that you enter in the <i>Remote Site authentication</i> section to authenticate the server. Complete the Public key fields if the public key algorithm negotiated is a key pair algorithm. The parameters that you set in the Remote Site take precedence over the same parameters set in the SSH Profile.	
Public key alias	Select the public key alias to use to authenticate the partner. When Gateway receives a key from a remote server, it compares the key with the public key alias that you specify here.
Public key group	Enter the public key group name. Maximum: 31 alphanumeric characters. When you import a key automatically, it is associated with the group that you specify here. If you do not specify a group in this field, the key is associated with the group GDEFAULT.
Remote certificate filter The following parameters are used to check the authenticity of a partner. Complete these fields if the public key algorithm negotiated is a certificate algorithm.	
Subject certificate alias	Select the certificate alias used to authenticate the partner. If you select to authenticate the partner by a certificate, you do not need to complete the Public key alias and Public key group fields.
Issuer certificate alias	Select the issuer (CA) certificate alias to use to authenticate the partner certificate.
Subject name pattern	Enter the subject distinguished name pattern. You can use wildcard characters (* and ?) to filter accepted certificates. Maximum: 512 alphanumeric characters.

Field	Description
Issuer name pattern	Enter the issuer distinguished name pattern. Maximum: 512 alphanumeric characters.
Remote PassPort PS partner entity filter	
Name	When Gateway operates in server mode with a partner defined in PassPort, enter the identity of the PassPort partner.
Newline convention	Set the end-of-line code, depending on the remote system type. Conversion only occurs in text mode and not in binary mode. Select one of the following values: • UNIX Gateway searches for the LF sequence to detect end-of-lines (EOL). • Windows Gateway searches for the CRLF sequence to detect EOLs • (Empty) When the line ending convention is set to Empty no line ending conversions occur. Once the EOL is identified, Gateway converts the EOL from the incoming data to the native system EOL of the system Gateway is installed on (Windows:CRLF, Unix:LF). UNIX: to prevent occurrences of CRLF becoming CRCRLF, Gateway verifies the character in front of each LF character it finds. If it finds a CR, it automatically replaces the whole CRLF sequence with the native EOL. Note: You can also set this parameter in the Transfer Request. The value set in the Transfer Request takes priority over the value set in the Remote Site. Known limitations: • when using an application in Text Mode with Record length = 1, the line ending is no longer included in the text file • when using an application in Text Mode with Record length = 2, the CRLF -> CRCRLF is no longer compensated

Field	Description
Application method	Select the method to use to retrieve the Application object: • User exit • File name • User message field
Pipeline level	Set the maximum number of outstanding buffers (non-acknowledged) that the client may send to the server before waiting for an acknowledgment. Possible values: 0 - 255. Default = 0 (the values 0 and 1 are equivalent.) This parameter is used in client mode only (outgoing transfers).
Transport security in SecureRelay for outgoing connection	Select this option to use SSH termination in Secure Relay.
Disable Gateway specific behavior	Select this option to disable the use of SFTP commands specific to Gateway (<code>SSH_FXP_OPEN@axway.com</code> and <code>SSH_FXP_OPENDIR@axway.com</code>). These specific commands are useful in the context of an exchange between Gateway and Gateway SFTP.

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[About SFTP](#)

[Axway PassPort PS connector](#)

Security guide > [SSH authentication](#)

Security guide > [About Secure Relay](#)

Axway Gateway: Managing Partners

New Remote Site: SWIFTNet tab

Field	Description
Login (mandatory)	Client user name to send to the Remote Site during the connection phase. Maximum: 63 alphanumeric characters.
Set password (mandatory)	Opens a dialog box to set the password to send to the Remote Site during the connection phase. Maximum: 20 alphanumeric characters.
RA instance (mandatory)	Name of the SWIFTNet RA (Remote API host adapter) local access point.
Message partner (mandatory)	Application name declared in SAG.
Event end point (mandatory)	Arbitrary name allowing SAG to route received events.
General sub-tab	
Alternate RA installation pathname	<i>Only if SWIFTNet RA has been installed in more than one place</i> Alternative path for SWIFTNet RA.
Alternate configuration file	Alternative configuration file for the instance of SWIFTNet RA.
Maximum number of client processes	Maximum number of simultaneous client processes allowed. Default = 1.
Maximum number of server processes	Maximum number of simultaneous server processes allowed. Default = 1.

Field	Description
Sign ident list	List of signatory SWIFTNet users. This is a list of user login ids with their corresponding passwords. For each user, enter the login id in the first field and the password in the second field. Then click Add to add the user to the list. You can also modify a user and/or password. Select the user in the list, make the required changes and then click Update. To delete a user from the list, click Delete.
Decrypt ident list	List of decrypting SWIFTNet users. This is a list of user login ids with their corresponding passwords. For each user, enter the login id in the first field and the password in the second field. Then click Add to add the user to the list. You can also modify a user and/or password. Select the user in the list, make the required changes and then click Update. To delete a user from the list, click Delete.
Backup sites	The list of backup sites selected. You can add or remove Sites from the list by clicking the Change... button. Up to 4 backup Sites can be selected.
Maximum number of initiator transfers	The value of the maximum number of transfers in Initiator mode, FileAct only. By default, this parameter is set to 0, meaning unlimited. The maximum value allowed is 999. Transfers switched from a nominal site to a backup site will be subject to the nominal's site limit. The site parameter is not available to the connection related user exits. Note: Both the per-site and the global maximum number of initiator transfers are applied simultaneously (if set to a non-0 value); whichever limit is reached first will come into effect.
SnF (Store-and-Forward) sub-tab	
Queues	List of all SnF queues to be managed.
SNL end point	Name allowing the reception of SnF events.
Input Channel	The name of the input channel. Note: The input channel must already exist! First create the channel using the pelctl create_input_channel command.
[List of output channels with their details]	

Field	Description
New, Modify and Delete buttons	Use these buttons to create, modify or delete output channels. When you click New or Modify , the <i>Output Channel Parameters</i> window opens.

Output channel parameters

Field	Description
Channel name	The name of the output channel.
Queue name	The output channel's queue name.
Window size	<i>Optional</i> The output channel window size. In SWIFT, the window size is an optional element that is used to limit the number of concurrent messages or files sent by SWIFT on a push session. If the element is not present, then a default value is used for the window size (specified in SWIFT documentation). SWIFT adjusts the window size to the maximum value in case the application requests a higher value than the maximum allowed value. Normally you do not need to change the default value. Only specify the window size if you want to restrict the value.
Receive retrieved messages	<i>Optional</i> Enable the attribute in order to receive retrieved messages. By setting this parameter, the output channel will be opened having the element "protocolLevel" set to value "2", as indicated by SWIFT documentation. Example: <code>-receive_retrieved_messages "Y"</code> Note: To receive retrieved messages, the queue for which the session is being started: • Must have been defined for service <code>"swift.snf.system (!p, !x)"</code> • Must have been used on the retrieval request payload
<i>Optional</i> Click Select subset to open the <i>Output Channel Subset</i> window and define the Subset elements for the output channel.	

Field	Description
<i>Output Channel Subset window</i>	Select the required subset elements: • InterAct • FileAct • System • Urgent • Normal • InterAct_Urgent • InterAct_Normal • FileAct_Urgent • FileAct_Normal • System_Urgent • System_Normal The order in which the subsets are listed is important! Use the up and down arrows to specify the order of the elements. Note: If no subset is selected, the output channel will receive all types of message of all priorities, in a first in, first out order.

Axway Gateway: Managing Partners

New Remote Site: JMS tab

Field	Description
User name	The user name (for the MOM user) for connection to the JMS provider. Maximum: 63 alphanumeric characters.
Set Password	Opens a dialog box to set the password for the MOM user, for connection to the JMS provider. Maximum: 127 alphanumeric characters.
Class loader	The Java class path used to locate <code>jar</code> files of the JMS provider. Enter the list of fully qualified <code>jar</code> files (including path) separated by the ':' character on UNIX systems and ';' on Windows systems.

Field	Description
Initial factory	The Java class name of the Initial Context Factory. For example: <code>com.sun.jndi.fscontext.RefFSContextFactory</code>
Connection factory	The JNDI lookup name of the connection factory to be used (configured by the JMS provider administrator).
URL	The connection URL of the JMS provider. It usually consists of: • a protocol • a hostname • a listening port For example: <code>tcp://localhost:4569</code> or <code>file:/var/mqm</code>
Type	The type of object used by the JMS provider to store/retrieve messages for this Site. Select the value: • QUEUE • TOPIC
JNDI name	The JNDI lookup name of the queue/topic to be used (configured by the JMS provider administrator).
Topic subscription	If topics are used, the subscription name used to subscribe to the topic.
Topic client Id	If topics are used, the client identifier used to subscribe to the topic.
Msg In selector	The JMS selector used to filter messages coming from JMS and transferred to Gateway.
Msg In directory	The output directory used to store the data of JMS messages transferred to Gateway.
Initialization properties This section displays the list of initialization properties for JMS connection. Modify the list according to the requirements of your JMS provider.	
Property	Property name

Field	Description
Value	Property value
New	Click New to add extra initialization properties.
Delete	Click Delete to remove an initialization property.

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[JMS connector](#)

[Configuring the Axway Gateway/JMS environment](#)

Axway Gateway: Managing Partners

New Remote Site: Net security tab

Use the **Net security** tab to define security options for this Site.

This tab is not available for SFTP.

Field	Description
Network security option	Select one of the following: • None • TLS (if you select this option, the following TLS security fields are displayed)
TLS security	
Security profile for outgoing connection	Enter the name of the outgoing TLS Security Profile. Maximum: 30 alphanumeric characters.
Security profile for incoming connection	Enter the name of the incoming TLS Security Profile. Maximum: 30 alphanumeric characters.
Remote certificate filter (Only FTP/HTTP)	

Field	Description
Subject certificate alias	Select the certificate alias used to authenticate the partner. If you select to authenticate the partner by a certificate, you do not need to complete the Public key alias and Public key group fields. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Issuer certificate alias	Select the issuer (CA) certificate alias to use to authenticate the partner certificate. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Subject name pattern	Enter a name pattern to use as a filter for accepted subject certificates. For example: This value can contain wildcard characters (* and ?).
Issuer name pattern	Enter a name pattern to use as a filter for the Issuer distinguished name. Maximum: 512 alphanumeric characters. This value can contain wildcard characters (* and ?).
Remote PassPort PS partner entity filter	
Name	When Gateway operates in server mode with a partner defined in PassPort, enter the identity of the PassPort partner.
Transport security in SecureRelay for outgoing connection	Select this option if you require client TLS termination in the DMZ.

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[Axway PassPort PS connector](#)

New Remote Site: X.400 tab

The Remote Site contains information on how to communicate with a remote Message Transfer Agent (MTA) or a Message Store (MS) mailbox.

P7 is the name of the protocol used to access a Message Store (MS) client from a User Agent (UA).

Field	Description
Version	Select: • MTA • MSP7
Password	
Set Password	Opens a dialog box to set the password. <i>For MTA:</i> Password that the local MTA uses to access the remote MTA. <i>For MSP7:</i> Password that the MS uses to authenticate itself. The MS in the connection-accept response sends the password and the P7-client validates the password before initiating any operations. Default: No authentication.
Set password to check	<i>For MTA:</i> Password that the remote MTA uses to access the local MTA. <i>For MSP7:</i> Password used by the P7-client to authenticate itself. The client at connection establishment sends the password and the MS validates the password before it accepts the connection.
Communication parameters	
These parameters identify the Service Access Points for the different communication layers. You must specify at least one of these parameters. Typically the TSAP is used. To specify a hexadecimal value, enter the value between apostrophes ('), for example, ' 7C, A2 '.	
PSAP	Presentation Service Access Point. This parameter defines the OSI address for the net, transport, session, and presentation layer.
SSAP	Session Service Access Point. This parameter defines the SSAP. Maximum 16 characters.

Field	Description
TSAP *	Transport Service Access Point. This parameter defines the TSAP. Maximum 20 characters.
Routing...	<i>MTA only</i> Click Routing.... Refer to Routing parameters .
Scheduling...	<i>MSP7 only</i> Click Scheduling.... Refer to Scheduling parameters .
Advanced...	Click Advanced.... For MTA-type Remote Sites, refer to Advanced parameters (MTA) . For MSP7-type Remote Sites, refer to Advanced parameters (MSP7) .

(*) Mandatory parameter

Additional parameters for MTA-type Remote Sites

Routing parameters

1. Complete the fields of the *Routing* window as follows:

Field	Description
Routing O/R address *	When there is no explicit information on where to route a message, the destination address is matched against any defined remote MTA routing address and the one which gives the best match is used.
Edit...	Click Edit... to configure the O/R address .
Clear	Click Clear to delete the displayed O/R address.

(*) Mandatory parameter

2. Click **OK** to confirm the settings.

Advanced parameters (MTA)

1. Complete the fields of the *Advanced* window as follows:

Field	Description
Version 84	Typically the MTA version is greater than 84. If the version is 84, select this option.
RTS X.410 mode	Typically, a version 84 system uses RTS X.410 mode whereas non-84 systems do not use RTS X.410 mode. Specify this option as appropriate.
External name	To have more than one remote MTA with the same MTA name, it is important to specify an external name that is to be used when contacting the MTA (MTA-MTA protocol). The MTA name specified in the X.400 MTA dialog is used locally and must be unique, whereas several MTAs can have the same external name. If no external name is specified, the MTA name specified in the X.400 MTA dialog is used.
ADMD	If you complete this parameter, the specified ADMD name is put in the originator's address if ADMD is set to an asterisk (*) when sending messages to this MTA. The reason for allowing this is that some ADMDs require the ADMD field to be set to their ADMD name when sending messages to that particular ADMD. The specified ADMD name is only substituted if the ADMD attribute in the O/R address of the originator is set to (*). Note: ADMD substitutions should only be used when you communicate with multiple ADMDs, and only if the ADMD requires that its ADMD name be used when contacting it.

- Click **OK** to confirm the settings.

Additional parameters for MSP7-type Remote Sites

Scheduling parameters

These parameters are used for the polling of external MS mailboxes.

- Complete the fields of the *Scheduling* window as follows:

Field	Description
Monthly	
New	Click this button to add Monthly scheduling parameters. Refer to <i>Monthly scheduling parameters</i> below.

Field	Description
Modify...	Click this button to change Monthly scheduling parameters. Refer to <i>Monthly scheduling parameters</i> below.
Delete	Click this button to delete the latest entry.
Weekly	
New	Click this button to add Weekly scheduling parameters. Refer to <i>Weekly scheduling parameters</i> below.
Modify...	Click this button to change Weekly scheduling parameters. Refer to <i>Weekly scheduling parameters</i> below.
Delete	Click this button to delete the latest entry.

Monthly scheduling parameters	
Field	Description
Month	Month of the polling sequence.
Date	
From	Start date of the monthly polling period.
To	Stop date of the monthly polling period.
Start time	Time when the polling sequence begins.
Stop time	Time when the polling sequence ends.
Frequency	Length of time between mailbox polls. If set to zero, there is exactly one mailbox poll at start time.
When you have finished, click OK to confirm the Monthly scheduling settings.	

Field	Description
Weekly scheduling parameters	
Day	
From	Start date of the weekly polling period.
To	Stop date of the weekly polling period.
Start time	Time when the polling sequence begins.
Stop time	Time when the polling sequence ends.
Frequency	Length of time between mailbox polls. If set to zero, there is exactly one mailbox poll at start time.
When you have finished, click OK to confirm the Weekly scheduling settings.	

- When you have finished completing all scheduling parameters, click **OK** to confirm the settings.

Advanced parameters (MSP7)

- Complete the fields of the *Advanced* window as follows:

Field	Description
Time between submission	Specifies the number of seconds between message submission. When the submission time event occurs, a check is made to see if the UA has any messages to be sent. If there are pending messages, a connection is established and all messages are sent. If this parameter is set to 0, the message submission will be done immediately after the UA has made a submission request. Default value: 0.
Max inactivity time	Specifies the number of seconds that the P7-client will keep an established MS connection up without any activity. If the parameter is set to 0, a connection is released as soon as all MS operations are made. Default value: 0.
Max connection attempts	Used by the P7-client when it fails to connect to the MS. The parameter states how many times a new connection attempt should be made before all messages waiting for submission will be returned to the UA as non-deliverable. Default value: 8.

Field	Description
Time between connection attempts	Used by the P7-client when it fails to connect to the MS. The parameter specifies the time, in seconds, that the P7-client should wait before it makes a new connection attempt. Default value: 60 seconds.
Local TSAP	Specifies the local TSAP. Maximum 20 characters. Default value: P7. To specify a hexadecimal value, enter the value between apostrophes ('), for example, '7C,A2'.
Poll at submission	Specifies whether or not the P7-client is allowed to retrieve messages after message submission. If selected, the P7-client is allowed to poll the Mailbox before the connection is released even if it is not yet time for the next mailbox poll. Default value: selected.
Submit at poll	Specifies whether or not the P7-client is allowed to submit messages after a mailbox poll. If selected, the P7-client is allowed to submit messages before the connection is released even if it is not yet time for the next message submission. Default value: selected.
Auto alert	Determines whether or not the P7-client should request the auto alert functionality in the MS. The auto alert functionality enables the MS to immediately inform the P7-client that a new message or report has been entered into the users mailbox. When the P7-client receives an alert indication, it will immediately fetch the new message or report. Note: The alert indication is only sent on an existing connection initiated by the P7-client. The auto alert function is used only if the configuration parameter Max inactivity time is greater than 0. If selected, the auto alert function is requested. If not selected, no auto alert function is requested. Default value: Not selected.

2. Click **OK** to confirm the settings.

[Working with Remote Sites](#)

[Site objects: Parameters List](#)

[X.400 connector](#)

Configuring X.400 O/R address

The *O/R Address* window is displayed if you click **Edit...** in:

- The configuration menu
- Remote Site
- Transfer Request

Message originators and recipients within X.400 are identified by their O/R address. The O/R address consists of a number of mandatory and optional address attributes. These attributes are organized hierarchically.

Specify the O/R address by modifying the address attributes in the *X.400 O/R Address* window. Only use [valid characters](#) in the O/R address, as listed in this topic.

Note: Not all fields are used for all objects. In this case they are grayed out.

General tab

Field	Description
Country (C)	Identifies the Country.
ADMD (A)	Administration Domain Name. Identifies an ADMD relative to the country specified by the Country Name. An ADMD provides a public messaging service within a single country. Several ADMDs may exist within a Country.
PRMD (P)	Private Domain Name. Identifies a PRMD relative to the ADMD or the Country. A PRMD is a messaging service within an organization. The ADMD operator must uniquely assign the Name of the Private Domain if the PRMD is connected to an ADMD service.
Organization (O)	Identifies an organization relative to the PRMD.
Org. Unit 1 (OU1)	Organization Unit Name 1-4. Can be used up to four times to identify divisions/departments within the organization.
Org. Unit 2 (OU2)	The names appear in order of importance. Always start with Org. Unit 1. You cannot enter an Org. Unit 2 unless you enter an Org. Unit 1 first, and so on.
Org. Unit 3 (OU3)	
Org. Unit 4 (OU4)	

Field	Description
Common name (CN)	Identifies a user, an application, or a distribution list. Caution: Common name is not supported in MTA version 84. You may encounter problems if your MTA is routed via a version 84 MTA.

Personal tab

Field	Description
Surname (S)	The Surname is a mandatory part of the Personal Name.
Given name (G)	The Given Name is an optional part of the Personal Name.
Initials (I)	The Initials are an optional part of the Personal Name.
Generation Qual (Q)	The Generation qualifier is an optional part of the Personal Name.

Domain defined tab

Field	Description
Type (DDA)	The Domain-defined attributes are specific to a messaging domain and are used to pass non-standard information.
Value	These attributes are used in addition to those that specify the user/application.

Additional tab

Field	Description
Free form name (FFN)	Free text.
Telephone number (TPH)	Telephone number of Originator/Recipient.

Valid characters in O/R addresses

Character description	Character
Uppercase alphabetic characters	A, B, C, ... Z
Lowercase alphabetic characters	a, b, c, ... z
Digits	0, 1, 2, ... 9
Space	
Apostrophe	'
Left and right parentheses	()
Plus sign	+
Comma and period	, .
Slash	/
Dash	-
Colon	:
Equal sign	=
Question mark	?

[About X.400](#)[X.400 connector](#)[Configuration: Connectors](#)[New Remote Site: X.400 tab](#)[New Transfer Request: X.420 tab](#)

Axway Gateway: Managing Partners

Command line operations for Local Sites

[`peladm create loc site`](#)

[peladm delete loc site](#)

[peladm update loc site](#)

[peldsp display loc site](#)

[peldsp select loc site](#)

[peldsp status loc site](#)

Creating Local Sites: *peladm create_loc_site*

Syntax	<code>peladm create_loc_site {-alias} [-proto_ident] [-comments] [-dir_path] [-mail_address] [- param1] [-param2]</code>
Description	Use this command to create a Local Site.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Local Site you are creating. <i>Not for FTP or SFTP Sites.</i> <code>-proto_ident (-pi)</code> : Enter the Site protocol name for the Site. <code>-comments (-cts)</code> : Enter a comment for user information. <code>-dir_path (-dp)</code> : Enter the path to the directory of the local file management system where received files are stored. <i>SMTP only</i> <code>-mail_address (-email)</code> : Enter the default email address to use if no address is specified in the Transfer Request. By default, Gateway uses the default <i>From</i> address. <code>-param1 (-p1)</code> : Enter a user parameter of up to 40 alphanumeric characters. <code>-param2 (-p2)</code> : Enter a user parameter of up to 40 alphanumeric characters.
Example	The following command creates a Local Site with the alias SITE_LOC. <pre>peladm create_loc_site -a SITE_LOC</pre>

Modifying Local Sites: *peladm update_loc_site*

Syntax	<code>peladm update_loc_site {-alias} [-proto_ident] [-comments] [-dir_path] [-mail_address] [- param1] [-param2]</code>
Description	Use this command to modify and update the attributes of a Local Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code>: Enter a Site alias for the Local Site you are modifying. <i>Not for FTP or SFTP Sites.</i> <code>-proto_ident (-pi)</code>: Enter the new Site protocol name for the Site. <code>-comments (-cts)</code>: Enter a modified comment for user information. <code>-dir_path (-dp)</code>: Enter a modified path to the directory of the local file management system where received files are stored. <i>SMTP only</i> <code>-mail_address (-email)</code>: Enter a modified default email address to use if no address is specified in the Transfer Request. By default Gateway uses the default From address. <code>-param1 (-p1)</code>: Enter a user parameter of up to 40 alphanumeric characters. <code>-param2 (-p2)</code>: Enter a user parameter of up to 40 alphanumeric characters.
Example	The following command resets the protocol identifier for the Local Site SITE. <pre>peladm update_loc_site -a SITE -pi SITE_PROTO_IDENT</pre>

Deleting Local Sites: *peladm delete_loc_site*

Syntax	<code>peladm delete_loc_site {-alias} [-proto_ident] [-comments] [-dir_path] [-mail_address] [- param1] [-param2]</code>

Description	Use this command to delete a Local Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter the Site alias for the Local Site you are deleting.
Example	The following command deletes the Local Site whose alias is SITE: <code>peladm delete_loc_site -a SITE</code>

Selecting Local Sites: *peldsp select_loc_site*

Syntax	<code>peldsp select_loc_site [-alias] [-proto_ident]</code>
Description	Use this command to filter the display of Local Sites using the criteria entered as parameters. To list all Local Sites, enter no parameters. You can then refine your selection using the optional parameters.
Parameters	<code>-alias (-a)</code> : Enter a Site alias for the Local Site as selection criteria. <i>Not for FTP or SFTP Sites.</i> <code>-proto_ident (-pi)</code> : Enter a Site protocol name as selection criteria.
Example	The following command generates a filtered display of Local Sites. Gateway displays the aliases of the Local Sites that match the selection criteria. <code>peldsp select_loc_site -a U</code> Each string parameter is seen as a truncated search key. The command returns a list of the Local Sites with aliases beginning with the letter U. UXTPDx02 UXTPDt03 UXTPLx02

Displaying Local Sites: *peldsp display_loc_site*

Syntax	<code>peldsp display_loc_site {-alias}</code>

Description	Use this command to display the attributes of a specific Local Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Local Site as selection criteria.
Example	When you use this command, Gateway displays each parameter on a separate line in the format: <code>name_of_field="value"</code> Each field name is prefixed by the string " <code>s_</code> ", referring to the Site object.

Displaying Local Site status: `peldsp status_loc_site`

Syntax	<code>peldsp status_loc_site {-alias}</code>
Description	Use this command to display the initial and dynamic status of a Local Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Local Site as selection criteria.
Example	When you use this command, Gateway displays each parameter on a separate line in the format: <code>name_of_field="value"</code> Each field name is prefixed by the string " <code>s_</code> ", referring to the Site object.

Axway Gateway: Managing Partners

Working with Remote Sites (command line)

[`peladm create site`](#)

[`peladm update site`](#)

[`peladm delete site`](#)

[`peldsp select site`](#)

[`peldsp display site`](#)

[`peldsp status site`](#)

[`pelctl start site`](#)

[pelctl stop site](#)

Creating Remote Sites: *peladm create_site*

Syntax	<code>peladm create_site {-alias} [optional parameters]</code>
Description	Use this command to create a Remote Site. The command has one mandatory parameter and many optional parameters.
Parameters	Mandatory <code>-alias (-a)</code>: Enter a Site alias for the Remote Site you are creating. Optional parameters Refer to Site objects: Parameters List.
Example 1	The following command creates a Remote Site with the properties: - alias: SITE - protocol: PeSIT HS E - network type: TCP/IP - called address: 1922575869 - max data block size: 2500 <pre>peladm create_site -a SITE -pr PHSE -ct TCPIP -dest_address 1922575869 -ds 2500</pre>

Example 2	The following command creates a Remote Site with the properties: • alias: SWIFT_AP_1 • protocol: SWIFTNet <pre>peladm create_site -alias SWIFT_AP_1 -protocol SWIFTNET -login_ident user1 -login_password password1 -init_conn_max 4 -resp_conn_max 1 -ra_instance Ra1 -message_partner appli1 -event_end_point appli1_ep</pre>

Modifying Remote Sites: *peladm update_site*

Syntax	<code>peladm update_site {-alias} [optional parameters]</code>
Description	Use this command to modify the properties of a Remote Site that you identify by its alias. You can modify all values except the alias. The syntax is the same as for the <code>peladm create_site</code> command.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Remote Site you are modifying. <i>Optional parameters</i> Refer to Site objects: Parameters List. Enter values for the parameters you want to modify.
Example	The following command resets the protocol for the Remote Site SITE to PeSIT HS E. <pre>peladm update_site -a SITE -pr PHSE</pre>

Deleting Remote Sites: *peladm delete_site*

Syntax	<code>peladm delete_site {-alias}</code>

Description	Use this command to delete a Remote Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Remote Site you are deleting.
Example	The following command deletes the Remote Site SITE. <code>peladm delete_site -a SITE</code>

Selecting Remote Sites: peldsp select_site

Syntax	<code>peldsp select_site [-alias] [-proto_ident] [-group] [-protocol] [comm_type] [dest_address]</code>
Description	Use this command to select the Remote Sites that match the selection criteria entered as parameters. All parameters are optional. To list all Remote Sites, enter no parameters. Enter the command again using the optional parameters to refine your selection.

Parameters

`-alias (-a)` : Enter a Site alias for the Remote Site as selection criteria.

Not for FTP or SFTP.

`-proto_ident (-pi)` : Enter a Site protocol name as selection criteria.

`-group (-gr)` : Enter the Group name of the Site or Sites you want to display.

`-protocol (-pr)` : Enter the protocol used by the Site or Sites you want to display. Enter one of the following values:

- **PEL** (Default)
 - **PHSE** = PeSIT HS E
 - **PHSD** = PeSIT HS D
 - **FTP**
 - **HTTP**
 - **FTP_HTTP**
 - **ODETTE** = OFTP
 - **SFTP**
 - **SMTP**
 - **POP3**
 - **TO_GTW**
 - **FROM_GTW**
 - **AS2**
 - **AS3**
 - **AS1_POP3**
 - **AS1_SMTP**
 - **RN_POP3**
 - **RN_SMTP**
 - **RN_HTTP**
 - **SWIFTNet**
 - **JMS**
-

`-comm_type (-ct)` : Enter the type of communications interface used by the Site or Sites you want to display. Enter one of the following values:

- PAD
- TCPIP (default)
- LU62
- DSA

`-dest_address (-da)` : Enter the destination address of the Site or Sites you want to display.

Example 1 The following command returns the list of Remote Sites that use PEL protocol and the TCP/IP network type.

```
peldsp select_site -pr PEL -ct TCPIP
```

Gateway lists the Remote Site aliases that match the selection criteria:

```
IMTPLX01
UXTPLX01
```

Example 2 Gateway handles each string parameter as a truncated search key. The following example returns the list of Remote Sites whose aliases begin with U and whose main destination address begins with 192.

```
peldsp select_site -a U -dest_address 192
```

Returns, for example:

```
UXTPDX01
UXTPDT02
UXTPLX01
```

Where the displayed results conform to the criteria.

Displaying Remote Sites: peldsp display_site

Syntax	<code>peldsp display_site {-alias}</code>
Description	Use this command to display the properties of a specific Remote Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Remote Site for which you want to display the properties.

Example	Gateway displays each parameter of the specified Site on a separate line, in the format: <code>name_of_field="value"</code> Each field name is prefixed by the string " <code>s_</code> ", referring to the Site object.

Displaying Remote Site status: `peldsp status_site`

Syntax	<code>peldsp status_site {-alias}</code>
Description	Use this command to display the initial and dynamic status of a Remote Site that you identify by its alias.
Parameters	<i>Mandatory</i> <code>-alias (-a)</code> : Enter a Site alias for the Remote Site for which you want to display the status.
Example	Gateway displays each status parameter of the specified Site on a separate line, in the format: <code>name_of_field="value"</code> Each field name is prefixed by the string " <code>s_</code> ", referring to the Site object.

Starting Remote Sites: `pelctl start_site`

Syntax	<code>pelctl start_site {-alias} [-init_mode] [-resp_mode] [-commpath_used] [-trace_mode] [-poll_mode]</code>

Description

Use this command to dynamically activate a Site. When you start a Site, its state changes from **NO** to **YES** (for a specified connection mode).

Gateway initiates no connections, Transfers, or any other operations with inactive Sites.

If the Site is not already active, the start operation activates the Site and:

- resets counters and timers for the Site, in particular:
 - current number of unsuccessful connection attempts (`Site retry_count` parameter)
 - current interval between connection attempts (`Site retry_delay` parameter)
- immediately processes all Transfer Requests with **to begin** status

If the Site is already active, the operation still resets Site counters and timers, and processes Transfer Requests with **to begin** status.

To change the initial state for a Site (that is, the default state that Gateway takes into account on startup), you must update the initial modes for that Site via the `peladm update_site` command. However, all modifications to the Site initialization modes take effect dynamically, and are lost when you restart Gateway.

Parameters	<i>Mandatory</i> <code>-alias (-a)</code>: Enter a Site alias for the Remote Site that you want to start. <code>-init_mode (-im)</code>: Enter this parameter to start the Site in Initiator Site mode. <code>-resp_mode (-rm)</code>: Enter this parameter to start the Site in Responder Site mode. <code>-commpath_used (-cu)</code>: Enter the communications path used to connect the Site. Enter one of the values: • MAIN = Main communication address • ALT1 = First alternative communication address • ALT2 = Second alternative communication address • ALT3 = Third alternative communication address <code>-trace_mode (-tm)</code>: Enter this parameter to activate the trace mode. <code>-poll_mode (-pm)</code>: Enter this parameter to activate the polling function.
Example	The following command starts the Site with the alias SITENAME in both Initiator mode and Responder mode. <pre>pelctl start_site -alias SITENAME -init_mode -resp_mode</pre>

Stopping Remote Sites: *pelctl stop_site*

Syntax	<pre>pelctl stop_site {-alias} [-init_mode] [-resp_mode] [-trace_mode] [-poll_mode]</pre>
Description	Use this command to disable the Transfer or trace activity on a Remote Site. Stopping a Site involves changing its state from YES to NO (for specified connection modes). Transfers that are currently in progress continue until they are completed (on condition that no error stops their execution). Gateway initiates no connections, transfers, or any other operations with inactive Sites. Stopping an inactive Site has no effect.

Parameters	<i>Mandatory</i>
	<code>-alias (-a)</code> : Enter a Site alias for the Remote Site that you want to start.
	<code>-init_mode (-im)</code> : Enter this parameter to disable Initiator mode for the selected Site.
	<code>-resp_mode (-rm)</code> : Enter this parameter to disable Responder mode for the selected Site.
	<code>-trace_mode (-tm)</code> : Enter this parameter to disable trace mode for the selected Site.
Example	<code>-poll_mode (-pm)</code> : Enter this parameter to disable polling for the selected Site.
	The following command disables Initiator mode and Responder mode for the Site SITENAME. <pre>pelctl stop_site -alias SITENAME -init_mode -resp_mode</pre>

Axway Gateway: Managing Partners

Site objects: Parameters List

[Standard Site parameters](#)[Communication path parameters](#)[Additional Site parameters for JMS](#)[Additional Site parameters for SWIFTNet](#)[Additional Site parameters for X.400](#)

Standard Site parameters

The following table lists standard Site command parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Meaning	Values
- acknowledgment_option (-ao) Acknowledgment option	<i>PeSIT HS E, FTP, OFTP and POP3 only</i> Acknowledgement behavior	One of the following values: • ack_by_monitor = automatically send an acknowledgment request • ack_by_user = wait for a user acknowledgment request • not_to_ack = no acknowledgment • ack_unknown = acknowledgment behavior undefined (default)
-alias (-a) Alias	Site alias	Maximum: 31 alphanumeric characters
-attach_extract_file (-aef) Attach / Extract file	<i>SMTP/POP3 only</i> Define whether or not to attach or extract a file by default.	One of: • Yes if the file is not formatted as an email • No if the file is already formatted as an email
-auth_obj_name Authentication certificate: Name	<i>OFTP R2.0 only</i> Specify the name of the authentication certificate to use.	Maximum: 639 alphanumeric characters

Parameter	Meaning	Values
-auth_obj_type Authentication certificate: Type	<i>OFTP R2.0 only</i> Specify the type of authentication certificate to use.	One of: • SECS_ALIAS • XPP_ENTITY • XPP_CERTIFICATE
-bypass_xsr (-byxsr) Bypass XSR for this site	If active, all outgoing connections to this site will be made directly, even if XSR is enabled. This setting overrides the global setting for the address masks to skip XSR in outgoing calls.	• Yes: all outgoing calls are made directly (bypassing XSR) • No: outgoing calls are made through XSR, if XSR is enabled
-change_data_addr (-cda) Allow different IP addresses for control and data sessions	<i>FTP only</i> Disables protection against FTP bounce attacks by allowing different IP addresses for control and data sessions	• Y • N
-check_login_ident (-chk1_id) <i>Depending on protocol:</i> (Pre-Connection) ID to receive from partner Username to check	<i>PeSIT, SFTP only</i> Pre-connection identifier to check from the partner	• PeSIT: Up to 8 alphanumeric characters • SFTP: Up to 24 alphanumeric characters

Parameter	Meaning	Values
-check_login_ password (- chk1_password) <i>Depending on protocol:</i> (Pre- Connection) Password to receive from partner Password to check	<i>PeSIT and SFTP</i> <i>Also for X.400 - described later in this topic</i> Pre-connection password to check from the partner	• PeSIT: Up to 8 alphanumeric characters • SFTP: Up to 14 alphanumeric characters
-check_ password (-c_ pswd) (Connection phase) Client password to check	Client connection password to check by the server	Maximum: 15 alphanumeric characters
-check_ password2 (-c_ pswd2) (Connection phase) Server password to check	<i>PeSIT HS</i> Corresponds to the password field in PI 5 of FPDU_ACKCONNECT	• PeSIT HS D: a numeric value less than 65536 • PeSIT HS E: up to 8 alphanumeric characters

Parameter	Meaning	Values
-check_window (-cw) <i>Depending on protocol:</i> Synchronization acknowledgment window Number of blocks between acknowledgments Network credit Pipeline level	• PeSIT: Synchronization acknowledgement window: number of synchronization points between two acknowledgement messages • PEL: Number of data blocks between acknowledgments • OFTP: Number of Data Exchange Buffers the sender can send before the end of file or the credit value is reached. The sender must then wait for an acknowledgement (a CDT response) before sending more DATA commands. • SFTP: Pipeline level: maximum number of outstanding buffers (non-acknowledged) that the client may send to the server before waiting for an acknowledgement	• PeSIT, PEL: Integer: 0 - 4 (0 means that no synchronization points are used.) Default: 4 • OFTP: Integer: 0 - 99 (0 means that no synchronization points are used.) • SFTP: Integer: 0 - 255 (the values 0 and 1 are equivalent.) Default: 0
-command_sensitivity (-cmd_s) Command sensitivity	<i>FTP security</i> Level of command sensitivity. This parameter allows you to define the number of commands that you can pass during a session. Each command corresponds to a certain credit, depending on how costly the command is in terms of memory and system resources. The higher the sensitivity, the fewer commands you can pass.	Integer: 1 - 5. 1 is the lowest level of sensitivity.
-comments (-cts) Comments	User comments	Maximum: 80 alphanumeric characters

Parameter	Meaning	Values
-commpath_used (-cu) <i>(None)</i>	<i>Starting a Remote Site and displaying Site status:</i> Communication path used to connect the Site	• MAIN = Main communication address • ALT1 = First backup communication address • ALT2 = Second backup communication address • ALT3 = Third backup communication address
-conn_hold_delay (-hold_delay) Connection hold delay	<i>PeSIT HS only</i> The time (in seconds) that the connection remains active without any transmission events. Use this parameter in conjunction with the parameters: • permanent_conn • heartbeat_rate	Integer
-data_size_max (-ds) Maximum data block size	Maximum data block size in bytes	Integer: 128 - 32768 (default: 4000)

Parameter	Meaning	Values
-data_window (-dw) <i>Depending on protocol:</i> Interval between synchronization points Data windows	Interval between synchronization points. Synchronization points are used to set the position in the file during the restart phase. • PeSIT: maximum number of Kbytes sent before the sending Site sends a synchronization point to the receiving Site • OFTP: maximum number of buffers sent or received before the Site puts a synchronization point into the Mailbox	• PeSIT: Integer: -1 – 999 (-1 = synchronization points not used) Default: 100 • OFTP: Integer: 1 – 999 Default: 100 (0 means that no synchronization points are used.)
-dir_path (-dp) Directory for received files	The directory of the local file management system where received files are stored.	Maximum: 127 alphanumeric characters

Parameter	Meaning	Values
-disable_ sftpv6 (-nov6) Disable SFTPv6	<i>SFTP only</i> Disable SFTPv6	One of: • Y - SFTPv6 will not be negotiated (server role - not advertised, client role - not selected) nor accepted (server role, when client is starting directly with SFTPv6, Gateway server will respond with SFTPv3). Only SFTPv3 will be used for this remote site. • N (default) - SFTPv6 is preferred (whenever available).
-eerp_style (- es) EERP style	<i>OFTP only</i> Acknowledgement style	One of: • FR = French • IN = International
-error_mail_ disposal (- emd) Mail disposition on error	<i>POP3 only</i> Behavior in case of error during transfer	One of: • undefined • leave • delete • recover (default)

Parameter	Meaning	Values
-ftpcs_list_fields (-ftpcs_lf) FTP CS List fields	In Responder mode, this parameter defines which columns will be sent in the <code>LIST</code> command to FTPCS clients. If the item described in the list is not a virtual file, the only significant fields are GTYPE, RIGHTS, FILENAME, SFILENAME, DATE and SIZE. The Gateway FTP server completes these fields. By default, the parameter lists all fields. Regardless of the field order that you specify, fields are always sent in the same predefined order. LOCID and GTYPE are required fields and are always sent to the client.	A list of one or more of the following, separated by a comma: • LOCID: local_ident • GTYPE: VFD type (real or virtual directory, mount point, real or virtual file) • RIGHTS: VFD rights • FILENAME: long name - s<xfer_ident>.<filename> for a virtual file, otherwise <filename> • SFILENAME: short name - <filename> only • APPLI: Application name • SIZE: file size • DATE: date of last modification/creation of transfer • TYPE: MVS transfer state • RSTATE: routing state (T: to route, R:

Parameter	Meaning	Values
		routed, A: acked) • UMSG: user message • ORG: origin • DEST: destination • P1: parameter 1 • P2: parameter 2 • PERM: permanent transfer • DATA_CODE: B for binary, T for test
-ftp_data_struct (-ftp_ds) Data structure	<i>FTP only</i> Data structure type	One of: • FILE = File structure type allowed (default) • NONE = No data structure type allowed • RECORD = Record structure type allowed • BOTH = File and record structure type allowed
-ftp_disable_xfb_commands (-ftp-dxc) Disable Gateway specific behavior	<i>FTP only</i> Disables the use of FTP commands specific to Gateway.	• Y • N (default)

Parameter	Meaning	Values
<code>-ftp_passive_mode (-ftp_pm)</code> Passive mode	<i>FTP only</i> Passive mode capability	One of: • NONE = No passive mode allowed • INIT = Passive command sent in Initiator mode • RESP = Passive command reception allowed • BOTH = Both <i>init</i> and <i>resp</i> passive mode (default)
<code>-ftp_restart_type (-ftp_rs)</code> Restart type	<i>FTP only</i> Restart mode	One of: • STREAM = Stream restart mode allowed • INFO = Block and compressed restart mode • BOTH = Stream, block and compressed restart modes allowed (default)

Parameter	Meaning	Values
<code>-get_appli_ method (-gam)</code> Application method	<i>FTP, HTTP, OFTP, SFTP, POP3 only</i> Method used to retrieve Application object name	One of: • AIF = Application In Filename field • AIM = Application In user Message field • AIE = Application In user Exit sub- routine (default)
<code>-group (-gr)</code> Group name	Group name	Maximum: 15 alphanumeric characters (default: GDEFAULT)
<code>-heartbeat_ rate (&#45;heartbea t)</code> Heartbeat rate	<i>PeSIT HS only</i> The interval (in seconds) between Heartbeat messages emitted by the server to confirm the presence of a file-requestor client. Use this parameter in conjunction with the parameters: • permanent_conn • conn_hold_delay	Integer: 0 - (2³¹ - 1)
<code>-init_conn_ count</code> Number of connections from this Site	<i>Displaying Remote Site status</i> Number of established connections in Initiator mode.	Integer: 0 - init_ conn_max

Parameter	Meaning	Values
-init_conn_max (-i_max) <i>Depending on protocol:</i> Maximum number of simultaneous connections to this Site Maximum number of client processes	<i>All protocols except SWIFTNet:</i> Maximum number of connections initiated to this Site. <i>POP3:</i> This value is always set to 1. <i>SWIFTNet:</i> Maximum number of simultaneous client processes allowed.	Integer: 0 - 999 (default: 1)
-init_mode (-im) Initiating Site	Set the initialization attribute to Initiator Site.	• Y (default) • N
-init_receiver (-ir) Initiator/Receiver	Receiver in Initiator mode	• Y (default) • N
-init_sender (-is) Initiator/Sender	Sender in Initiator mode	• Y (default) • N
-is_monitor (-ism) Send transfer attributes	<i>HTTP only</i> When set to Yes , Transfer CGI parameters (P_ORG, P_DEST, P_APPLI, P_MSG) are sent with the file.	• Y • N (default)
-local_certificate_alias (-lcerta) Certificate alias	<i>SFTP only</i> Certificate alias to use to authenticate your Site to the partner	Maximum: 31 alphanumeric characters
-local_private_key_alias (-lprvka) Private key alias	<i>SFTP only</i> Private key alias to use to authenticate your Site to the partner	Maximum: 31 alphanumeric characters

Parameter	Meaning	Values
-login_ident (-l_id) <i>Depending on protocol:</i> (Pre-Connection phase) ID to send to partner User name Login ID/Password to send (ID part)	PeSIT: Pre-connection identifier to send SFTP: User name to send to the Remote Site HTTP: Identifier to send during the connection phase JMS: The user name (for the MOM user) for connection to the JMS provider SWIFTNet: Client user name to send to the Remote Site during the connection phase.	- PeSIT: Maximum: 8 alphanumeric characters - SFTP: Maximum: 63 alphanumeric characters - HTTP: Maximum: 63 alphanumeric characters - JMS: Maximum: 63 alphanumeric characters - SWIFTNet: Maximum: 63 alphanumeric characters
-login_method (-lm) HTTP method	HTTP method to use with the login request	One of: - GET - POST

Parameter	Meaning	Values
-login_ password (-l_ pswd) <i>Depending on protocol:</i> (Pre-Connection phase) Password to send to partner Password ID/Password to send (Password part)	PeSIT: Pre-connection password to send SFTP: User password to send to the Remote Site HTTP: User password to send during the connection phase JMS: The password (for the MOM user) for connection to the JMS provider SWIFTNet: Password to send to the Remote Site during the connection phase X.400: <i>described later in this topic</i>	• PeSIT: Maximum: 8 alphanumeric characters • SFTP: Maximum: 127 alphanumeric characters • HTTP: Maximum: 127 alphanumeric characters • JMS: Maximum: 127 alphanumeric characters • SWIFTNet: Maximum: 20 alphanumeric characters

Parameter	Meaning	Values
-login_type (-lt) Identification method	<i>HTTP only</i> Identification method to use when transferring files	One of: • Anonymous = Sends neither <i>user</i> nor <i>password</i> • Web = standard HTTP identification method (adds <i>user</i> and <i>password</i> to HTTP headers) • CGI = adds <i>user</i> and <i>password</i> to the CGI parameters • Cookies = adds <i>user</i> and <i>password</i> to the CGI parameters of the login request. Each Transfer Request following the login request will not send those parameters but cookies received with login response.
-login_url (-lu) Login path	<i>HTTP only</i> Path to use with login request (for example, /cgi-bin/login.cgi). This is not an absolute URL like http://server:port...	Maximum: 256 alphanumeric characters

Parameter	Meaning	Values
-login_user_param (-lup) Login CGI parameters	<i>HTTP only</i> CGI parameters to add login requests. Use this parameter if you set <code>login_type</code> to <i>Cookies</i> .	Maximum: 80 alphanumeric characters Format: <code>login_param1=value1&login_param2=value2</code>
-mail_address (-email) Default email address	<i>SMTP only</i> Default email address to use if no address is specified in the Transfer Request. This parameter applies to both Remote and Local Sites: - Remote: default To address - Local: default From address	Maximum: 128 characters
-mail_xprio_alpha (-mxa) Alphanumeric XPriority header	<i>SMTP only</i> Controls the way the X-Priority header is built: - numeric (default) or - alphanumeric	Values: N (default value) X-Priority takes numeric values (1 to 5) Y X-Priority takes alphanumeric values (<i>HIGHEST</i>, <i>HIGH</i>, <i>NORMAL</i>, <i>LOW</i>, <i>LOWEST</i>)
-max_download_rate (-max_downr) Maximum downloads KB/min	<i>FTP security:</i> To reduce the risk of hammer attacks, limit the size of downloaded data per minute.	Integer (KB/min)

Parameter	Meaning	Values
-max_download_requests_rate (-maxdownreqr) Maximum downloads Requests/min	<i>FTP security:</i> To reduce the risk of hammer attacks, limit the number of download requests per minute.	Integer (number of requests /min)
-max_upload_file_size (-maxupsize) Max upload file size	<i>FTP security:</i> To reduce the risk of hammer attacks, limit the upload file size.	Integer (MB)
-max_upload_rate (-max_upr) Maximum uploads KB/min	<i>FTP security:</i> To reduce the risk of hammer attacks, limit the size of uploaded data per minute.	Integer (KB/min)
-max_upload_requests_rate (-maxupreqr) Maximum uploads Requests/min	<i>FTP security:</i> To reduce the risk of hammer attacks, limit the number of upload requests per minute.	Integer (number of requests /min)
-nack_option (-nao) Negative acknowledgement option	<i>PeSIT HS E only</i> The Nack (Negative acknowledgement) option indicates whether the Remote Site supports the negative EERP extension or not. If this option is not set, all negative acknowledgement requests are denied.	• Y • N (default)
-newline_convention (-nlc) Newline convention	<i>SFTP only</i> Line feed convention, depending on remote system type	One of: • U / u (UNIX) • W / w (Windows) • empty

Parameter	Meaning	Values
-network_security_option (-nsopt) Network security	Network Security Type to use with this Site.	One of: • NONE = No network security (default) • TLS = TLS network security • SSH = SSH network security
-ntlm_version (-ntlmv) NTLM version	<i>HTTP only</i> The NTLM version to be used for authentication to a HTTP server.	One of: • NTLM • NTLM2
-oftp_auth_method Authentication method	<i>OFTP R2.0 only</i> Specify the authentication method to use.	One of: • None • EncryptedData • EnvelopedData
-oftp_max_file_size (-oftp_mfs) Max. size for received files	<i>OFTP only</i> Enter the maximum size allowed for received files. Specify the value in KB.	Integer: 0 - 999 999 999 0 = no limit.
-old_password (-o_pswd) <i>(None)</i>	<i>PeSIT</i> Old password to specify in Initiator mode	
-old_password2 (-o_pswd2) <i>(None)</i>	<i>PeSIT HS E</i> Optional parameter corresponding to the old password field in PI 5 of FPDU_ACKCONNECT.	Maximum: 8 alphanumeric characters

Parameter	Meaning	Values
-partner_loc_alias (-pla) Partner local alias	Local Site alias to use during connection phase. If you do not complete this parameter, Gateway uses the default Local Site specified in the configuration menu.	Maximum: 25 alphanumeric characters
-password (-pswd) (Connection phase) Client password to send	<i>PeSIT</i> Connection password to send	Maximum: 15 alphanumeric characters
-password2 (-pswd2) (Connection phase) Server password to send	<i>PeSIT:</i> Corresponds to the password field in PI 5 of <code>FPDU_ACKCONNECT</code> <i>FTP:</i> Corresponds to the <code>ACCT</code> account command sent after a <code>PASS</code> command in client mode	• <i>PeSIT HS D:</i> Numeric value less than 65536 • <i>PeSIT HS E:</i> Up to 8 alphanumeric characters • <i>FTP:</i> Up to 15 alphanumeric characters
-permanent_conn (-perm_conn) Permanent connection	<i>PeSIT HS only</i> When the polling option is configured, this parameter enables you to establish permanent polled connections between server and client. Use this parameter in conjunction with the parameters: • <code>heartbeat_rate</code> • <code>conn_hold_delay</code>	• Y • N (default)
-poll_conn_count (None)	<i>Displaying Site status</i> Maximum number of established Change Direction connections	Integer: 0 - poll_conn_max

Parameter	Meaning	Values
-poll_conn_max (-pollmax) Maximum connections	<i>PeSIT HS E:</i> The maximum number of concurrent active connections allowed of Change Direction type. <i>POP3:</i> This value is always set to 1 and cannot be modified.	Integer: 0 - 999 (default: 0)
-poll_interval (-pollint) Polling interval	<i>PeSIT HS E and POP3 only</i> The polling interval defines the time (in minutes) to wait before starting the next polling cycle.	In the format <i>MM</i> (default: 0)
-poll_mode (-pm) Activated	<i>PeSIT HS E and POP3 only</i> Activates polling on the Remote Site	• Y • N
-poll_option (-poll) Change direction support	<i>PeSIT HS E only</i> The Change Direction mode indicates whether this Remote Site is polled on Gateway startup. Use this parameter to activate or deactivate the Change Direction connection dynamically.	• Y • N (default)
-pop3_auth_method POP3 authentication method	<i>POP3 only</i> Select the authentication method to use for POP3 connections	One of: • auto • basic (default) • apop
-proto_ident (-pi) Protocol identifier	<i>Not available for FTP and SFTP</i> Site protocol name	Maximum: 25 alphanumeric characters (default: <i>alias_name</i>)

Parameter	Meaning	Values
<code>-protocol (-pr)</code> Protocol	Interchange protocol	One of: • PEL (default) • PHSE = PeSIT HS E • PHSD = PeSIT HS D • FTP • HTTP • FTP_HTTP • ODETTE = OFTP • SFTP • SMTP • POP3 • TO_GTW • FROM_GTW • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_HTTP • RN_POP3 • RN_SMTP • JMS • SWIFTNET • X400
<code>-recv_xfer_active</code> <i>(None)</i>	<i>Displaying Remote Site status</i> Current number of Transfers received from the Site	Integer: 0 - <code>recv_xfer_total</code>
<code>-recv_xfer_total</code> Number of receptions	<i>Displaying Remote Site status</i> Total number of Transfers received from the Site	Integer

Parameter	Meaning	Values
<code>-remote_issuer_cert_alias (-risscerta)</code> Issuer certificate alias	<i>SFTP only</i> Issuer (CA) certificate alias	Maximum: 31 alphanumeric characters
<code>-remote_issuer_name_pattern (-rissnp)</code> Issuer name pattern	<i>SFTP only</i> Issuer (CA) name pattern	Maximum: 512 alphanumeric characters
<code>-remote_public_key_alias (-rpubka)</code> Public key alias	<i>SFTP only</i> Public key alias to use to authenticate the partner	Maximum: 31 alphanumeric characters
<code>-remote_public_key_group (-rpubkg)</code> Public key group	<i>SFTP only</i> Public key group for keys imported automatically	Maximum: 31 alphanumeric characters
<code>-remote_subject_cert_alias (-rsubca)</code> Subject certificate alias	<i>SFTP only</i> Certificate alias to use to authenticate the partner	Maximum: 512 alphanumeric characters
<code>-remote_subject_name_pattern (-rsubnp)</code> Subject name pattern	<i>SFTP only</i> Subject name pattern to use to filter accepted partner certificates	Maximum: 512 alphanumeric characters

Parameter	Meaning	Values
-resp_conn_count Number of connections accepted by this Site	<i>Displaying Remote Site status</i> Number of established connections in Responder mode	Integer: 0 - resp_conn_max
-resp_conn_max (-r_max) <i>Depending on protocol:</i> Maximum number of simultaneous connections from this Site Maximum number of server processes	Maximum number of connections from this Site. <i>POP3:</i> This value is always set to 0. <i>SWIFTNet:</i> Maximum number of simultaneous server processes allowed.	Integer: 0 - 999 (default: 1)
-resp_mode (-rm) Responding Site	Set the initialization attribute to Responder Site.	• Y (default) • N
-resp_receiver (-rr) Responder/Receiver	Receiver in Responder mode. <i>Mandatory for SMTP Sites.</i>	• Y (default) • N
-resp_sender (-rs) Responder/Sender	Sender in Responder mode. <i>Mandatory for POP3 Sites.</i>	• Y (default) • N
-restart_allowed (-restart) Restart allowed	Restart function	• Y (default) • N
-resync_allowed (-resynch) Resynchronization allowed	Resynchronization function	• Y (default) • N

Parameter	Meaning	Values
<code>-retry_count_max (-rc_max)</code> Maximum number of connection retries before time limit	Maximum number of connection retries before timeout	Integer: 0 - 999 (default: 10)
<code>-retry_count</code> Number of retries	Number of current connection retries	Integer
<code>-retry_delay</code> (None)	Current connection retry delay	Integer
<code>-retry_delay_max (-rd_max)</code> Maximum interval between retries	Maximum interval between connection retries (in seconds)	Integer: 1 - 9999 (default: 600)
<code>-retry_delay_min (-rd_min)</code> Time before first retry	Minimum interval between connection retries (in seconds)	Integer: 1 - 9999 (default: 10)
<code>-send_xfer_active</code> (None)	Number of active Transfers sent to the Site	Integer: 0 -send_xfer_total
<code>-send_xfer_total</code> Number of transmissions	Number of Transfers to be sent to the Site or that are in progress.	Integer

Parameter	Meaning	Values
-sentinel_ transfer_ filter (-stf) Sentinel Transfer filter	Set the notification level for Transfer state changes to send to the Sentinel server.	One of: • ALL • SUMMARY: Summary transfer states are: canceled, ended, acked, created, to_ begin. • NONE • UNDEFINED
-session_max Max number of sessions	<i>OFTP only</i> Maximum number of sessions (IN + OUT) for each remote site. Please note that Gateway supports per-partner granularity for selecting the template when using PassPort PM. • If a remote site with alias = <code>SSIDCODE.SIOICD+SSIDCODE.SIOORG</code> exists in Gateway, this site is selected as the template site. • If not, the default configured template site is used (<code>TODT</code>).	Integer: 0 - 999 (default : 1)
-smtp_auth_ method SMTP authentication method	<i>SMTP only</i> Select the authentication method to use for SMTP connections	One of: • NONE (default) • PLAIN • LOGIN • ANONYMOUS

Parameter	Meaning	Values
-snd_msg (-sm) User message	• OFTP: user message to be transmitted at OFTP connection phase • FTP: list of special commands to send to remote server using FTP <code>SITE</code> or <code>QUOTE</code> command • PeSIT: user message to be transmitted at the connection phase • HTTP: user message used to add CGI parameters to Transfer Requests	• OFTP: Up to 8 alphanumeric characters • PeSIT: Up to 512 alphanumeric characters • HTTP: formatted as follows: <code>xfer_param1=value1& xfer_param2=value2</code>
-special_logic Special Logic Coding	<i>OFTP only</i> Special logic coding capabilities	• Y • N (default)
-ssh_profile (-sshprf) SSH Profile	<i>SFTP only</i> SSH Profile to use in client mode	Maximum: 31 alphanumeric characters
-tls_sprof_in (-tsi) Security profile for incoming connection	<i>Optional</i> Incoming TLS Security Profile name. The network security option must be set to TLS .	Maximum: 30 alphanumeric characters
-tls_sprof_out (-tso) Security profile for outgoing connection	<i>Optional</i> Outgoing TLS Security Profile name. The network security option must be set to TLS .	Maximum: 30 alphanumeric characters
-trace_mode (-tm) Trace	Initialization attribute: enable or disable trace mode	• Y • N (default)

Parameter	Meaning	Values
-trans_mode (- trm) Transfer Mode	<i>FTP and OFTP only</i> Transfer mode capabilities • FTP: STREAM transfer mode is always available (non-modifiable) • OFTP: only COMPRESSED is available.	One of: • NONE = unused (default) • COMPRESSED = compressed mode allowed • BLOCK = block mode allowed • BOTH = compressed and block mode allowed
-user_agent (- ua) User agent	<i>HTTP only</i> Enter the ID of the client application used to contact the remote partner. This information is useful for certain logging and statistics processes.	Maximum: 128 alphanumeric characters
-username (- un) Transfer owner	If specified, defines the Transfer owner name to associate with Transfers created by Gateway. If you do not specify a value, the user name defined in the configuration menu is used.	Maximum: 31 alphanumeric characters

Parameter	Meaning	Values
<code>-use_trade_unpacking (-tradeu)</code>	<i>EDIINT only</i> Enables S/MIME, or alternatively ASx, to search in the transported file contents for security related information.	One of: • undefined = unused (default) • SMIME = enables S/MIME to unpack file • EDIINT = enables an EDIINT protocol to unpack file • RN = enables RosettaNet protocol to unpack

Parameter	Meaning	Values
-version (- ver) Version	<i>PEL and OFTP only</i> Protocol version.	<i>For PEL, one of:</i> • no value • HV = High Value • TR = Trailer • CRC = Cyclic Redundancy Code • HV/TR = High Value and Trailer • HV/CRC = High Value and Cyclic Redundancy Code • CRC/TR = Cyclic Redundancy Code and Trailer • HV/CRC/TR = High Value and Cyclic Redundancy Code and Trailer <i>For OFTP, one of:</i> • R1.2 • R1.3 (default) • R1.4 • R2.0

Parameter	Meaning	Values
-xsr_allowed_ra_list (-xsr_arl) Allowed Router Agents	Specify the Secure Relay Router Agents that are allowed to connect to this Site. Use this parameter with the <code>peladm update_site</code> command to add or remove Router Agents from the list. Separate Router Agent names with commas. To add RAs to the list, use the command: <pre>peladm update_site -a SITE_NAME -xsr_arl ADD,RA_NAME1,RA_NAME2</pre> You can omit ADD as adding RAs is the default function for this parameter. To remove RAs from the list, use the command: <pre>peladm update_site -a SITE_NAME -xsr_arl REM,RA_NAME1,RA_NAME2</pre>	ADD or REM (remove) followed by a list of Router Agent names separated by a comma. Examples: <pre>peladm update_site -a SAMPLE_SITE -xsr_arl RA1,RA2</pre> <pre>peladm update_site -a SAMPLE_SITE -xsr_arl REM,RA2</pre>

Communication path parameters

The following parameters describe the main communication path and the three backup communication paths.

To differentiate the parameters, they are marked with extension `_1`, `_2` or `_3`. For example, `comm_type_1` describes the communication type for the first backup communication type.

Parameter	Meaning	Value
-comm_options (-opt) Facilities	Network options	Syntax for: • TCP/IP: Not used • SNA LU 6.2: @lupartner@mode@tpn
-comm_type (-ct) Network type	Type of communication interface	One of: • PAD • TCPIP (default) • LU62 • DSA

Parameter	Meaning	Value
-dest_address (-da) Called address	Address to call (destination network address for the Remote Site)	Maximum: 63 alphanumeric characters. Syntax: • TCP/IP: IP address/TCP port number • SNA LU 6.2: @machine@locallu@opt
-ft_proxy (- fpx) FTP / HTTP proxy	<i>FTP/HTTP only</i> FTP or HTTP Proxy name for the main communication address	Maximum: 31 alphanumeric characters
-logon (-log) Logon message	<i>PeSIT and PEL</i> <i>protocols only</i> Message sent before protocol exchange to route the call to the right Application	• PEL: up to 31 characters • PeSIT: up to 8 characters
-logon_code (- log_co) Logon code	<i>Logon message code</i>	One of: • A = ASCII • E = EBCDIC • B = Binary • ISO = ISO • OEMPC = OEMPC • A1 = ASCII1 • A2 = ASCII2 • A3 = ASCII3 • A4 = ASCII4 • A5 = ASCII5 • E1 = EBCDIC1 • E2 = EBCDIC2 • E3 = EBCDIC3 • E4 = EBCDIC4 • E5 = EBCDIC5 • U = Undefined (default)

Parameter	Meaning	Value
-net_proxy (-npx) Network proxy	<i>SOCKS4 or HTTPS</i> Network Proxy name for the main communication address	Maximum: 31 characters
-org_address (-oa) Calling address	<i>Not applicable for TCP/IP</i> Calling address (originator network address for the Site)	Maximum: 63 alphanumeric characters. Syntax for SNA LU 6.2: @user@password@type
-user_data (-ud) User data in the call request packet	User data in the call request packet	• TCP/IP: Not used • SNA LU 6.2: Corresponds to the PIP fields in the Remote Site Network address tab. Refer to Address configuration for Remote Sites. Each of the 8 alphanumeric characters of the PIP must be hexadecimal. Example: to obtain the string: "hexa", enter "68657861"

Additional Site parameters for JMS

The following table lists Site parameters that are specific to JMS. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Description
-jms_ClassLoader Class loader	The Java class path used to locate <code>jar</code> files of the JMS provider. Enter the list of fully qualified <code>jar</code> files path separated by the ':' character on UNIX systems and ';' on Windows systems.
-jms_ConnectionFactory Connection factory	The JNDI lookup name of the connection factory to be used (configured by the JMS provider administrator).

Parameter	Description
-jms_init_ property (-jip) Property and Value	Add an initialization property for JMS connection. The syntax is <code>NewProperty=Value</code> .
-jms_ InitialFactory Initial factory	The Java class name of the Initial Context Factory. For example: <code>com.sun.jndi.fscontext.RefFSContextFactory</code>
-jms_ msgInSelector Msg In selector	The JMS selector used to filter messages coming from JMS and transferred to Gateway.
-jms_ msgInDirectory Msg In directory	The output directory used to store the data of JMS messages transferred to Gateway.
-jms_name JNDI name	The JNDI lookup name of the queue/topic to be used (configured by the JMS provider administrator).
-jms_ topicClientId Topic client Id	If topics are used, the client identifier used to subscribe to the topic.
-jms_ topicSubscription Topic subscription	If topics are used, the subscription name used to subscribe to the topic.
-jms_type Type	The type of object used by the JMS provider to store/retrieve messages for this Site. Possible values: • <code>QUEUE</code> • <code>TOPIC</code>

Parameter	Description
<code>-jms_URL</code> URL	The connection URL of the JMS provider. It usually consists of: • a protocol • a hostname • a listening port For example: <pre>tcp://localhost:4569</pre> or <pre>file:/var/mqm</pre>
<code>-remove_jms_init_property (-rjip)</code>	Remove an initialization property. Enter the name of the property to remove.

Additional Site parameters for SWIFTNet

The following table lists Site parameters that are specific to SWIFTNet. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Description
<code>-add_channel</code> New button on <i>SnF</i> sub-tab	Refer to SWIFTNet queues and output channels .
<code>-backup_site (-bks)</code> Backup sites aliases	Specify the aliases of the backup Sites to use for the current Remote Site. (Gateway checks that the specified backup Site aliases exist.) Up to 4 backup Sites can be defined, separated by space. For example, to create a nominal Site ABCD with backup Sites BKPSITE1 and BKPSITE2, enter: • <code>peladm create _site -a ABCD -bks "BKPSITE1 BKPSITE2"</code>
<code>-channel_name (-chnname)</code> Channel name	Refer to SWIFTNet queues and output channels .
<code>-channel_subset (-chnsbs)</code> Select subset...	Refer to SWIFTNet queues and output channels .

Parameter	Description
<code>-decrypt_ident_list (-dil)</code> <code>login1:password1</code> <code>login2:password2</code> <code>...</code> Decrypt ident list	List of decrypting SWIFTNet users. This is a list of user login ids with their corresponding passwords. For each user, use the : (colon) character to separate the login id and password. Use the space character to separate each login id/password pair.
<code>-event_end_point (-eep)</code> Event end point	<i>Mandatory</i> Arbitrary name allowing SAG to route received events.
<code>-message_partner (-msp)</code> Message partner	<i>Mandatory</i> Application name declared in SAG.
<code>-new_channel_name (-newcn)</code>	Refer to SWIFTNet queues and output channels.
<code>-new_queue_name (-newqn)</code>	Refer to SWIFTNet queues and output channels.
<code>-queue_name (-qname)</code> Queue name	Refer to SWIFTNet queues and output channels.
<code>-ra_config_file (-rac)</code> Alternate configuration file	Alternative configuration file for the instance of SWIFTNet RA.
<code>-ra_instance (-rai)</code> RA instance	<i>Mandatory</i> Name of the SWIFTNet RA (Remote API host adapter) local access point.
<code>-ra_pathname (-rap)</code> Alternate RA installation pathname	<i>Only if SWIFTNet RA has been installed in more than one place</i> Alternative path for SWIFTNet RA.

Parameter	Description
<code>-sign_ident_list</code> <code>(-sil)</code> <code>login1:password1</code> <code>login2:password2</code> <code>...</code> Sign ident list	List of signatory SWIFTNet users. This is a list of user login ids with their corresponding passwords. For each user, use the : (colon) character to separate the login id and password. Use the space character to separate each login id/password pair.
<code>-snf_queues (-swq)</code> Queues	List of all SnF queues to be managed.
<code>-snl_end_point (-sep)</code> SNL end point	Name allowing the reception of SnF events.
<code>-window_size (-wndsz)</code> Window size	Refer to SWIFTNet queues and output channels.
<code>-receive_retrieved_messages</code> <code>(-rcvretrmsg)</code> Receive retrieved messages	Refer to SWIFTNet queues and output channels.
<code>-sw_init_xfer_max</code> <code>(-swixm)</code> Maximum number of initiator transfers	The value of the maximum number of transfers in Initiator mode, FileAct only. By default, this parameter is set to 0, meaning unlimited. The maximum value allowed is 999. Transfers switched from a nominal site to a backup site will be subject to the nominal's site limit. The site parameter is not available to the connection related user exits.

Additional Site parameters for X.400

The following table lists Site parameters that are specific to X.400. You can use these parameters with the [peladm create site](#) and [peladm update site](#) commands when managing Remote Sites. The corresponding field name in the Gateway GUI is displayed in ***italics***.

Parameter	Description
-proto_ident (-pi) Version	Specify either MTA or MSP7 .
-login_password (-l_pswd) Password - Own	<i>For MTA:</i> Password that the local MTA uses to access the remote MTA. <i>For MSP7:</i> Password that the MS uses to authenticate itself. The MS in the connection-accept response sends the password and the P7-client validates the password before initiating any operations. Default: No authentication.
-check_login_password (-chk_l_password) Password - Partner	<i>For MTA:</i> Password that the remote MTA uses to access the local MTA. <i>For MSP7:</i> Password used by the P7-client to authenticate itself. The client at connection establishment sends the password and the MS validates the password before it accepts the connection.
-EXTERNAL_MTA_NAME External name	To have more than one remote MTA with the same MTA name, it is important to specify an external name that is to be used when contacting the MTA (MTA-MTA protocol). The MTA name specified in the X.400 MTA dialog is used locally and must be unique, whereas several MTAs can have the same external name. If no external name is specified, the MTA name specified in the X.400 MTA dialog is used.
-ADMD ADMD	If you use this parameter, the specified ADMD name is put in the originator's address if ADMD is set to an asterisk (*) when sending messages to this MTA. The reason for allowing this is that some ADMDs require the ADMD field to be set to their ADMD name when sending messages to that particular ADMD. The specified ADMD name will only be substituted if the ADMD attribute in the O/R address of the originator is set to (*). Note: ADMD substitutions should only be used when you communicate with multiple ADMDs, and only if the ADMD requires that its ADMD name be used when contacting it.

Parameter	Description
Communication parameters The parameters <code>PSAP_SELECTOR</code> , <code>SSAP_SELECTOR</code> and <code>TSAP_SELECTOR</code> identify the service access points for the different communication layers. You must specify at least one of these parameters. Typically the TSAP is used.	
-PSAP_SELECTOR PSAP	Presentation Service Access Point. This parameter defines the OSI address for the net, transport, session, and presentation layer. To specify a hexadecimal value, enter the value between apostrophes ('), for example, ' 7C , A2 '.
-SSAP_SELECTOR SSAP	Session Service Access Point. This parameter defines the SSAP. Maximum 16 characters. To specify a hexadecimal value, enter the value between apostrophes ('), for example, ' 7C , A2 '.
-TSAP_SELECTOR TSAP	Transport Service Access Point. This parameter defines the TSAP. Maximum 20 characters. To specify a hexadecimal value, enter the value between apostrophes ('), for example, ' 7C , A2 '.
-RTS_X410_MODE RTS X.410 mode	Typically, a version 84 system uses RTS X.410 mode whereas non-84 systems do not use RTS X.410 mode. Specify this option as appropriate. Possible values: • 1 - use RTS X.410 mode • 0 - do not use RTS X.410 mode (default)
-MTA_VERSION Version 84	MTA version. Typically the MTA version is greater than 84. If the version is 84, specify 84. Possible values: • 84 • 0 (default)
MTA_ADDR_... parameters When there is no explicit information on where to route a message, the destination O/R address is matched against any defined remote MTA routing address and the one which gives the best match is used.	

Parameter	Description
-MTA_ADDR.CN Routing O/R address [part]	CN part of O/R address. Note: In this case, CN refers to Country (not Common Name).
-MTA_ADDR.ADMO Routing O/R address [part]	ADMO part of O/R address.
-MTA_ADDR.PRMO Routing O/R address [part]	PRMO part of O/R address.
-MTA_ADDR.ORG_NAME Routing O/R address [part]	ORG_NAME part of O/R address.
-MTA_ADDR.ORG_UNIT_NAME_1 Routing O/R address [part]	ORG_UNIT_NAME_1 part of O/R address.
-TIME_BETWEEN_SUBMISSION Time between submission	<i>MSP7 only</i> Specifies the number of seconds between message submission. When the submission time event occurs, a check is made to see if the UA has any messages to be sent. If there are pending messages, a connection is established and all messages are sent. If this parameter is set to 0, the message submission will be done immediately after the UA has made a submission request. Default value: 0.
-MAX_INACTIVITY_TIME Max inactivity time	<i>MSP7 only</i> Specifies the number of seconds that the P7-client will keep an established MS connection up without any activity. If the parameter is set to 0, a connection is released as soon as all MS operations are made. Default value: 0.

Parameter	Description
-MAX_NR_OF_CON_ATTEMPTS Max connection attempts	<i>MSP7 only</i> Used by the P7-client when it fails to connect to the MS. The parameter states how many times a new connection attempt should be made before all messages waiting for submission will be returned to the UA as non-deliverable. Default value: 8.
-TIME_BETWEEN_CON_ATTEMPTS Time between connection attempts	<i>MSP7 only</i> Used by the P7-client when it fails to connect to the MS. The parameter specifies the time, in seconds, that the P7-client should wait before it makes a new connection attempt. Default value: 60 seconds.
-CALLING_PSAP_TSAP_SELECTOR Local TSAP	<i>MSP7 only</i> Specifies the local TSAP. Maximum 20 characters. Default value: P7. To specify a hexadecimal value, enter the value between apostrophes ('), for example, ' 7C, A2 '.
-POLL_AT_SUBMISSION Poll at submission	<i>MSP7 only</i> Determines whether or not the P7-client is allowed to retrieve messages after message submission. If selected, the P7-client is allowed to poll the Mailbox before the connection is released even if it is not yet time for the next Mailbox poll. Possible values: • 1 - selected (default) • 0 - not selected
-SUBMIT_AT_POLL Submit at poll	<i>MSP7 only</i> Determines whether or not the P7-client is allowed to submit messages after a Mailbox poll. If selected, the P7-client is allowed to submit messages before the connection is released even if it is not yet time for the next message submission. Possible values: • 1 - selected (default) • 0 - not selected

Parameter	Description
<code>-AUTO_ALERT</code> Auto alert	<i>MSP7 only</i> Determines whether or not the P7-client should request the auto alert functionality in the MS. The auto alert functionality enables the MS to immediately inform the P7-client that a new message or report has been entered into the users mailbox. When the P7-client receives an alert indication, it will immediately fetch the new message or report. Note: The alert indication is only sent on an existing connection initiated by the P7-client. The auto alert function is used only if the configuration parameter Max inactivity time is greater than 0. If selected, the auto alert function is requested. If not selected, no auto alert function is requested. Possible values: • 1 - selected • 0 - not selected (default)
Scheduling parameters	
<code>-add_ms_weekly_poll_entry (-add_mswpe)</code>	<i>MSP7 only</i> Add weekly poll entry at the latest position. Start_weekday:Stop_weekday:Start_hour:Start_min:Start_sec:Stop_hour:Stop_min:Stop_sec:Frequency_hour:Frequency_min:Frequency_sec
<code>-del_ms_weekly_poll_entry (-del_mswpe)</code>	<i>MSP7 only</i> Delete latest weekly poll entry.
<code>-update_ms_weekly_poll_entry (-upd_mswpe)</code>	<i>MSP7 only</i> Update weekly poll entry occurrence. Can contain empty parameters. Occurrence:Start_weekday:Stop_weekday:Start_hour:Start_min:Start_sec:Stop_hour:Stop_min:Stop_sec:Frequency_hour:Frequency_min:Frequency_sec
<code>-add_ms_monthly_poll_entry (-add_msmpe)</code>	<i>MSP7 only</i> Add monthly poll entry at the latest position. Month:Start_monthday:Stop_monthday:Start_hour:Start_min:Start_sec:Stop_hour:Stop_min:Stop_sec:Frequency_hour:Frequency_min:Frequency_sec

Parameter	Description
<code>-del_ms_monthly_poll_entry (-del_msmpe)</code>	<i>MSP7 only</i> Delete latest monthly poll entry.
<code>-update_ms_monthly_poll_entry (-upd_msmpe)</code>	<i>MSP7 only</i> Update monthly poll entry occurrence. Can contain empty parameters. Occurrence:Month:Start_monthday:Stop_monthday:Start_hour:Start_min:Start_sec:Stop_hour:Stop_min:Stop_sec:Frequency_hour:Frequency_min:Frequency_sec

Axway Gateway: Managing Partners

Proxy objects

[Proxies](#)

[Proxies and Gateway](#)

Proxy objects introduction

A proxy server is a machine that plays the role of buffer between the file sender and the file receiver, or between the local area network and the external network. The proxy itself controls and implements connections with the outside world. When an internal application tries to connect to the external network, the application must first connect to the proxy. The application transmits the information necessary to establish an outgoing connection (including the target network address, port number, and so on).

The proxy receives data and forwards it via a different port. This eliminates the direct path between two networks and prevents unauthorized users from obtaining internal addresses and details of a private network.

Proxy servers are available for common Internet services. For example, an HTTP proxy can provide Web access, and an SMTP proxy can provide email services access.

Proxies and Gateway

Gateway is proxy compliant and integrates functions to inter-operate with proxies.

In Gateway, you use the Proxy object exclusively in client mode for contacting a proxy server. The protocol settings of a Proxy object define the TCP address and access port for the connection server and the set of connection parameters (for example, the user name and corresponding password).

Gateway supports the following types of proxy:

- **FT proxies**
 - FTP
 - HTTP
- **NET proxies**
 - SOCKS 4
 - HTTPS

SOCKS 4 proxies

SOCKS Protocol version 4 sends a network frame that contains the final destination address, the port number, and an identifier that makes authentication to the proxy possible.

FTP proxies

You use the FTP proxy object exclusively in client mode. The protocol settings of a proxy must contain the TCP address type and address details for the connection server. This address must include the set of parameters that allow connection (for example, the user name and corresponding password).

HTTP/HTTPS proxies

To connect to an HTTPS server through a web proxy, you set the HTTP protocol and the Remote Site representing the HTTPS server to use TLS/SSL security.

HTTPS proxies:

- connect to the web proxy
- send an HTTP connect request to the remote HTTPS server
- wait for the proxy response
- initiate the TLS/SSL connection with the remote server
- send the HTTP requests

HTTP proxies:

- connect to the web proxy
- send the HTTP request formed with the complete URL (http://...) and the proxy identification parameters
- wait for the response that can come from either the web proxy or the remote web server

It is possible to configure Gateway to do tunneling for various protocols over HTTP. To do this, a proxy object of type HTTPS needs to be created and configured with the address of the remote proxy. The

type needs to be HTTPS even if the connection to the proxy is not secure (HTTP). In the remote site, the field "Network Proxy" needs to point to the previously defined proxy. In this configuration, Gateway uses the "CONNECT" method to authenticate the specified proxy.

[Working with Proxies](#)

[Working with Proxies \(command line\)](#)

[Proxies: Parameters List](#)

[Using an FTP proxy](#)

Axway Gateway: Managing Partners

Working with Proxies

[Creating a Proxy object](#)

[Displaying a list of all Proxy objects](#)

[Displaying selected Proxy objects](#)

[Viewing Proxy object attributes](#)

[Modifying a Proxy object](#)

[Deleting a Proxy object](#)

Creating a new Proxy object

To define a new Proxy object:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Right-click **Proxy...**

Gateway displays the *New Proxy* window.

3. Complete the fields in the *New Proxy* window as described in the following table.

Field	Description
Name	Enter the name of the Proxy. Maximum: 31 alphanumeric characters.
Comments	<i>Optional</i> Enter a free-text description of the Proxy. Maximum: 80 alphanumeric characters.

IP address	Enter the IP address or host name for the Proxy.	
Port	Enter the port number for the Proxy.	
User	<i>Optional</i> Enter a unique User name for the server. Maximum: 25 alphanumeric characters.	
Password	<i>Optional</i> Enter the user password. Maximum: 15 alphanumeric characters.	
Protocol	Select one of the following protocols: • FTP • HTTP • HTTPS • SOCKS4	
Type	<i>Mandatory for FTP</i> Use this field to define the connection sequence at the session level . Select one of the following types:	
	Type	Syntax
	USER_AT_SITE	USER &(user)@&(host_port) PASS &(password)
	USER_PASS_ACCT	USER &(user)@&(host):& (port) &(proxy_user) PASS &(password) ACCT &(proxy_password)
	SITE	USER &(proxy_user) PASS &(proxy_password) SITE &(host):&(port) USER &(user) PASS &(password)
	OPEN	USER &(proxy_user) PASS &(proxy_password) OPEN &(host):&(port) USER &(user) PASS &(password)
	USER_OPEN	OPEN &(host_port) USER &(user) PASS &(password)
	User script	Complete the script field.

Script	Complete this field if you selected User script as the Type. Enter the user script to execute. You can enter the script name directly. Alternatively, specify the name of a text file that the script references. For example: <code>file=proxy.txt</code> Set the end-of-line character using <code>'\n'</code> .
Allow different IP addresses for control and data sessions	The remote IP address of a proxy control session may be different from the remote address of its data session. This is not normally allowed in Gateway for protection against bounce attacks. If you check this option, FTP no longer verifies that the remote IP address is the same for the control session and data session.

Displaying a list of all Proxy objects

To display a list of all Proxy objects stored in the database:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Proxy** sub-node.

In the right panel, Gateway displays a list of all **Proxy** objects.

Displaying selected Proxy objects

If your Gateway configuration includes multiple Proxy objects, you can limit the display by applying display selection criteria:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Proxy** sub-node.

Gateway displays a list of all existing Proxy objects in the right pane.

3. Right-click the **Proxy** sub-node, then select **Select...** from the context menu.

Gateway displays the *Select Proxies* window.

4. In the *Select Proxies* window, select the name of the protocol with which you link to the Proxies you want to display and click **OK**.

Gateway regenerates the right window Proxy list, taking into account your selection criteria.

Viewing Proxy attributes

To view the attributes of an existing Proxy:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click to select **Proxy...**

Gateway displays all existing Proxies in the right (display) pane.

3. In the right pane, right-click to select the Proxy that you want to modify.

Gateway displays a context menu.

4. Select **View...** from the context menu.

Gateway opens the *Proxy (read only)* window. This window provides you with a view of all attributes for the selected Proxy.

5. When you have finished viewing the information, click **Close** to quit the window.

Modifying a Proxy object

To modify an existing Proxy:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click to select **Proxy...**

Gateway displays all existing Proxies in the right (display) pane.

3. In the right pane, right-click to select the Proxy that you want to modify.

Gateway displays a context menu.

4. Select **Modify** from the context menu.

Gateway opens the *Proxy* editing window.

5. Use the Proxy editing window to modify the Proxy characteristics and then click **OK** to confirm.

Deleting a Proxy object

To delete an existing Proxy object:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the Proxy sub-node.

Gateway displays a list of all Proxies in the right pane.

3. In the right pane, right-click the name of the Proxy you want to delete. Select **Delete...** from the context menu.

Gateway displays a confirmation dialog box.

- 4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
Alternatively, click **No** or **Cancel** to abandon the operation.
- 5. Gateway deletes the Proxy object from the database and removes the entry from the display window.

[Working with Proxies \(command line\)](#)

[Using an FTP proxy](#)

Working with Proxies (command line)

[peladm create proxy](#)

[peladm update proxy](#)

[peladm delete proxy](#)

[peldsp display proxy](#)

[peldsp select proxy](#)

Creating a Proxy: peladm create_proxy

Syntax	<code>peladm create_proxy [-name (-n)] [optional parameters]</code>
Description	Use this command to create a new Proxy.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy you are creating.
	<i>optional parameters</i> : Refer to Proxies: Parameters List .
Example	The following command creates an FTP Proxy: <code>peladm create_proxy -name FProxy -protocol FTP -comments "FTP proxy" -login_user Shepherd -login_password -type USER_AT_SITE -change_data_addr N</code>

Modifying a Proxy: peladm update_proxy

Syntax	<code>peladm update_proxy [-name (-n)] [optional parameters]</code>

Description	Use this command to modify a Proxy definition.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.
	<i>optional parameters:</i> Refer to Proxies: Parameters List .
Example	The following command modifies the properties of the Proxy named <i>Proxy1</i>. <pre>peladm update_proxy -name Proxy1 -protocol FTP -change_data_addr Y</pre>

Deleting a Proxy: peladm delete_proxy

Syntax	<code>peladm delete_proxy [-name (-n)]</code>
Description	Use this command to delete an existing Proxy.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.
Example	The following command deletes the Proxy named <i>Proxy1</i>. <pre>peladm delete_proxy -name Proxy1</pre>

Displaying a Proxy: peldsp display_proxy

Syntax	<code>peldsp display_proxy [-name (-n)]</code>
Description	This command displays the full definition of the Proxy that you specify.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.
Example	The following example displays all attributes of Proxy <i>Proxy1</i>. Enter the command: <pre>peldsp display_proxy -name Proxy1</pre> Gateway returns: <pre>pr_name='Proxy1' pr_protocol=FTP pr_address='proxy/21' pr_comments=''</pre>

Selecting Proxies: peldsp select_proxy

Syntax	<code>peldsp select_proxy [-name (-n)] [-protocol (-pr)]</code>

Description	Use this command to display a subset of proxies that matches the defined selection criteria. To list all Proxies, enter the command without any selection parameters.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.
	<code>-protocol (-pr)</code> : Name of the network layer protocol of the proxy you want to display. Enter one of the values: • SOCKS4 • FTP
Example	The following command selects all FTP Proxies. <code>peldsp select_proxy -protocol FTP</code>

[Proxies: Parameters List](#)

[Using an FTP proxy](#)

Axway Gateway: Managing Partners

Working with Proxies (command line)

[peladm create proxy](#)

[peladm update proxy](#)

[peladm delete proxy](#)

[peldsp display proxy](#)

[peldsp select proxy](#)

Creating a Proxy: `peladm create_proxy`

Syntax	<code>peladm create_proxy [-name (-n)] [optional parameters]</code>
Description	Use this command to create a new Proxy.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy you are creating.
	<i>optional parameters</i> : Refer to Proxies: Parameters List .

Example	The following command creates an FTP Proxy: <pre>peladm create_proxy -name FProxy -protocol FTP -comments "FTP proxy" -login_user Shepherd -login_password -type USER_AT_SITE -change_data_addr N</pre>

Modifying a Proxy: peladm update_proxy

Syntax	<code>peladm update_proxy [-name (-n)] [optional parameters]</code>
Description	Use this command to modify a Proxy definition.
Parameters	<code>-name (-n)</code>: Enter the name of the Proxy. <i>optional parameters</i>: Refer to Proxies: Parameters List.
Example	The following command modifies the properties of the Proxy named <i>Proxy1</i>. <pre>peladm update_proxy -name Proxy1 -protocol FTP -change_data_addr Y</pre>

Deleting a Proxy: peladm delete_proxy

Syntax	<code>peladm delete_proxy [-name (-n)]</code>
Description	Use this command to delete an existing Proxy.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.
Example	The following command deletes the Proxy named <i>Proxy1</i>. <pre>peladm delete_proxy -name Proxy1</pre>

Displaying a Proxy: peldsp display_proxy

Syntax	<code>peladm display_proxy [-name (-n)]</code>
Description	This command displays the full definition of the Proxy that you specify.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.

Example	The following example displays all attributes of Proxy <i>Proxy1</i>. Enter the command: <pre>peldsp display_proxy -name Proxy</pre> Gateway returns: <pre>pr_name='Proxy1' pr_protocol=FTP pr_address='proxy/21' pr_comments=''</pre>

Selecting Proxies: *peldsp select_proxy*

Syntax	<code>peldsp select_proxy [-name (-n)] [-protocol (-pr)]</code>
Description	Use this command to display a subset of proxies that matches the defined selection criteria. To list all Proxies, enter the command without any selection parameters.
Parameters	<code>-name (-n)</code> : Enter the name of the Proxy.
	<code>-protocol (-pr)</code> : Name of the network layer protocol of the proxy you want to display. Enter one of the values: • SOCKS4 • FTP
Example	The following command selects all FTP Proxies. <pre>peldsp select_proxy -protocol FTP</pre>

[Proxies: Parameters List](#)

[Using an FTP proxy](#)

Axway Gateway: Managing Partners

Proxies: Parameters List

The following table lists all Proxy command parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Meaning	Values
<code>address (-addr)</code>	Proxy TCP/IP address	<host>/<port>
<i>IP address</i>		

Parameter	Meaning	Values
<code>change_data_addr</code> (-cda) Allow different IP addresses for control and data sessions	Disable protection against FTP bounce attacks	Y/N
<code>comments</code> (-cts) Comments	Comment field	Maximum: 80 alphanumeric characters
<code>login_password</code> (-lp) Password	Proxy password	Maximum: 25 alphanumeric characters
<code>login_user</code> (-lu) User	Proxy user name	Maximum: 25 alphanumeric characters
<code>name</code> (-n) Name	Proxy name	Maximum: 31 alphanumeric characters
<code>protocol</code> (-pr) Protocol	Protocol for the Proxy	One of the following values: • FTP • SOCKS4
<code>script</code> (-s) Script	User script if you specified USER as the Proxy type	
<code>type</code> (-ty) Type	<i>FTP only</i> Define the connection sequence at the session level	One of the following values: • USER_AT_SITE • USER_PASS_ACCT • SITE • OPEN • USER_OPEN • USER

Managing Connection Gate objects

About CGates and CGateGroups

[Introduction](#)

[Protocol connection](#)

[CGate hierarchy](#)

[SGates and SGateGroups](#)

Introduction

This documentation refers to *Connection Gates* as *CGates*.

CGate and CGateGroup objects are identification filters used internally in server mode at the protocol connection level to control the connections of remote partners.

CGates determine whether a client has the right to connect to Gateway and exchange files with it or not. CGates are also used to assign connected user rights to the Virtual File Directory tree.

CGateGroups are used to group together a set of CGates with similar properties. For example, all CGates that use FTP protocol could belong to the same FTP CGateGroup.

Note that you cannot use the CGate identification mechanism when working with SMTP or POP3 protocols, since these protocols only function in client mode.

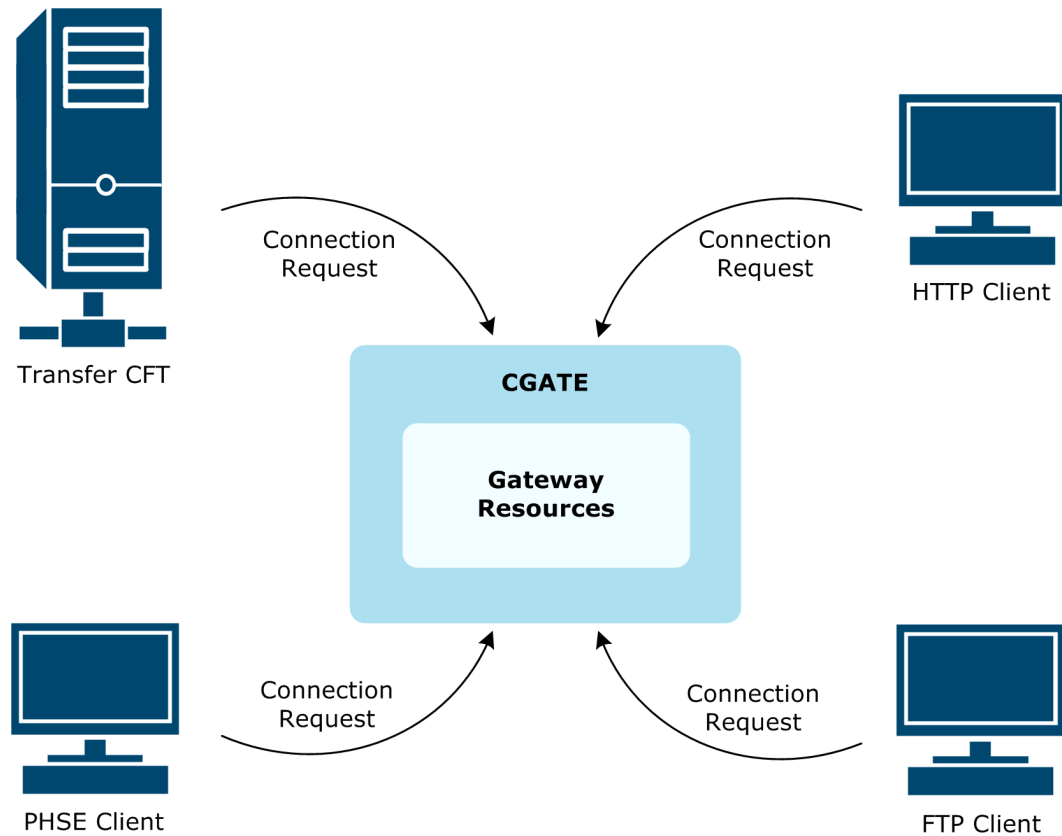
Protocol connection

If the protocol connection is successful, the CGate supplies the following information to the file transfer process for use during the current session:

- Virtual File Directory (VFD) rights for the connected user. You must assign user access rights via the CGate object for FTP, SFTP, HTTP and SHTTP transfers.
- Protocol-dependent parameters:
 - PHSE (server password to return to the client)
 - FTP (home and root directories within the VFD)
 - HTTP (*home_page*, *list_template*, home and root directories within the VFD)
- User parameters 1 and 2 for selective file visibility from the connected client side

Connection to Gateway resources

The following figure illustrates how all connections to Gateway resources pass via the CGate object:



CGate hierarchy

CGates are organized hierarchically within CGateGroups. A CGateGroup can nest other CGateGroups. However, only one CGateGroup in a given hierarchy contains CGates.

CGateGroups are used to:

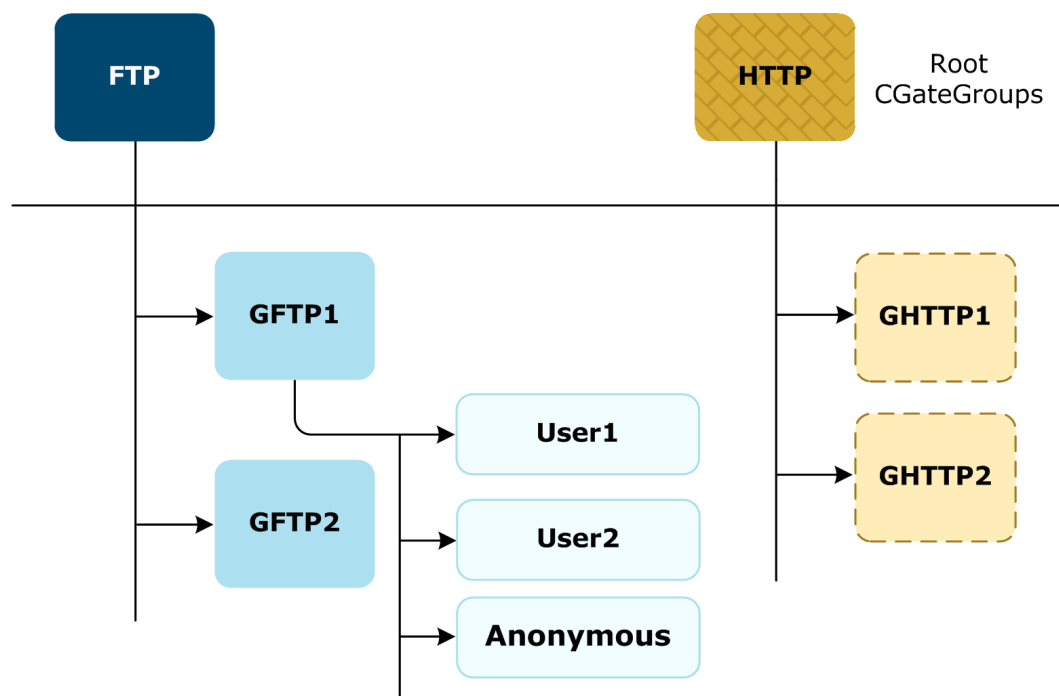
- Group CGates (that correspond to one user or one user profile), depending on technical aspects such as:
 - The protocol
 - The Called SAP (the listening SAP on the Gateway host machine) and address (Gateway may serve as a bridge between separate networks)
- Link users to a Template Site in order to use the Dynamic Site creation process, since a given hierarchy is always protocol-specific.
- Define parameter values for a set of CGates (below the current CGateGroup in the hierarchy) without duplicating the information. Certain CGate parameters are added within a given hierarchy. Others are overridden by the hierarchy.

CGates are organized in a hierarchy in which each CGate belongs to a CGateGroup. The top-level object in the hierarchy is the *root* CGateGroup. A CGateGroup can nest multiple sub-CGateGroups or CGates. However, a CGateGroup cannot contain CGateGroups *and* CGates at the same level. The CGate hierarchy is the set of objects that comprises the CGate (at the bottom of the hierarchy) and all its associated parent CGateGroups right up to the root CGateGroup.

The purpose of this hierarchical organization is to group CGates with shared properties, such as protocol or TCP/IP address. In addition, this organization allows for parameter inheritance and avoids the duplication of information.

Example

The following diagram schematically illustrates an example of CGate organization:



The hierarchical organization for CGates and CGateGroups follows these rules:

- A CGate always belongs to a single parent CGateGroup.
- A CGateGroup can contain multiple CGates or multiple sub-CGateGroups. However, a CGateGroup cannot contain both sub-CGateGroups and CGates at the same level.
- A CGateGroup that does not belong to a parent group is referred to as a *root CGateGroup*.

SGates and SGateGroups

Gateway now supports the use of [Super CGates \(SGates\)](#) that offer a higher capacity than the standard CGates.

CGates: Protocol connection phase

[Connection identification](#)

[Default protocol connection mechanism via CGates](#)

[Setting protocol connection parameters](#)

[How Gateway selects a CGate](#)

[Best matching process](#)

[Checking CGate parameters](#)

For more information on TLS security, refer to *Security guide* > [SSL and TLS protocols](#).

[Forcing CGates](#)

Connection identification

Gateway manages connection identification at the protocol connection level. Although the CGate and CGateGroup objects provide a default identification mechanism, you can customize the Protocol connection user exit [ExitRcProtoConn_C](#) to override this definition.

Protocol connection exit scheduling and its interaction with the default identification process depend on the [Connection control method](#) parameter that you set in the configuration menu. The connection identification can also be set per protocol using the [Set option per protocol...](#) option. You can select any of the following methods:

- **Monitor:** Applies no identification via CGate or exit – implements identification via static Site object
- **CGate:** Applies default identification
- **Exit:** Executes Protocol connection exit
- **CGate & Exit:** Applies CGate identification and then executes exit
- **Exit & CGate:** Executes exit and then applies CGate identification
- **Exit & SGate:** Executes exit and then applies SGate identification
- **Use global** (*only for Set option per protocol*): Use the global connection control method (as opposed to a specific connection control method for this protocol)

Use the **Exit & CGate** option if you want to [force the use of a particular CGate](#).

Default protocol connection mechanism via CGates

The default connection identification process via CGate objects is as follows:

1. Gateway first identifies a CGateGroup that corresponds to the connected user and then refines the search via the CGate objects within that group. Since CGates can contain wildcard characters (? and *), this selection is based on a best-matching process. Gateway selects a CGate whose

parameters most closely correspond to those taken from the network.

2. After selecting a CGate, Gateway then checks the protocol-dependent parameters, security parameters and so on. If the checks fail, Gateway rejects the connection.
3. Gateway then retrieves and merges the CGate hierarchy to build a set of the following output parameters:
 - protocol-dependent parameters for use while the connection is active
 - connected user rights on the VFD device
 - an optional Template Site (if the client station is not declared as a Remote Site object)
4. If the **Template Site** value is set in the CGateGroup object, Gateway uses the Template Site and the parameters received from the network to create a Dynamic Site. Otherwise, Gateway tries to find an existing Site definition that matches the parameters taken from the network. If Gateway fails to retrieve a Site, the connection is rejected.

Setting protocol connection parameters

To allow Gateway to select a CGate, you must first define the protocol connection parameters via the CGateGroup and CGate objects.

CGateGroup properties

Define the following properties in the top-level **CGateGroup**. You can only set these parameters once in a given hierarchy, and all CGates below this CGateGroup inherit these properties:

- Protocol
- HTTP host name (optional)
- TCP/IP connection SAP
- TCP/IP called address (optional)

To take a set of called addresses into account, use wildcard characters (?) and *) in the **TCP/IP called address** field.

CGate properties

Define the following properties in the **CGate** object:

- *Protocol identifiers*: Protocol identifiers can contain wildcard characters (?) and *) and are protocol-dependent. That is, the parameters that you complete for the CGate object depend on the protocol defined for the parent CGateGroup:

FTP, SFTP & HTTP	◦ login_user, login_password
PEL, OFTP	◦ client_ident, client_password
PeSIT HS D, PeSIT HS E	◦ client_ident, client_password, server_ident and the PeSIT pre-connection parameters (login_user, login_password)

- *Calling Address Control patterns*: used to specify authorized or forbidden TCP/IP calling addresses. You can define this property in both CGateGroup objects and in CGates. Since this is an additive parameter, Gateway takes both object definitions into account.

How Gateway selects a CGate

To select a CGate, Gateway first identifies a CGateGroup whose protocol, called TCP/IP SAP and optionally called TCP/IP address, matches those of the incoming connection. Within that CGateGroup, Gateway then selects a CGate object whose protocol identifiers match the connection.

Gateway then checks that the parameters taken from the network correspond to the selected CGate. It checks the following parameters:

- Protocol identifier password
- TLS security parameters (TLS profile, TLS client and server authentication)
- CGate state (must be enabled)
- Calling address and SAP

Best matching process

Since many fields may contain wildcard characters (? and *), Gateway uses a *best matching process* to identify the CGate that most closely corresponds to a given connection. Gateway applies this process to:

- Protocol identifiers
- Called address
- Called SAP

Protocol identifier matching

If CGate protocol identifiers contain wildcards, multiple CGate definitions may match the protocol parameters of a given connection. Gateway uses a *best matching process* based on the number of shared characters in the login field to identify the CGate that most closely corresponds to the incoming connection.

The following table describes the protocol identifier matching process for each protocol:

Protocol	Description
FTP, SFTP and HTTP	Gateway searches for the number of shared characters in the <code>login_user</code> parameter taken from the network (sent by the client trying to connect) and the <code>login_user</code> parameter for the current CGate object.

Protocol	Description
PEL, OFTP	Gateway searches for the number of shared characters in the <code>client_ident</code> parameter taken from the network (sent by the client trying to connect) and <code>client_ident</code> parameter taken from the current CGate object.
PeSIT HS D, PeSIT HS E	Gateway takes into account both the <code>client_ident</code> (received PI 3) and the <code>server_ident</code> (received PI 4) parameters: • If two CGate definitions contain the same <code>client_ident</code> value, Gateway selects the CGate whose <code>server_ident</code> value best matches the corresponding parameter taken from the network. • If two CGate definitions contain different <code>client_ident</code> values, Gateway selects the CGate whose <code>client_ident</code> best matches the corresponding parameter taken from the network, regardless of the respective CGate <code>server_ident</code> fields. In other words, the <code>client_ident</code> parameter has priority.

Examples

FTP / HTTP / SFTP / PEL / OFTP

If the client user login is "dupont" and if the `login_user` or `client_ident` fields for the two CGates CG1 and CG2 are respectively "du*" and "dup?nt", then CG2 has a higher number of shared characters and is selected.

PeSIT HS D / PeSIT HS E

Example 1

Given the following CGate definitions, where the `client_ident` values are different:

CG1: `client_ident` = "T*" and `server_ident` = "G*"

CG2: `client_ident` = "T?TI" and `server_ident` = "G*"

If a client sends `client_ident` / PI 3 = "TITI" and `server_ident` / PI 4 = "GATEWAY", Gateway selects the CG2 CGate object.

Example 2

Given the following CGate definitions, where the `client_ident` values are the same:

CG1: `client_ident` = "T*" and `server_ident` = "GATE*"

CG2: `client_ident = "T*" and server_ident = "G"`

If a client sends `client_ident / PI 3 = "TITI"` and `server_ident / PI 4 = "GATEWAY"`, Gateway selects the CG1 CGate object.

Example 3

Given the following CGate definitions:

CG1: `client_ident = "T*" and server_ident = "GATE*"`

CG2: `client_ident = "T?TI" and server_ident = "G"`

If a client sends `client_ident / PI 3 = "TITI"` and `server_ident / PI 4 = "GATEWAY"`, Gateway selects the CG2 CGate object (`client_ident` takes priority over `server_ident`).

Note: Since the parameters listed above serve as selection keys, keep in mind that the fields taken from the network must contain a non-empty value.

Called address matching

The Gateway host machine may be connected to more than one network and have several TCP/IP local addresses. It is therefore useful, although not mandatory, to link CGates to a specific network (IP address). To validate CGate selection and accept the connection, Gateway compares the `called_address` field (that can contain the wildcard characters ? and *), taken from the selected CGate parent groups with the `called_address` field taken from the network. If the CGateGroup field value is empty or if the two fields match, Gateway accepts the connection. Otherwise it rejects the connection.

Note: You set the **called address** field in a CGateGroup object and can only set this value once in a given hierarchy.

Called SAP matching

The TCP/IP SAP called from the client side must match one of the listener connection ports associated with the current CGate hierarchy. The `sap` field is part of the CGateGroup object and can only be set once in a given hierarchy. This field value follows the syntax given in *Checking CGate Parameters*.

Checking CGate parameters

After selecting a CGate that best matches the incoming connection, Gateway checks the following parameters:

- Protocol identifier passwords
- Calling address filter list
- TLS security parameters

Note: Each failure during the checking operation generates a message in the Gateway LOG file.

Checking protocol identifiers

Gateway checks the following protocol identifier parameters:

- **FTP, HTTP and SFTP:**login_password
- **PeSIT:**client_password (and pre-connection parameters login_user and login_password)
- **PEL, OFTP:**client_password

Note: When the corresponding CGate fields are set to asterisk (*), every value taken from the network is accepted and it is not mandatory to send these values from the client side.

Checking calling address filter list

Each CGate and CGateGroup can contain an access control list. The access control list allows you to define the IP addresses from which a defined user is authorized to connect to Gateway. Each element of this list comprises two fields:

- An IP address pattern (that can include the ? and * wildcards)
- An access flag (which can either be ALLOWED or DENIED)

During the connection process, Gateway checks the remote address provided by the client against the access control list of each member of the selected CGate hierarchy. Since the IP address pattern may contain wildcards, Gateway applies a best matching selection process (based on the number of shared characters) to each CGate or CGateGroup list. In addition, Gateway uses the best matching address pattern from the higher group of the hierarchy (from the CGateGroup nearest the root) to determine whether the connection is accepted or not.

Note: If a list is left empty, all the calling addresses are allowed by default.

Examples

Case 1: Calling address filter list in the CGate object only

Calling address taken from the network: 172.17.160.40

Calling address filter list (CGate)

- Pattern = authorization
- 172* = Denied
- 172.1?.160.* = Accepted

In this case, the connection is accepted.

Case 2: Calling address filter list in both the CGate and the parent CGateGroup

Calling address taken from the network: 172.17.160.40

Calling address filter list (CGate)

- Pattern = authorization
- 172* = Denied
- 172.1?.160.* = Accepted

Calling address filter list (parent CGateGroup)

- Pattern = authorization
- 172* = Denied

The calling address filter list in the CGateGroup object takes priority, so in this case, the connection is denied.

Checking security (TLS) parameters

When you use TLS or SSL to secure incoming connections, the TLS Handshake has already ended when the protocol connection occurs. Gateway may reject a client if the security level is lower than the expected level.

The Gateway network layer (also known as *NET*) provides three security-related parameters that Gateway checks against those found in the selected CGate definition. The following TLS-related parameters are only part of the CGate object, and not the CGateGroup object:

- `tls_client_auth`: Flag that indicates whether authentication is required for the client trying to connect. (**TLS Client Authentication** check box)
- `tls_sprof`: Security profile name used for the incoming call TLS Handshake negotiation. This field can contain wildcards (? and *). (**TLS Security profile name** field)
- `tls_sprof_user_param`: User parameter that you can complete in the security profile definition. This field can contain wildcards (? and *). (**TLS Security user parameters** field)

See the [Security Guide](#) for further information.

Forcing CGates

This feature allows you to force the selection of a specific CGate, rather than allowing Gateway to select the CGate to use for identification.

To implement this functionality, you must first go to the configuration menu and set the [Connection control method](#) to **Exit & CGate**. If you set any other connection control method, the `cgate_forced` parameter is ignored. An external process, such as access to an LDAP directory, can then be scheduled for user identification. In such a case, it is useful to force the selection of a specific CGate object and define Gateway users externally via a set of parameters that includes:

- Identification parameters (such as `user_login` and `user_password` for FTP clients)
- Links with Gateway resources profiles (such as VFD rights in the CGate/CGateGroup objects) via a single field (the CGate name)

To force CGate selection, you must set the following parameters in the Protocol connection user exit [ExitRcProtoConn_C](#) output structure (`Out_ExitRcProtoConn_t`):

- `cgate_forced`: name of the CGate object to select (a string of up to 32 characters)
- `cgate_forced_p`: indicate whether or not to take the `cgate_forced` parameter into account at the CGate selection level (unsigned char, set to 0 or 1)

Protocol connection using `cgate_forced` field

When you set the `cgate_forced_p` field in the output structure `Out_ExitRcProtoConn_t` to **1**, Gateway:

- Takes the `cgate_forced` parameter into account
- Automatically searches for the specified CGate object

If Gateway finds the specified CGate object, then it:

- Does not apply any additional selection or checking algorithm (refer to [Checking CGate parameters](#))
- Automatically selects this CGate object for the current connection
- Merges the CGate hierarchy and supplies the resulting parameters to the file transfer layer (refer to [Passing parameters to File Transfer Layer](#))
- Applies the CGate output parameters [override mechanism](#) from the Protocol connection exit, regardless of whether the CGate object selection was forced

If Gateway cannot find the specified CGate object, then the connection fails. The name provided in the `cgate_forced` field does not correspond to any existing CGate object definition known to Gateway.

Note: When you use the CGate forcing mechanism, Gateway applies no selection and checking algorithms to the forced CGate object. This means you must take the following points into account:

- The forced CGate object must exist on Gateway.
- You must manually process the optional checks that Gateway normally performs on CGate parameters. Set up these checks in the Protocol connection exit.
- If you create a Dynamic Site, the Template Site object (taken from the selected CGate hierarchy) must correspond to the current connection (protocol type, and so on). Otherwise, the connection will fail.

Axway Gateway: Managing Partners

CGates: Transfer phase

[How Gateway selects a Site](#)

[Passing parameters to File Transfer Layer](#)

How Gateway selects a Site

For the transfer phase, Gateway must select a Site object that represents the remote partner. The following three cases are possible:

- Gateway identifies a Remote Site whose protocol parameters match those taken from the network during the protocol connection phase. In this case, this Site becomes the connected Site.
- No configured Remote Site matches the client trying to connect. If correctly configured, Gateway can dynamically create a Remote Site using a template. This template is a special type of Remote Site that only plays the role of Initiator and has no defined network parameters. The template is duplicated and completed depending on the current connection being processed. The following Template Sites are automatically created on Gateway startup:
 - TPEL: PEL connections
 - TODT: OFTP connections
 - TFTP: FTP connections
 - TSFTP: SFTP connections
 - TFTP_HTTP: FTP/HTTP connections
 - TPHSE: PHSE connections
 - THTTP: HTTP connections

To enable Gateway to use this mechanism, you must set the *template_site* field of one of the CGateGroup members in the selected CGate hierarchy. Since a CGate hierarchy can comprise multiple CGateGroups, Gateway takes into account the *template_site* field value taken from the CGateGroup nearest the root.

- No defined Remote Site matching the connection is defined and no *template_site* field in the whole selected CGate hierarchy is set. Although the CGate definition is valid, the connection is rejected.

Passing parameters to File Transfer Layer

When Gateway selects a CGate (and its associated CGateGroup hierarchy), it passes a set of parameters for the current connection to the File Transfer Layer (depending on the protocol).

You can set the same parameters in several CGates and CGateGroups within a given hierarchy. In some cases all parameters are taken into account (additive parameters). In other cases, parameters are overwritten by object definitions higher up the hierarchy ([overwritten parameters](#)).

Additive parameters

The following CGate parameters are additive:

- User parameters 1 and 2: used to customize VFD logical file availability. All user parameters that you define within a given hierarchy are taken into account.
- VFD rights (described below)

Defining VFD rights

Gateway uses CGates to assign Virtual File Directory access rights to connecting users.

You can define VFD access rights in both CGates and CGateGroups. This means that you can define a set of shared rights for a group of users in the CGateGroup and can refine this definition, if required, by defining specific additional rights for a given CGate within that group.

Note: VFD rights are only meaningful for connections via FTP, SFTP or HTTP protocols.

The definition of VFD rights for a CGate (or CGateGroup) comprises the following fields:

Field	Description
Directory path	The VFD absolute path of the directory (either logical, mounted or real) to which the rights apply
VFD File pattern	File name mask – may contain wildcards (? and *). The <i>file rights</i> (see below) only apply to the files (either logical or real) that match the file mask.
Subdirectory inheritance	The current rights definition applies to the sub-directories of the directory that the <i>directory path</i> field references (and recursively).
Directory rights	Authorized operations on the current directory (referenced by the <i>directory path</i> field). You can authorize the following operations: • <i>List (L)</i>: Users are authorized to list the contents of the directory (<i>LIST FTP</i> command). • <i>Create (C)</i>: Users are authorized to create sub-directories in the current directory. • <i>Remove (R)</i>: Users are authorized to remove sub-directories from the current directory.
File rights	Authorized operations on the files located in the current directory and that match the <i>file pattern</i> field. You can authorize the following operations: • <i>Read (R)</i>: Users are authorized to read the file (via FTP or HTTP Get operation). If the <i>file pattern</i> field is empty, this operation corresponds to the user rights for downloading files. • <i>Write (W)</i>: Users are authorized to overwrite the file (via FTP or HTTP Put operation). If the <i>file pattern</i> field is empty, this operation corresponds to the user rights for uploading files. • <i>Delete (D)</i>: Users are authorized to delete the file.

Example of CGate hierarchy VFD rights

The following example shows the VFD rights defined for a CGate hierarchy that comprises a root CGateGroup **G1**, a sub-group **G1-1** and a CGate **U1**. The resolved VFD rights are associated with a user that matches the **U1** CGate definition.

	dir_path	file_pattern	dir_rights	file_rights	subdir_inheritance
CGateGroup G1	/upload	*.txt	LC	W	N
CGateGroup G1-1	/download	*.*	L	R	N
CGate U1	/upload/user1	*.*	L	none	N
	/common	*.*	LCR	RW	Y

Gateway resolves the VFD rights for CGate U1 users as follows: they inherit all VFD rights defined for CGate U1, CGateGroup G1-1 and CGateGroup G1.

Overwritten parameters

All remaining CGate parameters are overwritten within the hierarchy. Gateway always applies the parameter settings from the CGateGroup nearest the root. Some parameters are independent of the connection type and others are protocol-dependent.

Parameters	Meaning
PeSIT HS D & E specific parameters:	
server_password	Server connection partner password returned to the connection partner
HTTP, FTP, SFTP specific parameters:	
home_dir	VFD directory on which the first LIST command (HTTP or FTP) is processed just after protocol connection
root_dir	VFD root directory of the connected user. Regardless of the VFD rights defined for the current CGate, the connected user only has access to the VFD directory tree under the root directory (HTTP or FTP connections).
HTTP only:	
http_home_page	Default HTTP home page that you can reference via a symbolic link in the Gateway website HTML pages
http_list_template	VFD file path that points to an HTML page used for browsing the VFD directory tree

Example

In this example the CGate hierarchy comprises a single CGateGroup (the root) **G1** and a CGate **U1** with the following **http_home_page** field values:

G1 definition:	http_home_page: /http/download/home_page.html
U1 definition:	http_home_page: /home/index.html

The valid `http_home_page` value obtained by merging the CGate hierarchy is:
`/http/download/home_page.html` since this is the set value taken from the CGateGroup nearest the root (G1).

Substituting symbolic variables

Values for all input and output parameters (except those used for CGate selection and VFD rights) can contain symbolic variables.

Syntax

Symbolic variable syntax is **\$(VAR)** where VAR is any of the following:

`PROTOCOL`, `LOGIN_USER`, `LOGIN_PASSWORD`, `LOGIN_SAP`, `CLIENT_IDENT`, `SERVER_IDENT`, `CLIENT_PASSWORD` and `TLS_CN`

For example, if you define the user login and password fields as follows:

```
login_user=*
```

```
login_password=$(LOGIN_USER)@axway.com
```

then, for an FTP client station identification, any user name is accepted, but the password must be `<Username>` followed by the string `@axway.com`.

So if the FTP user name is **rftp**, the associated password must be `rftp@axway.com`.

Symbolic variable syntax is **&(VAR)** when VAR is either of the following:

- **ORIGINATOR: transfer protocol originator**
- **DESTINATION: transfer protocol destination**

For example:

```
route_remote_agent=&(DESTINATION)
```

```
route_local_agent= &(ORIGINATOR)
```

Note: The symbolic variable `TLS_CN` is only available if the incoming connection is secured with TLS or SSL with mutual authentication. It consists of the common name (CN) of the user. For compatibility with previous releases and protocol specifications, such as the format of the FTP login user field, this field cannot contain spaces. You must check whether the certificates used correspond to this rule.

Axway Gateway: Managing Partners

About SGates

[Connection control using SGates](#)

[SGates versus CGates](#)

[SGateGroups versus CGateGroups](#)

[When to use SGates](#)

Connection control using SGates

Gateway now supports the use of *Super CGates* (SGates) that offer a higher capacity than the standard CGates.

The following table summarizes the differences between CGates and SGates.

	CGates	SGates
Total number of supported CGates or SGates	Several thousand	Several hundred thousand
VFD rights	Maximum: 30	Maximum: 300
Selection using <i>Best matching</i> principle	Yes	No Use an exit to force the selection.

These SGates are not available by default. If you want to use SGates, you must modify a configuration file. If you have existing CGates, you can upgrade them to SGates if desired.

SGates versus CGates

The SGate object can be used instead of a CGate object in the particular case where SGate names are unambiguously selected by the Exit PCNX user routine. The *best matching* facility is not supported with SGates. Only the `cgate_forced` mechanism from Exit PCNX is currently supported.

Bearing in mind this restriction, the new SGate object can overcome the following limitations of the CGate object implementation.

Maximum number of rights per SGate

The VFD rights of each SGate are packed into a list of 16 KB records containing up to 36 rights each. Up to 65535 of these 16 KB records can exist for each SGate.

Although an SGate with a very large number of rights is possible, it is not recommended to define more than 300 rights for any SGate (approx. 10 x 16 KB records), since such large objects can very significantly increase the memory footprint of Gateway processes and disk I/O load.

For the same reason, it is advisable to ensure that the average number of rights per SGate is not significantly greater than 36 (1 x 16 KB record per SGate). In other words, the number of SGates using a list of VFD rights larger than 36 entries must be small compared to those using fewer than 36 VFD rights.

Maximum number of SGates

The fixed part of each SGate (SGate with no VFD rights) is stored as a record of about 5 KB in size. These records are gathered in a single data file that can be considered as not being limited in size given current technology (2^{47} bytes).

The SGate records are retrieved by name through a two-level hash index of $512 \times 512 = 262\,144$ entries.

Although a very large SGate file is technically possible, it is advisable, for I/O performance considerations, to avoid too many collisions in the hash index. For example, a 200 000 SGate file will normally encounter no collisions in the hash index, but a 700 000 SGate file will show about two such collisions on average.

Scalability

Unlike the CGate object management that is implemented through the SUP process (the central process of Gateway), the SGate object management is based on concurrent access to the SGate data file and VFD rights data file with proper synchronization using POSIX locking system services.

The VFD rights list is no longer read by the SUP process and passed to File Transfer process through IPC as for CGates, but read in parallel by all File Transfer processes.

This new way of concurrently handling VFD rights list accesses for SGates can also improve scalability of SGates against CGates.

SGateGroups versus CGateGroups

The SGateGroup object is the replacement of the CGateGroup object when SGates are used instead of CGates.

The same limitations that apply to SGates also apply to SGateGroups.

When to use SGates

As the current SGate implementation offers only a subset of CGate related services, it is not recommended to use SGates if the corresponding CGates meet the performance requirements of your application.

Basically, using fewer than 1000 CGates should not significantly impair Gateway performance for most applications.

Depending on the machine on which Gateway is running, and the workload of your application, performance problems can arise when the number of CGates reaches a value between 1000 and 10 000.

Switching to SGates is likely to fix the performance issue in such situations.

[Working with SGates](#)

[Working with CGates \(command line\)](#)

[Working with CGateGroups \(command line\)](#)[CGates and CGateGroups: Parameters List](#)[About CGates and CGateGroups](#)[Overview: Connection Gates](#)

Axway Gateway: Managing Partners

Working with CGates and CGateGroups

[Creating a CGateGroup or CGate](#)[Modifying a CGate](#)[Modifying a CGateGroup](#)[Viewing CGate attributes](#)[Viewing CGateGroup attributes](#)[Deleting a CGate](#)[Deleting a CGateGroup](#)

Overview

All CGate objects must belong to a CGateGroup parent object. You must therefore create the CGateGroup before you can create a CGate object.

CGateGroup and CGate objects share many of the same parameters, displayed on the following tabs:

CGateGroup tab	CGate tab	Description
General	General	Different parameters for CGate and CGateGroup objects
Output parameters	Identical parameters for both CGate and CGateGroup objects	
VFD Rights		
Calling Address Controls		
–	Security	CGate objects only

Creating new CGate and CGateGroup objects

To create a new root CGateGroup object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup...**

Gateway displays the *New CGateGroup* window. Complete this window to define the characteristics of your new CGateGroup.

To create a new CGateGroup object within another parent CGateGroup:

1. Right-click an existing CGateGroup in which you want to create a new CGateGroup.
2. Select **New > CGateGroup** from the context menu to display the *New CGateGroup* window. Complete this window to define the characteristics of your new CGateGroup.

To create a new CGate object:

1. Right-click the parent CGateGroup where you want to create the CGate.
2. Select **New > CGate** from the context menu to display the *New CGate* window. Complete this window to define the characteristics of your new CGate.

CGateGroup: General tab

Complete the fields on the CGateGroup **General** tab as follows:

Field	Description
Name	Enter a unique name for the CGateGroup. Maximum: 32 alphanumeric characters.
Parent CGateGroup	This field displays the name of the parent CGateGroup with which the new object is associated. This field is grayed out and is non-modifiable. If the new object is the root CGateGroup, this field remains blank.
Comments (optional)	Enter a free-text description for the object. Maximum: 80 alphanumeric characters.
State	Select the state of the object: • Undefined: Gateway takes into account the State definition from the next CGateGroup or CGate object in the hierarchy • Enabled: all CGateGroups and CGates below this object in the hierarchy are also enabled, regardless of the State setting for those objects • Disabled: all CGateGroups and CGates below this object in the hierarchy are also disabled, regardless of the State setting for those objects
Identification	

Protocol	Select one of the following: • All • PeSIT D • PeSIT E • FTP • HTTP • FTP/HTTP • PEL • OFTP • SFTP • AS2 • AS3 • RosettaNet HTTP <i>CGate identification does not apply to SMTP and POP3 protocols since they do not support server mode.</i>
Called address (optional)	Enter the called TCP/IP address for this CGateGroup. You can only specify this address once in a given hierarchy. Maximum: 64 alphanumeric characters, including wildcards.
Called SAP	Specify the TCP/IP local connection port. Use the following syntax: • x for port x • x:y for port x to y • x y for ports x and y • * for all SAPs
HTTP host name	<i>HTTP only</i> Specify the HTTP host name for this CGate selection. You can only specify this Host Name once in a given hierarchy. Maximum: 64 alphanumeric characters.
Use Dynamic Site	
Template Site	Select the Template Site that you want Gateway to use to create a dynamic Site.

CGate: General tab

Complete the fields on the CGate **General** tab as follows:

Field	Description
Name	Enter a unique identifier for the new object. Maximum: 32 alphanumeric characters.

Parent CGateGroup	This field displays the name of the parent CGateGroup with which the new object is associated. This field is grayed out and is non-modifiable.
Comments (optional)	Enter a free-text description for the object. Maximum: 80 alphanumeric characters.
State	Select the state of the object: • Enabled • Disabled The CGateGroup State definition overrides the value that you set for the CGate. The value that you set here is only taken into account if the CGateGroup State definition is set to Undefined.
HTTP / FTP / SFTP / PeSIT pre-connection Identification / Checking	
Login user name	Enter the user login depending on the protocol: • FTP / HTTP / SFTP: enter the user name of the Client Station • PeSIT HS: enter the USERID from the pre-connection message Maximum: 63 alphanumeric characters, including wildcards for generic pattern matching.
Change password	Define password modification rights for a user connected to this CGate: • Never: a connected user can never modify the password • Forced: a connected user must modify the password for this CGate on next connection • Allowed: a connected user can modify the password The syntax for changing the password is oldpass:newpass:newpass
Set login password	Opens a dialog box to enter the login password depending on the protocol: • HTTP/FTP/SFTP: enter the password associated with the user login field • PeSIT: enter the PASSWORD from the pre-connection message Maximum: 64 alphanumeric characters, including wildcards for generic pattern matching.
PHSE Identification / Checking	
Client ID	Enter the PeSIT client identifier. Maximum: 31 alphanumeric characters, including wildcards for generic pattern matching.

Server ID	Enter the PeSIT server identifier. Maximum: 31 alphanumeric characters, including wildcards for generic pattern matching.
Set client password	Opens a dialog box to enter the PeSIT client password. Maximum: 31 alphanumeric characters, including wildcards for generic pattern matching.

CGate: Security tab

The fields on the **Security** tab are only relevant to TLS connections and SSH connections for SFTP.

Complete the fields as follows:

Field	Description
TLS remote authentication	
TLS Client Authentication	Check this box to indicate that TLS client authentication is mandatory.
TLS profile name	Enter the incoming TLS Profile name. Maximum: 31 alphanumeric characters.
TLS profile user parameter	Specify the user parameter taken from the incoming security profile name. Maximum: 31 alphanumeric characters.
SSH remote authentication	
If you select to authenticate the partner by a certificate, you do not need to complete the Public key alias or Public key group fields. If you select to authenticate the partner by a public key, you do not need to complete the Remote certificate fields.	
Public key alias	Select a public key alias. The public key allows you to check the signature of the partner (encrypted with its private key). The only reliable way to guarantee the identity of a partner is to keep a copy of the partners public key in the Gateway database. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Public key group (optional)	Enter a group name for the public key used to authenticate the partner. Default = GDEFAULT. Maximum: 15 alphanumeric characters.
TLS (only FTP/HTTP) and SSH remote certificate filter	
Subject certificate alias	Select the certificate alias used to authenticate the partner. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.

Issuer certificate alias	Select the issuer certificate alias from the drop-down list. This is the alias of the CA certificate used to sign and issue the partner certificate. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Subject name pattern	To filter accepted Subject certificate names, you can optionally enter a pattern using wildcard characters (* and ?). For example, if you specify the name pattern: *O=Sopra,*CN=t*, Gateway accepts all certificates from the organization Sopra with a certificate name that begins with the letter t. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Issuer name pattern	To filter accepted Issuer certificate names, you can optionally enter an Issuer name pattern using wildcard characters (* and ?). Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.
Remote PassPort PS partner entity filter	
Name	When Gateway operates in server mode with a partner defined in PassPort, enter the identity of the PassPort partner.

CGate/CGateGroup: Output parameters tab

Use the **Output parameters** tab to define parameters for the current connection. The fields on this tab are identical for both CGateGroup and CGate objects.

Complete the fields as follows:

Field	Description
General information	
Root directory	<i>HTTP and FTP only</i> Select the VFD root directory of the connected user. Regardless of the VFD rights defined for the current CGate, the connected user only has access to the VFD directory tree below the root directory. Maximum: 256 alphanumeric characters. Use the Browse button to display the VFD directory tree. The value that you enter in this field is overwritten by any root directory value that you specify in a parent CGateGroup.

Home directory	<i>HTTP and FTP only</i> Select the VFD directory on which the first LIST command is processed following protocol connection. Maximum: 256 alphanumeric characters. Use the Browse button to display the VFD directory tree. The value that you enter in this field is overwritten by any home directory value that you specify in a parent CGate.
User parameters	
Param 1	User parameters 1 and 2 are additive parameters that allow you to customize VFD logical file availability and restrict file visibility to defined clients. To make a Transfer visible from a given client, the parameter settings must match one of the user parameters in the selected CGate hierarchy.
Param 2	These parameters also exist in the Gateway Mailbox record (this data is duplicated in the corresponding VFD Item or logical file). You can implicitly set these parameters from a Remote Site by using RP1=param1 and RP2= param2 as a user message in the SEND command. The outgoing Transfer inherits these parameter values from the incoming Transfer.
HTTP specific	
HTTP home page	Enter the default HTTP home page that is referenced via a symbolic link in the Gateway website HTML pages. Maximum: 256 alphanumeric characters. Use the Browse button to display the VFD directory tree. This value is overwritten by any value that you specify in a parent CGate.
HTTP list template	Enter the absolute file path that points to an HTML page used for browsing the VFD directory tree. Maximum: 256 alphanumeric characters. This value is overwritten by any value that you specify in a parent CGateGroup.
FTP quotas	
Use the quotas section to defend against hammer attacks. A hammer attack can occur when a user floods the server with requests until it can no longer respond to any other users.	
Maximum uploads KB/min	Set the maximum number of Kbytes that you can upload per minute.
Maximum uploads Requests /min	Set the maximum number of Transfer Requests that you can upload per minute.
Maximum file size	Set the maximum file size in megabytes.

Maximum downloads KB/min	Set the maximum number of Kbytes that you can download per minute.
Maximum downloads Requests /min	Set the maximum number of Transfer Requests that you can download per minute.
Command sensitivity	Move the sensitivity cursor to the level of sensitivity required. This parameter allows you to define the number of commands that you can pass during a session. Each command corresponds to a certain credit, depending on how costly the command is in terms of memory and system resources. The higher the sensitivity, the fewer commands you can pass.
PHSE specific	
Server password	<i>CGate only</i> Enter the server connection partner password that is returned to the server. Maximum: 31 alphanumeric characters.
FTPCS	

FTPCS List fields	In Responder mode, define which columns to send to FTPCS clients in the LIST command. Check the FTPCS option. Then click the FTPCS List fields button to display the list of fields that you can send to the FTPCS client. By default, all columns are listed. Regardless of the field order that you specify, fields are always sent in the same predefined order (as listed below). Check one or more of the following: • LOCID: local_ident • GTYPE: VFD type (real or virtual directory, mount point, real or virtual file) • RIGHTS: VFD rights • FILENAME: long name - s<xfer_ident>.<filename> for a virtual file. Otherwise <filename>. • SFILENAME: short name - <filename> only • APPLI: Application name • SIZE: file size • DATE: date of last modification/ creation of transfer • TYPE: MVS transfer state • RSTATE: routing state (T: to route, R: routed, A: acked) • UMSG: user message • ORG: origin • DEST: destination • P1: parameter 1 • P2: parameter 2 • PERM: permanent transfer • DATA_CODE: B for binary, T for test If the item described in the list is not a virtual file, the only significant fields are GTYPE, RIGHTS, FILENAME, SFILENAME, DATE and SIZE. The Gateway FTP server completes these fields. LOCID and GTYPE are required fields and are always sent to the client.

When receiving files, the specified directory must be relative to the home directory or root directory. If the path specified in the transfer is not present on the target VFD, the transfer is stored in home directory (or root directory if home directory is not present).

CGate/CGateGroup: VFD Rights tab

Use the **VFD Rights** tab to define user access to the Virtual File Directory tree. VFD rights are additive parameters: all VFD rights that you define within a given CGate hierarchy are taken into account. The fields on this tab are identical for both CGate and CGateGroup objects.

1. Complete the fields as follows:

Field	Description
Directory path	Use the Browse button to display the VFD directory tree and select a directory. Regardless of the VFD rights defined for the current CGate, the connected user only has access to the VFD directory tree below the root directory. Maximum: 127 alphanumeric characters.
File pattern	Use this field to further refine the VFD rights to specific file types. For example, enter <code>*.txt</code> in this field to apply the VFD rights only to <code>.txt</code> files within the specified directory.
General information	
Subdirectory inheritance	Check this box to indicate that all sub-directories inherit the parameters that you set here.
Directory rights	
Directory rights	Check one or more of the following boxes to define the type of operations users can perform on VFD directories: • List • Create • Remove
File rights	
File rights	Check one or more of the following boxes to define the type of operations users can perform on VFD files: • Read • Delete • Write

2. After completing this tab, click the **Add** button to add the VFD rights definition in the upper part of the window.

CGate/CGateGroup: Calling Address Controls tab

Use the **Calling Address Controls** tab to authorize or deny access from specific TCP/IP calling addresses. The fields on this tab are identical for both CGate and CGateGroup objects.

1. Complete the fields as follows:

Field	Description
Calling address	Enter the TCP/IP calling address for which you want to authorize or deny access. You can use wildcard characters (* and ?). Maximum: 31 alphanumeric characters.
Access allowed	Check this box to authorize access from the specified calling address. If you do not check this box, access is denied.

- After completing this tab, click the **Add** button to add your definition to the list displayed in the upper part of the window.

Modifying a CGate object

To modify an existing CGate via the GUI:

- In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > CGate

Gateway displays all existing CGateGroups in the tree structure of the left pane.

- Click to select the CGateGroup that contains the CGate you want to modify.

Gateway displays all CGates that comprise the CGateGroup in the right (display) pane.

- In the right pane, right-click to select the CGate that you want to modify.

Gateway displays a context menu.

- Select **Modify** from the context menu.

Gateway opens the *CGate* editing window.

- Use the *CGate* editing window to modify the CGate characteristics, and then click **OK** to confirm the changes.

Modifying a CGateGroup object

To modify an existing CGateGroup via the GUI:

- In the left pane of the GUI main window, click  to expand the **Partner Management** node.
- Click the **CGate** node.

Gateway lists all existing CGateGroups in the right (display) pane.

- In the right pane, right-click to select the CGateGroup that you want to modify.

Gateway displays a context menu.

- Select **Modify** from the context menu.

Gateway opens the *CGateGroup* editing window.

5. Use the *CGateGroup* editing window to modify the *CGateGroup* characteristics, and then click **OK** to confirm the changes.

Viewing *CGate* attributes

To view the attributes of an existing *CGate* object via the GUI:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > CGate

Gateway displays all existing *CGateGroups* in the tree structure of the left pane.

2. Click to select the *CGateGroup* that contains the *CGate* you want to view.

Gateway displays all *CGates* that comprise the *CGateGroup* in the right (display) pane.

3. In the right pane, right-click to select the *CGate* that you want to view.

Gateway displays a context menu.

4. Select **View** from the context menu.

Gateway opens the *CGate (read only)* window. This window provides you with a view of all attributes for the selected *CGate*.

5. After viewing the information, click **Close** to quit the window.

Viewing *CGateGroup* attributes

To view the attributes of an existing *CGateGroup* object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.

2. Click the **CGate** node. Gateway lists all existing *CGateGroups* in the right (display) pane.

3. In the right pane, right-click to select the *CGateGroup* that you want to modify.

Gateway displays a context menu.

4. Select **View** from the context menu.

Gateway opens the *CGateGroup (read only)* window. This window provides you with a view of all attributes for the selected *CGateGroup*.

5. After viewing the information, click **Close** to quit the window.

Deleting a *CGate* object

To delete an existing *CGate* object:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > CGate

Gateway displays all existing *CGateGroups* in the tree structure of the left pane.

2. Click to select the CGateGroup that contains the CGate you want to delete.

Gateway displays all CGates that comprise the CGateGroup in the right (display) pane.

3. In the right pane, right-click to select the CGate that you want to delete.

Gateway displays a context menu.

4. Select **Delete...** from the context menu.

Gateway displays a confirmation dialog box.

5. In the confirmation dialog box, click **Yes** to confirm the delete operation.

Alternatively, click **No** or **Cancel** to abandon the operation.

Gateway deletes the CGate object from the data base and removes the entry from the display window.

Deleting a CGateGroup object

Note: Before you can delete a CGateGroup, you must delete all of the CGates that the CGateGroup contains.

To delete an existing CGateGroup object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.

2. Click the **CGate** node.

Gateway lists all existing CGateGroups in the right (display) pane.

3. In the right pane, right-click to select the CGateGroup that you want to delete.

Gateway displays a context menu.

4. Select **Delete...** from the context menu.

Gateway displays a confirmation dialog box.

5. In the confirmation dialog box, click **Yes** to confirm the delete operation.

Alternatively, click **No** or **Cancel** to abandon the operation.

Gateway deletes the CGateGroup object from the database and removes the entry from the display window (right pane) and from the directory tree (left pane).

[Working with CGates \(command line\)](#)

[Working with CGateGroups \(command line\)](#)

[CGates and CGateGroups: Parameters List](#)

[About CGates and CGateGroups](#)

[Overview: Connection Gates](#)

Working with SGates

[Configuring Gateway to use SGates](#)

[Selecting the SGate name from Exit PCNX](#)

[Creating/Updating/Displaying SGates](#)

[Upgrading existing CGates to SGates](#)

[Porting and compatibility of command scripts](#)

Configuring Gateway to use SGates

The standard configuration of Gateway does not allow access to SGates by default.

You can force the display of SGates as follows:

1. Go to the directory:
`<Gateway installation directory>/run_time/etc`
2. Edit the file `pelsetup.ini`. Go to the `[monitor]` section and add one of the following lines:
`xgate_gui_support=1` to display CGates only (default mode)
`xgate_gui_support=2` to display SGates only
`xgate_gui_support=3` to display CGates and SGates
3. Restart Gateway to take this parameter into account.
4. In the configuration menu of the GUI, set the value of the [Connection control method](#) parameter to Exit & SGate.

Note: You can only set the **Connection control method** to Exit & SGate in modes 2 (SGates only) and 3 (CGates and SGates). This option is not available in mode 1 (CGates only).

Selecting the SGate name from Exit PCNX

The SGate facility requires the SGate name to be selected by the Exit PCNX (Protocol Connection user exit `ExitRcProtoConn_C`). If your application is based on *best matching*, you must make appropriate changes for it to be based on the `cgate_forced` option.

Refer to the provided Exit PCNX C sample source for information about how to use this option.

Creating/Updating/Displaying SGates

Using online commands

All sub-commands of the `peladm` and `peldsp` online commands can operate on SGates in the same way as for CGates.

For example, to create an SGate, use the `peladm create_sgate` command followed by the same argument list you would enter to create the equivalent CGate.

Using the GUI

As the SGate is intended only to solve performance issues facing a large number of SGates, using the GUI in such a situation would be very impractical. As a result, the support of SGates from the GUI is only provided for testing purposes.

You must activate SGate support from the GUI if required, as described in [Configuring Gateway to use SGates](#).

Upgrading existing CGates to SGates

To upgrade existing CGates to SGates, perform the following steps using the `pelbase export` command:

1. Export the CGateGroups to a text file.
2. Export the CGates to a text file.
3. Import the exported CGateGroups as SGateGroups.
4. Import the exported CGates as SGates.

Example

```
pelbase export_categroup -f groups.txt
pelbase export_cgate -f gates.txt
pelbase import -sgate_group -if groups.txt
pelbase import -sgate -if gates.txt
```

Porting and compatibility of command scripts

Any command, or script of commands, written either in SHELL or PERL, can be ported to SGates by changing all occurrences of the word `cgate` to `sgate` in the sub-command fields, as long as all sub-commands are not expressed using their short form.

As such a process can be very impractical and error prone, the Gateway command interpreter has been changed in order to redirect CGate related sub-commands to the SGate equivalent sub-commands.

To activate this behavior, set the environment variable `p_cmd_cgate_to_sgate` to 1.

The following example using CGate-related commands creates SGate equivalent objects (SGateGroup GFTP and SGate FTP).

```
export p_cmd_cgate_to_sgate=1

peladm create_cgategroup -n GFTP -s E -pr FTP -sap "*" -ts TFTP

peladm create_cgate -n FTP -pg GFTP -s E -lu user1 -lp user1 \
-rd "/rfd" -hd "/rfd" -vr "/rfd*:R:L:Y;/rfd*:DRW:LCR:Y"
```

An environment variable has been chosen rather than a configuration variable in order to allow any combination of the two ways of dealing with CGate/SGate compatibility issues. It allows full user control of the behavior of SHELL or PERL scripts operating on CGates, either running locally (machine running Gateway) or remotely (through the Client API or online commands).

Note: The `p_cmd_cgate_to_sgate` environment variable has no effect on the `pelbase` command, called either from SHELL or from PERL.

[About SGates](#)

[Working with CGates \(command line\)](#)

[Working with CGateGroups \(command line\)](#)

[CGates and CGateGroups: Parameters List](#)

[About CGates and CGateGroups](#)

[Overview: Connection Gates](#)

Axway Gateway: Managing Partners

Command line operations

Working with CGates (command line)

[`peladm create cgate`](#)

[`peladm delete cgate`](#)

[`peladm update cgate`](#)

[`peldsp select cgate`](#)

[`peldsp display cgate`](#)

Creating a CGate: *peladm create_cgate*

Syntax	<code>peladm create_cgate {-name} {-parent_group} [optional parameters]</code>
Description	Use this command to create a new CGate object and set its attributes. A CGate object must be linked to an existing CGateGroup object. Otherwise, the CGate object creation fails.
Parameters	<i>Mandatory</i> <code>-name (-n)</code> : Enter a unique identifying name for the CGate you are creating.
	<i>Mandatory</i> <code>-parent_group (-pg)</code> : Enter the name of the CGateGroup in which the new CGate is to be included.
	<i>Optional parameters:</i> Refer to CGates and CGateGroups: Parameters List .
Example	The following command creates a new CGate called <i>TEST</i> that belongs to CGateGroup <i>GROUP</i>. It has a user parameter with three values: g1, g2 and g3 and two VFD rights definitions: <pre> peladm create_cgate -n TEST -pg GROUP -user_param1 "g1, g2, g3" -vfd_right "/*:RW:LC:N ;/ftp:*.*:RW:L:Y" vfd_dir_path = "/" file_pattern = "" file_rights = "RW" (Read, Write) directory_rights = "LC" (List, Create) subdirectory_inheritance = "N" (No) vfd_dir_path = "/ftp" file_pattern = " *.*" file_rights = "RW" (Read, Write) directory_rights = "L" (List) subdirectory_inheritance = "Y" (Yes) </pre>

Modifying a CGate: *peladm update_cgate*

Syntax	<code>peladm update_cgate {-name} [optional parameters]</code>
Description	Use this command to modify and update the attributes of a CGate object.
Parameters	<i>Mandatory</i> <code>-name (-n)</code> : Enter a unique identifying name for the CGate you are modifying.
	<i>Optional parameters:</i> Refer to CGates and CGateGroups: Parameters List .

Example	The following command updates the CGate <i>TEST</i>. The home directory is <code>/upload/</code> and the root directory is <code>/upload</code>. The command removes the first and second VFD rights definitions and adds the following new VFD rights definitions: • VFD directory path: <code>/upload/</code> • file pattern: <code>""</code> • file rights: Read and Write • directory rights: List • sub-directory inheritance: Yes <pre>peladm update_cgate -n TEST -del_vfd_right "1; 2" -add_vfd_right "/upload/::RW:L:Y" -home_dir "/upload/" -root_dir "/upload"</pre>

Deleting a CGate: *peladm delete_cgate*

Syntax	<code>peladm delete_cgate {-name} [optional parameters]</code>
Description	Use this command to delete a CGate object.
Parameters	<i>Mandatory</i> <code>-name (-n)</code> : Enter a unique identifying name for the CGate you are deleting.
Example	The following command deletes the CGate <i>TEST</i>. <pre>peladm delete_cgate -n TEST</pre>

Selecting CGates: *peldsp select_cgate*

Syntax	<code>peldsp select_cgate [-name] [-parent_group] [-protocol] [-state] [-calling_addr] [-recursive_mode]</code>
Description	Use this command to select and display all CGate objects that match the criteria that you specify as parameters. To list all CGate objects, run the command with no parameters. You can then use the optional parameters to refine the search.

Parameters	<code>-name (-n)</code> : Enter a name as selection criteria for the display of CGates.
	<code>-parent_group (-pg)</code> : Enter the name of the CGateGroup in which the new CGate is to be included.
	<code>-recursive mode (-rm)</code> : Flag indicating whether to apply the selection only to the CGate objects directly linked to the provided CGateGroup, or recursively to the whole hierarchy under the provided CGateGroup. Enter one of the values: • Y (yes) • N (no)
	<code>protocol (-pr)</code> : one of the following values: • FTP = FTP • HTTP = HTTP • FTP_HTTP = FTP+HTTP • PHSE = PeSIT HS E • PHSD = PeSIT HS D • ODETTE = OFTP • PEL = PEL • AS2 • AS3 • SFTP • RN_HTTP • * = undefined
	<code>state (-s)</code> : Enter the state, possible values: • E = Enabled • D = Disabled
	<code>calling_addr (-cinga)</code> : Enter a Calling Address Control definition. A Calling Address definition follows the syntax: "calling_address_pattern": "authorization" • <i>calling_address_pattern</i>: Up to 31 alphanumeric characters, including wildcards • <i>authorization</i>: enumerated [A (allowed) D (denied)] Only one Calling Address Control definition can be specified.

Example	The following command selects and displays CGates whose names begin with F and that belong to the parent CGateGroup GR_TEST. Recursive mode is set to Yes to apply the selection recursively to the CGate sub-tree behind the CGateGroup GR_TEST. <pre>peldsp select_cgate -name F* -parent_group GR_TEST -recursive_mode Y</pre>

Displaying a CGate: peldsp display_cgate

Syntax	<code>peldsp display_cgate {-name}</code>
Description	Use this command to display the attributes of a CGate that you identify by its name.
Parameters	<i>Mandatory</i> <code>-name (-n)</code> : Enter a name as selection criteria for the display of CGates.

Example

Gateway returns a display of each parameter associated with the specified CGate object on one line in the form:

```
name_of_field(_optional counter)="value"
```

For example:

```
cg_file_rights_1='RW'
```

Each field name is prefixed with the string "cg_", related to the **CGate** object.

Gateway displays an optional counter field when you display fields that have multiple instances within a single CGate object. This property applies to VFD Rights, User Parameter 1 and 2 and the Calling Address Control.

When, for example you enter the command:

```
peldsp display_cgate -n TEST
```

Gateway may return information similar to the following:

```
cg_name='TEST'
cg_comments=''
cg_state='E'
cg_parent_group='GROUP'
cg_login_user='*'
cg_login_password=''
cg_change_password_option='N'
cg_client_ident='*'
cg_client_password=''
cg_server_ident='*'
cg_tls_sprof=''
cg_tls_client_auth='N'
cg_tls_sprof_user_param=''
cg_root_dir=''
cg_home_dir=''
cg_http_home_page=''
cg_http_list_template=''
cg_server_password=''
cg_user_param1_1='g1'
cg_user_param1_2='g2'
cg_user_param1_3='g3'
cg_rights_nb='2'
cg_path_1='/'
cg_file_pattern_1=''
cg_subdir_inheritance_1='N'
cg_file_rights_1='RW'
cg_dir_rights_1='LC'
cg_path_2='/ftp'
cg_file_pattern_2='*.*'
cg_subdir_inheritance_2='Y'
cg_file_rights_2='RW'
```

	cg_dir_rights_2='L'

[CGates and CGateGroups: Parameters List](#)

Working with CGateGroups (command line)

[peladm create cgategroup](#)

[peladm delete cgategroup](#)

[peladm update cgategroup](#)

[peldsp select cgategroup](#)

[peldsp display cgategroup](#)

Creating a CGateGroup: peladm create_cgategroup

Syntax	peladm create_cgategroup {-name} [<i>optional parameters</i>]
Description	Use this command to create a new CGateGroup object and set its attributes. A CGateGroup is optionally linked as a child object to an existing CGateGroup object. Otherwise, it is considered to be the root CGateGroup.
Parameters	<i>Mandatory</i> -name (-n) : Enter a unique identifying name for the CGateGroup you are creating.
	<i>Optional parameters:</i> Refer to CGates and CGateGroups: Parameters List .

Example	The following command creates a new FTP CGateGroup object named <i>TEST</i>. Since the new CGateGroup has no parent group, it is a root CGateGroup. <pre> peladm create_cgategroup -n GR_TEST -protocol FTP -template_site TFTP -sap "*" -user_param1 "g1, g2, g3" -vfd_right "/*:RW:LC:N ;/ftp:*.*:RW:L:Y" </pre> The created CGateGroup is valid regardless of the called SAP (sap = "*"). It has a user parameter 1 with three values ("g1", "g2", "g3") and the following two VFD rights definitions: <pre> vfd_dir_path = "/" file_pattern = "" file_rights = "RW" (Read, Write) directory_rights = "LC" (List, Create) subdirectory_inheritance = "N" (No) vfd_dir_path = "/ftp" file_pattern = " *.*" file_rights = "RW" (Read, Write) directory_rights = "L" (List) subdirectory_inheritance = "Y" (Yes) </pre>

Modifying a CGateGroup: *peladm update_cgategroup*

Syntax	peladm update_cgategroup {-name} [<i>optional parameters</i>]
Description	Use this command to modify and update a CGateGroup object.
Parameters	<i>Mandatory</i> -name (-n) : Enter a unique identifying name for the CGateGroup you are modifying.
	<i>Optional parameters:</i> Refer to CGates and CGateGroups: Parameters List.
Example	The following command updates the CGateGroup object <i>GR_TEST</i>, assigning <i>root_dir</i> and <i>home_dir</i> parameters. <pre> peladm update_cgategroup -n GR_TEST -home_dir "/upload/" -root_dir "/upload" </pre>

Deleting a CGateGroup: *peladm delete_cgategroup*

Syntax	peladm delete_cgate {-name} [<i>optional parameters</i>]

Description	Use this command to delete a CGateGroup object that you specify by name. Note that the CGateGroup must be empty, (contain no CGates or child CGateGroups), before you can delete it.
Parameters	<i>Mandatory</i> <code>-name (-n)</code> : Enter a unique identifying name for the CGateGroup you are deleting.
Example	The following command deletes the CGateGroup object "GR_TEST". <pre>peladm delete_cgategroup -n GR_TEST</pre>

Selecting CGateGroups: *peldsp select_cgategroup*

Syntax	<code>peldsp select_cgategroup [-name] [-parent_group] [-recursive_mode]</code>
Description	Use this command to select CGateGroup objects that match the criteria entered as parameters. To list all CGateGroups, run the command with no parameters. You can then use the optional parameters to refine the search.
Parameters	<code>-name (-n)</code> : Enter a name as selection criteria for the display of CGatesGroups. This parameter can contain wildcard characters ("?" or "*"). <i>Mandatory</i> <code>-parent_group (-pg)</code> : Enter the name of the CGateGroup in which the new CGate is to be included. <code>-recursive_mode (-rm)</code> : Flag indicating whether to apply the selection only to the CGateGroup objects directly linked to the provided CGateGroup, or recursively to the whole hierarchy under the provided CGateGroup. Enter one of the values: • Y (yes) • N (no)
Example	The following command selects and displays CGatesGroups whose names begin with <i>G</i> and that belong to the parent CGateGroup <i>GR_TEST</i>. Recursive mode is set to Yes to apply the selection recursively to the CGateGroup sub-tree behind the CGateGroup <i>GR_TEST</i>. <pre>peldsp select_cgateGroup -name G?_TEST -parent_group GR_TEST -recursive_mode Y G1_TEST G2_TEST G3_TEST</pre>

Displaying a CGateGroup: `peldsp display_cgategroup`

Syntax	<code>peldsp display_cgategroup {-name}</code>
Description	Use this command to display the properties of a CGateGroup that you specify by name.
Parameters	<i>Mandatory</i> <code>-name (-n)</code> : Enter a name as selection criteria for the display of CGateGroups.
Example	Gateway returns a display of each parameter associated with the specified CGateGroup object. Gateway displays the results on the user screen in the format displayed on one line in the form. <pre>name_of_field(_optional counter)="value"</pre> For example: <pre>cgg_file_rights_1='RW'</pre> Each field name is prefixed with the string <code>cgg_</code>, related to the CGate object. Gateway displays an optional counter field when you display fields that have multiple instances within a single CGateGroup object. This property applies to VFD Rights, User Parameter 1 and 2 and the Calling Address Control. When, for example you enter the command: <pre>peldsp display_cgategroup -n GR_B</pre> Gateway returns information similar to the following: <pre>peldsp display_cgategroup -n GR_B cgg_name='GR_B' cgg_comments='10' cgg_state='U' cgg_parent_group='GR_A' cgg_protocol='*' cgg_sap='' cgg_called_addr='' cgg_http_host_name='' cgg_root_dir='' cgg_home_dir='' cgg_http_home_page='' cgg_http_list_template='' cgg_template_site='' cgg_rights_nb='0'</pre>

[CGates and CGateGroups: Parameters List](#)

CGates and CGateGroups: Parameters List

The following table lists all CGate and CGateGroup command parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in *italics*.

Most parameters are shared by the two object types. When they are not shared, the relevant object is indicated in parentheses: (CGATE) or (CGATEGROUP). In addition, when the parameter is shared, the way it is inherited within a given hierarchy is specified in brackets:

- [S]: single definition in given hierarchy
- [O]: overwritten by hierarchy
- [A]: additive
- [H]: hidden

Parameter	Meaning	Values
add_calling_addr (-acinga) (None)	Modifying CGates/ CGateGroups List of Calling Address Controls to add to the CGate or CGateGroup being updated	Each rights definition follows the syntax: "calling_address_pattern": "authorization" where: • <i>calling_address_pattern</i>: Up to 31 alphanumeric characters, including wildcards • <i>authorization</i>: enumerated [A (allowed) D (denied)] This parameter can contain multiple calling address control definitions, each separated by the semi-colon character (;).
add_calling_addr_file (-acingaf) (None)	Modifying CGates/ CGateGroups Text file that contains the Calling Address Controls to add to the CGate or CGateGroup being updated	Each line must contain a Calling Address Control definition following the syntax described above.

Parameter	Meaning	Values
add_vfd_right (-avr) (None)	Modifying CGates/ CGateGroups List of VFD rights to add to the CGate or CGateGroup being updated	Each rights definition follows the syntax: <VFD_dir_path>: <file_pattern>: <file_rights>: <directory_rights>: <subdir_ inheritance> where: • VFD_dir_path: Up to 127 alphanumeric characters • file_pattern: Up to 127 alphanumeric characters • file_rights: One to three alphanumeric characters, any combination of R, W, D • directory_rights: One to three alphanumeric characters, any combination of R, C, L • subdir_inheritance: enumerated [Y N] This parameter can contain multiple rights definitions, each separated by the semi-colon character (;).
add_vfd_right_file (-avrfile) (None)	Modifying CGates/ CGateGroups Text file that contains the VFD rights to add to the CGate or CGateGroup being updated	Each line must contain a rights definition that follows the syntax described above.

Parameter	Meaning	Values
called_addr (-ceda) (CGATEGROUP) [S] <i>Called address</i>	Called address field for CGate selection.	Maximum 64 alphanumeric characters, including wildcards
calling_addr (-cinga) <i>Calling address</i>	List of Calling Address Controls to associate with the CGate or CGateGroup being created	Each rights definition follows the syntax: "calling_address_pattern": "authorization" • <i>calling_address_pattern</i>: Up to 31 alphanumeric characters, including wildcards • <i>authorization</i>: enumerated [A (allowed) D (denied)] This parameter can contain multiple Calling Address Control definitions, each separated by the semi-colon character (;).
calling_addr_file (-cingaf) (None)	Text file that contains the Calling Address Controls to associate with the CGate or CGateGroup being created	Each line must contain a Calling Address Control definition following the syntax described above

Parameter	Meaning	Values
<code>change_password_option (-cpo)</code> (CGATE) <i>Change password</i>	Defines password modification rights for a user connected to this CGate	One of: • N = Never: a connected user can never change the password of this CGate • A = Allowed: a connected user can change the password of this CGate • F = Forced: a connected user must change the password of this CGate on first connection When the password is changed, the value of this parameter automatically switches to A .
<code>client_ident (-ci)</code> (CGATE) <i>Client ID</i>	<i>PeSIT HS E only</i> Client connection partner identifier	Maximum 31 alphanumeric characters, including wildcards for generic pattern matching
<code>client_password (-cp)</code> (CGATE) <i>Client password</i>	<i>PeSIT HS E only</i> Client connection partner password	Maximum 31 alphanumeric characters, including wildcards for generic pattern matching

Parameter	Meaning	Values
command_sensitivity (-cmd_s) Command sensitivity	<i>FTP security</i> Level of command sensitivity. This parameter allows you to define the number of commands that you can pass during a session. Each command corresponds to a certain credit, depending on how costly the command is in terms of memory and system resources. The higher the sensitivity, the fewer commands you can pass.	Integer: 1 - 5. 1 is the lowest level of sensitivity.
comments (-cts) Comments	Comments associated with the CGate or CGateGroup object	Maximum 80 alphanumeric characters
del_calling_addr (-dcinga) (None)	Modifying CGates/CGateGroups List of Calling Address Controls to remove from the CGate or CGateGroup being updated	Each Calling Address Controls definition is identified by an integer (internally set by Gateway) which you can obtain by displaying the contents of the CGateGroup object. This parameter can contain multiple Calling Address Controls definition identifiers, each separated by the semi-colon character (;).
del_vfd_right (-dvr) (None)	Modifying CGates/CGateGroups List of VFD rights to remove from the CGate or CGateGroup being updated	Each rights definition is identified by an integer (internally set by Gateway) which you can obtain by displaying the contents of the CGate or CGateGroup object. This parameter can contain multiple rights definitions identifiers, each separated by the semi-colon character (;).

Parameter	Meaning	Values
ftpcs_list_fields (-ftpcs_lf) FTP CS List fields	In Responder mode, this field defines which columns will be sent in the LIST command to FTPCS clients. If the item described in the list is not a virtual file, the only significant fields are GTYPE, RIGHTS, FILENAME, SFILENAME, DATE and SIZE. The Gateway FTP server completes these fields. By default, the parameter lists all fields. Regardless of the field order that you specify, fields are always sent in the same predefined order. LOCID and GTYPE are required fields and are always sent to the client.	A list of one or more of the following, separated by a comma: • LOCID: local_ident • GTYPE: VFD type (real or virtual directory, mount point, real or virtual file) • RIGHTS: VFD rights • FILENAME: long name - s<xfer_ident>.<filename> for a virtual file. Otherwise <filename>. • SFILENAME: short name - <filename> only • APPLI: Application name • SIZE: file size • DATE: date of last modification/ creation of transfer • TYPE: MVS transfer state • RSTATE: routing state (T: to route, R: routed, A: acked) • UMSG: user message • ORG: origin • DEST: destination • P1: parameter 1 • P2: parameter 2 • PERM: permanent transfer • DATA_CODE: B for binary, T for test

Parameter	Meaning	Values
home_dir (-hd) [O] Home directory	<i>HTTP, SFTP and FTP only</i> VFD directory on which the first LIST command is processed just after the protocol connection	Maximum 256 alphanumeric characters
http_home_page (-hnp) [O] HTTP home page	<i>HTTP only</i> Default HTTP home page that can be referenced through a symbolic link in the Gateway website HTML pages	Maximum 256 alphanumeric characters
http_host_name (-hhn) (CGATEGROUP) [S] HTTP host name	<i>HTTP only</i> HTTP host name field for CGate selection	Maximum 64 alphanumeric characters
http_list_template (-hlt) [O] HTTP list template	<i>HTTP only</i> Absolute file path pointing at an HTML page used for browsing the VFD directory tree	Maximum 256 alphanumeric characters
login_user (-lu) (CGATE) Login user name	- PeSIT HS: USERID from pre-connection message - FTP, SFTP and HTTP: the user name of the Client Station	Maximum 63 alphanumeric characters, including wildcards for generic pattern matching
login_password (-lp) (CGATE) Login password	- PeSIT HS: PASSWORD from pre-connection message - FTP, SFTP and HTTP: the password associated with the user login field	Maximum 25 alphanumeric characters, including wildcards for generic pattern matching
max_download_rate (-max_downr) Maximum downloads KB/min	<i>FTP security</i> To reduce the risk of hammer attacks, limit the size of downloaded data per minute.	Integer (MB/min)

Parameter	Meaning	Values
<code>max_download_requests_rate (-maxdownreqr)</code> Maximum downloads Requests/min	<i>FTP security</i> To reduce the risk of hammer attacks, limit the number of download requests per minute.	Integer (number of requests /min)
<code>max_upload_rate (-max_upr)</code> Maximum uploads KB/min	<i>FTP security</i> To reduce the risk of hammer attacks, limit the size of uploaded data per minute.	Integer (MB/min)
<code>max_upload_requests_rate (-maxupreqr)</code> Maximum uploads Requests/min	<i>FTP security</i> To reduce the risk of hammer attacks, limit the number of upload requests per minute.	Integer (number of requests /min)
<code>max_upload_file_size</code> Maximum upload file size	<i>FTP security</i> To reduce the risk of hammer attacks, limit the upload file size.	Integer (MB)
<code>name (-n)</code> Name	CGate or CGateGroup unique identifier	Maximum 21 alphanumeric characters
<code>parent_group (-pg)</code> Parent CGateGroup	CGate or CGateGroup parent group name	Maximum 21 alphanumeric characters

Parameter	Meaning	Values
protocol (-pr) (CGATEGROUP) [S] <i>Protocol</i>	CGateGroup associated protocol	One of: • FTP = FTP • HTTP = HTTP • FTP_HTTP = FTP+HTTP • PHSE = PeSIT HS E • PHSD = PeSIT HS D • ODETTE = OFTP • PEL = PEL • AS2 • AS3 • SFTP • RN_HTTP • * = undefined
recursive_mode (-rm) (None)	Selecting a CGate/ CGateGroup Flag indicating whether to apply the selection only to the CGate objects directly linked to the provided CGateGroup, or recursively to the whole hierarchy under the provided CGateGroup.	One of: • Y (yes) • N (no)
remote_issuer_ cert_alias (-risscerta) (CGATE) <i>Issuer certificate alias</i>	Remote certificate filter: Certificate alias of the entity that issued and signed the subject certificate	Maximum 31 alphanumeric characters
remote_issuer_ name_pattern (-rissnp) (CGATE) <i>Issuer name pattern</i>	Remote certificate filter: Pattern to use to filter Issuer names.	Use wildcard characters (? and *)

Parameter	Meaning	Values
remote_public_key_alias (-rpubka) (CGATE) Public key alias	SSH remote authentication: Alias of the public key used to authenticate the partner	Maximum 31 alphanumeric characters
remote_public_key_group (-rpubkg) (CGATE) Public key group	SSH remote authentication: Group name for the public key	Maximum 15 alphanumeric characters
remote_subject_cert_alias (-rsubcerta) (CGATE) Subject certificate alias	Remote certificate filter: Alias of the partner certificate	
remote_subject_name_pattern (-rsubnp) (CGATE) Subject name pattern	Remote certificate filter: Name pattern to filter accepted subject certificates. For example: *O=Sopra,* CN=t*	Can contain wildcard characters (* and ?)
root_dir (-rd) [O] Root directory	<i>HTTP, SFTP and FTP only</i> VFD root directory of the connected user. Regardless of the VFD rights definition for the current CGate, the connected user only has access to the VFD directory tree under the root directory.	Maximum 256 alphanumeric characters

Parameter	Meaning	Values
sap (<i>sap</i>) (CGATEGROUP)[S] <i>Called SAP</i>	TCP/IP local connection port	TCP/IP port with the following syntax: • 'x' for port x • 'x:y' for port x to y • 'x y' for port x and y Example: '23000:23010 24000 24002 25000:25010' '*' = all SAPs
server_ident (- si) (CGATE) <i>Server ID</i>	<i>PeSIT HS E only</i> Server connection partner identifier	Maximum 31 alphanumeric characters, including wildcards for generic pattern matching
server_password (-sp) (CGATE) <i>Server password</i>	<i>PeSIT HS E only</i> Server connection partner password returned to the connection partner	Maximum 31 alphanumeric characters
state (-s) <i>State</i>	CGate or CGateGroup state	One of: • E = enabled • D = disabled • U = undefined (for CGateGroups only)
template_site (-ts) (CGATEGROUP) [S] <i>Template Site</i>	Alias of Template Site that represents the requesting client station	Maximum 31 alphanumeric characters
tls_client_auth (-tlscauth) (CGATE) <i>TLS Client Authentication</i>	<i>TLS connections only</i> Indicates whether or not client authentication is mandatory	• Y • N
tls_sprof (-tlss) (CGATE) <i>TLS profile name</i>	<i>TLS connections only</i> Incoming security profile name	Maximum 31 alphanumeric characters

Parameter	Meaning	Values
<code>tls_sprof_user_param (-tlssup)</code> (CGATE) <i>TLS profile user parameter</i>	<i>TLS connections only</i> User parameter taken from the incoming security profile name.	Maximum 31 alphanumeric characters
<code>user_param1 (-usp1)</code> <code>user_param2 (-usp2)</code> [A] <i>Param 1 / Param 2</i>	User parameters used to customize VFD file availability and restrict visibility of routed files to defined clients.	Maximum 40 alphanumeric characters

Parameter	Meaning	Values
vfd_right (-vr) [A] (None)	List of VFD rights to associate with the CGate or CGateGroup that you create.	Each rights definition follows the syntax: <pre><VFD_dir_path>: <file_pattern>: <file_rights>: <directory_rights>: <subdir_inheritance></pre> where: • VFD_dir_path: Up to 127 alphanumeric characters • file_pattern: Up to 127 alphanumeric characters • file_rights: One to three alphanumeric characters, any combination of R, W, D • directory_rights: One to three alphanumeric characters, any combination of R, C, L • subdir_inheritance: enumerated [Y N] This parameter can contain multiple rights definitions, each separated by the semi-colon character (;).
vfd_right_file (vrf) [A] (None)	Text file that contains the VFD rights to associate with the CGate or CGateGroup that you create.	Each line must contain a rights definition that follows the syntax described above.

About diffusion lists

Introduction

A file transfer involves the exchange of a single file from an originating computer to one or more destination computers (Sites). Gateway enables you to create lists of computers that you want to use as destinations for a given type of file. When you create a Transfer Request, you can then specify the list as the destination. Gateway attempts to deliver the file to each computer on the list.

Diffusion List object

The *Diffusion List object* enables you to use a Transfer Request to send a single file to multiple destinations.

A Diffusion List object comprises:

- One or more destination Sites and/or one or more Diffusion Lists, **or**
- A protocol-level destination, if the Diffusion List is associated with a Template Site. However it is only useful to associate a Template Site with a Diffusion List if you are using Gateway in Responder mode.

When you submit a Transfer Request in which you have entered a Diffusion List name as the destination alias, Gateway:

- Creates a parent Request (Diffusion List Request)
- Creates one Transfer Request for each Site referenced in the Diffusion List
- Displays  next to the Transfer identifier in the Transfer list (right pane)

The destination is either a Site (including the Sites referenced by nested lists), or a protocol-level destination (when associated with a Template Site).

To associate multiple protocols with the same Diffusion List, you can combine the Diffusion List functions with a Model object.

Using a template site with a diffusion list

When you first start Gateway, it creates a set of default, protocol-specific Template Sites. A Template Site enables you to take into account a client workstation that is not specifically declared in Gateway as a Site. You should only associate a Template Site with a Diffusion List if you are using Gateway in Responder mode.

If you specify a Template Site for the Diffusion List, all elements in the List are exclusively protocol-level destinations. This means that you cannot use a nested list. Since Sites considered to be protocol-level destinations do not necessarily exist, if you create or modify the List using a Template Site, Gateway performs no checks to determine whether the Sites exist.

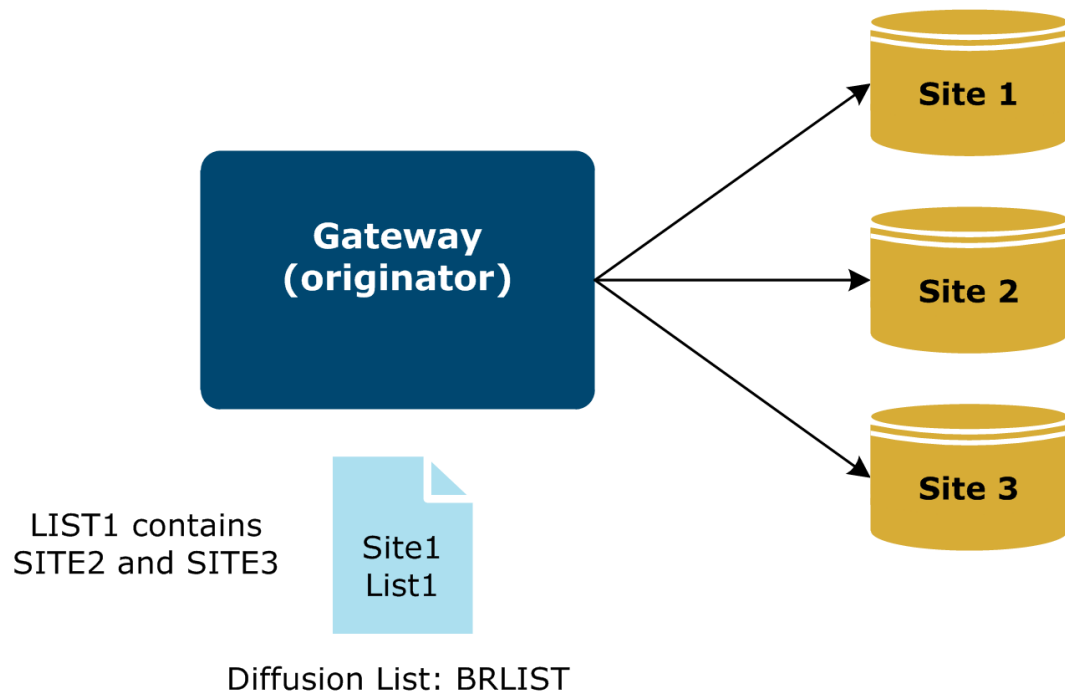
Example

- Create a Diffusion List BRLIST.
- Use the list as the destination alias in a Transfer Request.

Gateway transfers the file to each Site of BRLIST (SITE1) as well as to each Site contained in each list of BRLIST (SITE2 and SITE3).

The Gateway Mailbox records the following events:

- One transfer to the destination BRLIST (generic record)
- One transfer to the destination SITE1
- One transfer to the destination SITE2
- One transfer to the destination SITE3



[Working with Diffusion Lists](#)

[Working with Diffusion Lists \(command line\)](#)

[Diffusion Lists: Parameters List](#)

Axway Gateway: Managing Partners

About diffusion lists

Introduction

A file transfer involves the exchange of a single file from an originating computer to one or more destination computers (Sites). Gateway enables you to create lists of computers that you want to use as destinations for a given type of file. When you create a Transfer Request, you can then specify the list as the destination. Gateway attempts to deliver the file to each computer on the list.

Diffusion List object

The *Diffusion List object* enables you to use a Transfer Request to send a single file to multiple destinations.

A Diffusion List object comprises:

- One or more destination Sites and/or one or more Diffusion Lists, **or**
- A protocol-level destination, if the Diffusion List is associated with a Template Site. However it is only useful to associate a Template Site with a Diffusion List if you are using Gateway in Responder mode.

When you submit a Transfer Request in which you have entered a Diffusion List name as the destination alias, Gateway:

- Creates a parent Request (Diffusion List Request)
- Creates one Transfer Request for each Site referenced in the Diffusion List
- Displays  next to the Transfer identifier in the Transfer list (right pane)

The destination is either a Site (including the Sites referenced by nested lists), or a protocol-level destination (when associated with a Template Site).

To associate multiple protocols with the same Diffusion List, you can combine the Diffusion List functions with a Model object.

Using a template site with a diffusion list

When you first start Gateway, it creates a set of default, protocol-specific Template Sites. A Template Site enables you to take into account a client workstation that is not specifically declared in Gateway as a Site. You should only associate a Template Site with a Diffusion List if you are using Gateway in Responder mode.

If you specify a Template Site for the Diffusion List, all elements in the List are exclusively protocol-level destinations. This means that you cannot use a nested list. Since Sites considered to be protocol-level destinations do not necessarily exist, if you create or modify the List using a Template Site, Gateway performs no checks to determine whether the Sites exist.

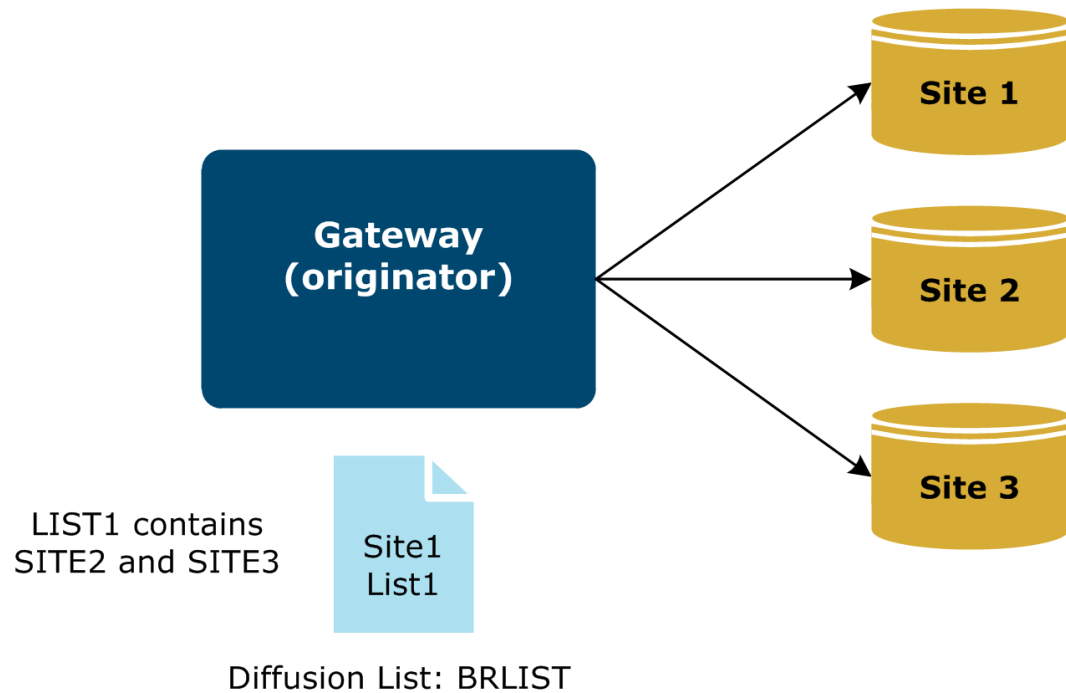
Example

- Create a Diffusion List BRLIST.
- Use the list as the destination alias in a Transfer Request.

Gateway transfers the file to each Site of BRLIST (SITE1) as well as to each Site contained in each list of BRLIST (SITE2 and SITE3).

The Gateway Mailbox records the following events:

- One transfer to the destination BRLIST (generic record)
- One transfer to the destination SITE1
- One transfer to the destination SITE2
- One transfer to the destination SITE3



[Working with Diffusion Lists](#)

[Working with Diffusion Lists \(command line\)](#)

[Diffusion Lists: Parameters List](#)

Axway Gateway: Managing Partners

Working with Diffusion Lists

[Creating a new Diffusion List object](#)

[Displaying a list of all Diffusion List objects](#)

[Displaying a list of selected Diffusion List objects](#)

[Displaying the attributes of a specified Diffusion List object](#)

[Modifying a Diffusion List object](#)

[Deleting a Diffusion List object](#)

Creating a new Diffusion List

To create a new Diffusion List object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Diffusion List** sub-node.
Gateway displays a context menu.
3. From the context menu, select **New**.
Gateway displays the *New Diffusion List* dialog window. This window comprises two tabs.
4. Complete the fields of the **General** tab using the following table.

Field	Description
Name	Enter the Diffusion List name. Maximum: 31 alphanumeric characters.
Model	Enter the name of the default Model object attached to the List. Maximum: 26 alphanumeric characters. Alternatively, select a Model from the drop-down list.
Group	Enter the name of the Application group associated with the List. Maximum: 16 alphanumeric characters. The default Group name is GDEFAULT .
Template Site	Select a Template Site to associate with the List. If you select a template in this field, Gateway displays the Destination list tab.
Comments	Enter a free-text description for the List. Maximum: 80 alphanumeric characters.

5. If you did **not** select a Template Site in the **General** tab, the second tab of the *New Diffusion List* window is the **List Selection** tab. Use the **List Selection** tab to select the Sites and/or Lists you want to include in your Diffusion List.

The **List Selection** tab has two panes:

Pane	Description
Selected item	The Selected item panel displays the lists and sites that you select in the Available items tab. These are the items that comprise your new Diffusion List. To add an item to the Selected item pane, use the Available items pane below. To remove an item from the Selected item pane, click the item to select it, then click the Unselect button.
Available items	The Available items pane displays all existing Diffusion Lists and Sites that you can use to compose your new Diffusion List. To add an item to the new Diffusion list, click to select the item in the Available items pane. Then click the Select button. Gateway adds the item to the list of the Selected items pane.

6. If you selected a Template Site in the **General** tab, Gateway displays the **Destination list** tab in place of the **List Selection tab** described in the previous step.
- You associate a Template Site with a Diffusion List only if you are using Gateway in Responder mode.

Use the **Destination list** tab to create one or more target destinations for the transfer of a single file responder-mode using a template. Remote clients can then retrieve files made available to them by connecting to the local Gateway using a name from the list.

The following table describes the components of the **Destination list** tab.

Pane	Description
Add (data field)	Enter a remote user name to use to retrieve an available file from your local Gateway
Add (button)	Click the Add button to add the name of the above data field to the list of remote users
Result display pane	This pane displays the list of remote users that exist for the selected template
Delete (button)	To delete an element from the list of users in the Result display pane, select the item in the pane and then click Delete

7. When you have finished composing your Diffusion List, click **OK** to validate and close the window.

Displaying all Diffusion List objects

To display a list of all Diffusion List objects stored in the database:

- In the left pane of the GUI main window, click  to expand the **Partner Management** node.
- Click the **Diffusion List** sub-node.
Gateway displays a list of all existing Diffusion List objects in the right (display) pane.

Displaying a list of selected Diffusion List objects

You can reduce the number of Diffusion List objects listed in the display pane by applying selection criteria to the display.

To display a selected list of Diffusion List objects:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Diffusion List** sub-node.
Gateway displays a list of all existing Diffusion List objects in the right pane.
3. Right-click the **Diffusion List** sub-node, then select **Select...** from the context menu.
Gateway displays the *Select Diffusion Lists* window.
4. In the *Select Diffusion Lists* window, use the data fields to enter criteria for Diffusion List display selection. You can use one or both of the following selection criteria:
 - Maximum number of lists (default = 50)
 - List Name
5. Click **OK** to accept the display selection criteria.
Gateway regenerates the right window Diffusion List display, taking into account your selection criteria.

Viewing Diffusion List object attributes

To view the attributes of a specific Remote Site:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Diffusion List** sub-node.
Gateway displays a list of all existing Diffusion List objects in the right (display) pane.
3. In the right pane, right-click the name of the Diffusion List object you want to view. Select **View...** from the context menu.
Gateway displays the *Diffusion List (read only)* window. This window comprises two tabs that provide you with a view of all attributes for the selected Diffusion List.
4. When you have finished viewing the information, click **Close** to quit the window.

Modifying a Diffusion List object

To modify the attributes of an existing Diffusion List object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Diffusion List** sub-node.
Gateway displays a list of all existing Diffusion List objects in the right (display) pane.
3. In the display frame (right frame), right-click the Diffusion List object that you want to modify.
Then in the context menu click **Modify...**
Gateway opens the *Diffusion List* editing window. This window comprises two tabs.
4. Modify the value fields of the tabs according to your requirements.
5. Click **OK** to confirm changes and complete the modification procedure.

Deleting a Diffusion List object

To delete an existing Diffusion List object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Diffusion List** sub-node.
Gateway displays a list of all existing Diffusion List objects in the right (display) pane.
3. In the right pane, right-click the name of the Diffusion List object you want to delete. Select **Delete...** from the context menu.
Gateway displays a confirmation dialog box.
4. In the confirmation dialog box, click **Yes** to confirm the delete operation.
Alternatively, click **No** or **Cancel** to abandon the operation.
Gateway deletes the Diffusion List object from the database and removes the entry from the display window.

[Overview: Diffusion Lists](#)

[Working with Diffusion Lists \(command line\)](#)

Axway Gateway: Managing Partners

Working with Diffusion Lists (command line)

[`peladm create list`](#)

[`peladm update list`](#)

[`peladm delete list`](#)

[peldsp select list](#)

[peldsp display list](#)

Creating a Diffusion List: *peladm create_list*

Syntax	peladm create_list {-list_name} [-group] [-comments] [-model] [-site_file] [-sites] [-add_sites] [-del_sites] [-template_site]
Description	Use this command to create a Diffusion List. The <i>string</i> syntax followed by the * character is used to search for all Sites that begin with this string. Note: Processing of -sites and -add_sites parameters is always performed before processing -del_site (delete). You can simplify List creation by pointing to a text file that comprises all the elements to include in the Diffusion List. When you create a Diffusion List, the Sites or Lists you refer to must exist, unless you use a Template Site. In this case, the destination Sites are protocol-level destinations. No check is performed to determine whether they exist.
Parameters	<i>Mandatory</i> -list_name (-n) : Enter a name for the Diffusion List object you are creating. -group (-gr) : Enter a name for the name of the Application group associated with the Diffusion List. -comments (-cts) : Enter a comment about the Diffusion List. -model (-ml) : Enter the name of a default Model that you want to associate with the Diffusion List. -site_file (-sf) : Enter the name of a text file that contains a set of Sites, protocol-level destinations and/or Diffusion Lists to use as targets for the Diffusion List. -sites (-s) : Enter names of Sites, protocol-level destinations or other Diffusion Lists to use as targets for the Diffusion List. -add_sites (-a) : Enter the names of additional Sites to add to the Diffusion List. -del_sites (-d) : Enter the names of Sites that you want to remove from the Diffusion List. -template_site (-ts) : Enter the name of a Template Site to associate with the Diffusion List.

Example	The following command creates a Diffusion List object named <i>LIST1</i> with a default template (MODELA) and several target destinations. <pre> peladm create_list -list_name LIST1 -comments "First List" -model MODELA -sites "SITEA SITEB SITEC LIST11 ASITE* BSITE*" </pre> The destinations defined for this Diffusion are: • SITEA, SITEB, SITEC • LIST11 and the ASITE* and BSITE* generic Sites Note: The Sites and Diffusion List you select when creating a Diffusion List already exist.

Modifying a Diffusion List: peladm update_list

Syntax	<pre> peladm update_list {-list_name} [-group] [-comments] [-model] [-site_file] [-sites] [-add_sites] [-del_sites] [-template_site] </pre>
Description	Use this command to modify and update a Diffusion List. The Diffusion List name (-list_name) value identifies the Diffusion List to modify. Enter new values of optional the parameters you want to change.
Parameters	<i>Mandatory</i> -list_name (-n) : Enter a name for the Diffusion List object you want to modify -group (-gr) : Enter a name of the Application group associated with the Diffusion List -comments (-cts) : Enter a comment about the Diffusion List -model (-ml) : Enter the name of a default Model that you want to associate with the Diffusion List -site_file (-sf) : Enter the name of a text file that contains a set of Sites, protocol-level destinations and/or Diffusion Lists to use as targets for the Diffusion List -sites (-s) : Enter names of Sites, protocol-level destinations or other Diffusion Lists to use as targets for the Diffusion List -add_sites (-a) : Enter the names of additional Sites to add to the Diffusion List -del_sites (-d) : Enter the names of Sites that you want to remove from the Diffusion List -template_site (-ts) : Enter the name of a Template Site to associate with the Diffusion List

Example	The following command: • updates LIST1 with a default template (MODELB) • replaces the existing elements in the Diffusion List with SITE1 • adds SITE2 and LIST <pre>peladm update_list -n LIST1 -m1 MODELB -s "SITE1" -a "SITE2 LIST"</pre>

Deleting a Diffusion List: peladm delete_list

Syntax	<code>peladm delete_list {-list_name}</code>
Description	Use this command to delete a specified Diffusion List.
Parameters	<i>Mandatory</i> <code>-list_name (-n)</code>: Enter a name for the Diffusion List object you want to delete.
Example	The following command deletes the Diffusion List <i>LIST1</i>: <pre>peladm delete_list -list_name LIST1</pre>

Selecting a Diffusion List: peldsp select_list

Syntax	<code>peldsp select_list [-list_name]</code>
Description	Use this command to select for display the Diffusion List that corresponds to the list name criteria. To list all Diffusion Lists, do not specify any parameters. Use the optional parameters to refine the selection.
Parameters	<code>-list_name (-n)</code> : Enter the name of the Diffusion List you want to display.
Example	The following command returns the names of the Diffusion Lists that correspond to the selection criteria. <pre>peldsp select_list -n LI</pre> Each parameter is considered to be a <i>truncated</i> search key. This command returns the Diffusion Lists whose names begin with <i>LI</i>. It could, for example, return: <pre>peldsp select_list -n LI LIST1_DEFAULT LIST2 LIST34</pre>

Displaying a Diffusion List: *peldsp display_list*

Syntax	<code>peldsp display_list [-list_name]</code>
Description	Use this command to display the parameters of a Diffusion List that you identify by its name.
Parameters	<code>-list_name (-n)</code> : Enter the name of the Diffusion List for which you want to display the attributes.
Example	Gateway returns a display of each parameter associated with the specified Diffusion List object on one line in the form: <pre>name_of_field(_optional counter)="value"</pre> Each field name is prefixed with the <code>d_</code> string which points to a Diffusion List-type object. For example when you enter the following command: <pre>peldsp display_list -n LIST2</pre> Gateway might display the following fields and values (depending on your configuration): <pre>d_list_name='LIST2' d_group_name='GDEFAULT' d_model_name='MODELA' d_comment='used template site' d_template_site='TPHSE' d_site_1='DEST1' d_site_2='DEST2' d_site_3='SITE'</pre>

[Diffusion Lists: Parameters List](#)

Axway Gateway: Managing Partners

Working with Diffusion Lists (command line)

[peladm create list](#)

[peladm update list](#)

[peladm delete list](#)

[peldsp select list](#)

[peldsp display list](#)

Creating a Diffusion List: *peladm create_list*

Syntax	<code>peladm create_list {-list_name} [-group] [-comments] [-model] [-site_file] [-sites] [-add_sites] [-del_sites] [-template_site]</code>
Description	Use this command to create a Diffusion List. The <i>string</i> syntax followed by the * character is used to search for all Sites that begin with this string. Note: Processing of <code>-sites</code> and <code>-add_sites</code> parameters is always performed before processing <code>-del_site</code> (delete). You can simplify List creation by pointing to a text file that comprises all the elements to include in the Diffusion List. When you create a Diffusion List, the Sites or Lists you refer to must exist, unless you use a Template Site. In this case, the destination Sites are protocol-level destinations. No check is performed to determine whether they exist.
Parameters	<i>Mandatory</i> <code>-list_name (-n)</code> : Enter a name for the Diffusion List object you are creating. <code>-group (-gr)</code> : Enter a name for the name of the Application group associated with the Diffusion List. <code>-comments (-cts)</code> : Enter a comment about the Diffusion List. <code>-model (-ml)</code> : Enter the name of a default Model that you want to associate with the Diffusion List. <code>-site_file (-sf)</code> : Enter the name of a text file that contains a set of Sites, protocol-level destinations and/or Diffusion Lists to use as targets for the Diffusion List. <code>-sites (-s)</code> : Enter names of Sites, protocol-level destinations or other Diffusion Lists to use as targets for the Diffusion List. <code>-add_sites (-a)</code> : Enter the names of additional Sites to add to the Diffusion List. <code>-del_sites (-d)</code> : Enter the names of Sites that you want to remove from the Diffusion List. <code>-template_site (-ts)</code> : Enter the name of a Template Site to associate with the Diffusion List.

Example	The following command creates a Diffusion List object named <i>LIST1</i> with a default template (MODELA) and several target destinations. <pre>peladm create_list -list_name LIST1 -comments "First List" -model MODELA -sites "SITEA SITEB SITEC LIST11 ASITE* BSITE*"</pre> The destinations defined for this Diffusion are: • SITEA, SITEB, SITEC • LIST11 and the ASITE* and BSITE* generic Sites Note: The Sites and Diffusion List you select when creating a Diffusion List already exist.

Modifying a Diffusion List: peladm update_list

Syntax	<code>peladm update_list {-list_name} [-group] [-comments] [-model] [-site_file] [-sites] [-add_sites] [-del_sites] [-template_site]</code>
Description	Use this command to modify and update a Diffusion List. The Diffusion List name (<code>-list_name</code>) value identifies the Diffusion List to modify. Enter new values of optional the parameters you want to change.
Parameters	<i>Mandatory</i> <code>-list_name (-n)</code> : Enter a name for the Diffusion List object you want to modify <code>-group (-gr)</code> : Enter a name of the Application group associated with the Diffusion List <code>-comments (-cts)</code> : Enter a comment about the Diffusion List <code>-model (-ml)</code> : Enter the name of a default Model that you want to associate with the Diffusion List <code>-site_file (-sf)</code> : Enter the name of a text file that contains a set of Sites, protocol-level destinations and/or Diffusion Lists to use as targets for the Diffusion List <code>-sites (-s)</code> : Enter names of Sites, protocol-level destinations or other Diffusion Lists to use as targets for the Diffusion List <code>-add_sites (-a)</code> : Enter the names of additional Sites to add to the Diffusion List <code>-del_sites (-d)</code> : Enter the names of Sites that you want to remove from the Diffusion List <code>-template_site (-ts)</code> : Enter the name of a Template Site to associate with the Diffusion List

Example	The following command: • updates LIST1 with a default template (MODELB) • replaces the existing elements in the Diffusion List with SITE1 • adds SITE2 and LIST <pre>peladm update_list -n LIST1 -m1 MODELB -s "SITE1" -a "SITE2 LIST"</pre>

Deleting a Diffusion List: peladm delete_list

Syntax	<code>peladm delete_list {-list_name}</code>
Description	Use this command to delete a specified Diffusion List.
Parameters	<i>Mandatory</i> <code>-list_name (-n)</code>: Enter a name for the Diffusion List object you want to delete.
Example	The following command deletes the Diffusion List <i>LIST1</i>: <pre>peladm delete_list -list_name LIST1</pre>

Selecting a Diffusion List: peldsp select_list

Syntax	<code>peldsp select_list [-list_name]</code>
Description	Use this command to select for display the Diffusion List that corresponds to the list name criteria. To list all Diffusion Lists, do not specify any parameters. Use the optional parameters to refine the selection.
Parameters	<code>-list_name (-n)</code> : Enter the name of the Diffusion List you want to display.
Example	The following command returns the names of the Diffusion Lists that correspond to the selection criteria. <pre>peldsp select_list -n LI</pre> Each parameter is considered to be a <i>truncated</i> search key. This command returns the Diffusion Lists whose names begin with <i>LI</i>. It could, for example, return: <pre>peldsp select_list -n LI LIST1_DEFAULT LIST2 LIST34</pre>

Displaying a Diffusion List: `peldsp display_list`

Syntax	<code>peldsp display_list [-list_name]</code>
Description	Use this command to display the parameters of a Diffusion List that you identify by its name.
Parameters	<code>-list_name (-n)</code> : Enter the name of the Diffusion List for which you want to display the attributes.
Example	Gateway returns a display of each parameter associated with the specified Diffusion List object on one line in the form: <pre>name_of_field(_optional counter)="value"</pre> Each field name is prefixed with the <code>d_</code> string which points to a Diffusion List-type object. For example when you enter the following command: <pre>peldsp display_list -n LIST2</pre> Gateway might display the following fields and values (depending on your configuration): <pre>d_list_name='LIST2' d_group_name='GDEFAULT' d_model_name='MODELA' d_comment='used template site' d_template_site='TPHSE' d_site_1='DEST1' d_site_2='DEST2' d_site_3='SITE'</pre>

[Diffusion Lists: Parameters List](#)

Axway Gateway: Managing Partners

Diffusion Lists: Parameters List

The following table lists all Diffusion List command parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in *italics*.

Parameter	Meaning	Values
<code>add_sites (-a)</code>	Parameter used to add one or more Sites to the Diffusion List	Maximum: 1024 alphanumeric characters
<code>comments (-cts)</code> <i>Comments</i>	User comment	Maximum: 80 alphanumeric characters

Parameter	Meaning	Values
del_sites (-d)	Parameter used to delete one or more Sites from the Diffusion List	Maximum: 1024 alphanumeric characters
group (-gr) Group	Name of the Application group associated with the List	Maximum: 16 alphanumeric characters (default = GDEFAULT)
list_name (-n) Name	Diffusion List name	Maximum: 31 alphanumeric characters
model (-ml) Model	Default Model associated with the Diffusion List	Maximum: 26 alphanumeric characters
sites (-s) Sites	Sites, protocol-level destinations or other Diffusion Lists	Maximum: 1024 alphanumeric characters, including the asterisk (*) to define a generic Site
sites_file (-sf)	Name of the text file that contains a set of Sites, protocol-level destinations and/or Diffusion Lists	Maximum: 127 alphanumeric characters
template_site (-ts) Template Site	Template Site associated with the List	Maximum: 31 alphanumeric characters

[Working with Diffusion Lists \(command line\)](#)

Axway Gateway: Managing Partners

Trading Partner objects

Gateway *Trading Partner objects* are used for the following types of partner:

- S/MIME
- EDIINT: AS1, AS2, AS3
- RosettaNet
- SWIFTNet
- OFTP (for OFTP R2.0 security services)

To manage Trading Partner objects, refer to:

- [Working with Trading Partners](#)
- [Working with Trading Partners \(command line\)](#)

Depending on the protocol you are using, also refer to the following topics for specific details about the use of the Trading Partner object:

- [Configuring and using EDIINT](#)
- [Configuring the RosettaNet environment](#)
- [Configuring the Gateway/SWIFTNet environment](#)

[Overview: Trading Partner objects](#)

[Customizing Trading Partner parameters](#)

Axway Gateway: Managing Partners

Trading Partner objects

Gateway *Trading Partner objects* are used for the following types of partner:

- S/MIME
- EDIINT: AS1, AS2, AS3
- RosettaNet
- SWIFTNet
- OFTP (for OFTP R2.0 security services)

To manage Trading Partner objects, refer to:

- [Working with Trading Partners](#)
- [Working with Trading Partners \(command line\)](#)

Depending on the protocol you are using, also refer to the following topics for specific details about the use of the Trading Partner object:

- [Configuring and using EDIINT](#)
- [Configuring the RosettaNet environment](#)
- [Configuring the Gateway/SWIFTNet environment](#)

[Overview: Trading Partner objects](#)

[Customizing Trading Partner parameters](#)

Axway Gateway: Managing Partners

Working with Trading Partner objects

[Creating a Trading Partner object](#)

[Displaying a list of all Trading Partner objects](#)

[Displaying a list of selected Trading Partner objects](#)

[Viewing the attributes of a Trading Partner object](#)

[Modifying a Trading Partner object](#)

[Deleting a Trading Partner object](#)

Creating a Trading Partner object

To create a new Trading Partner object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.
Gateway displays a context menu.
3. Select **New**
Gateway displays a context menu.
4. Select the type of partner:
 - **SMIME partner**
 - **AS1 partner**
 - **AS2 partner**
 - **AS3 partner**
 - **RosettaNet partner**
 - **SWIFTNet partner**
 - **OFTP partner** (for OFTP R2.0 security services)
Gateway displays the *Trading Partner* window. The parameters displayed depend on the *type of partner* (protocol).
If you are using:
 - SWIFTNet, go to [Configuring the Gateway/SWIFTNet environment](#)
 - S/MIME, AS1, AS2, AS3, RosettaNet or OFTP, continue with the next step
5. Use the information in the following table to complete the fields of the *Trading Partner* window.

Trading Partner – S/MIME, AS1, AS2, AS3, RosettaNet or OFTP

Field	Definition	Partner type...
Identification criteria		

Field	Definition	Partner type...
Alias	Enter a name for the Trading Partner object to use for local identification operations. The name must be unique. Gateway stores this alias as part of the data of each Transfer Request. You can use the alias as a substitute for the partner name parameter when depositing Transfer Requests.	Local and Remote
Format	This field displays the protocol format that you selected when creating the new Trading Partner: • SMIME • AS1 • AS2 • AS3 • RosettaNet • SWIFTNet • CMS (Cryptographic Message Syntax) for OFTP R2.0 security services	
Local Partner	Click if required: • selected = local Trading Partner object • not selected = remote Trading Partner object (default)	
Name/Email	For S/MIME or AS1, enter an email address. For other protocols, enter a unique name for the Trading Partner object. Maximum: 128 characters. When searching the partner in Gateway, the first part of the email (before the @) is case sensitive, while the domain (after the @) is case insensitive .	

Field	Definition	Partner type...
Comments	<i>Optional</i> Enter a free-text description of the Trading Partner object. Maximum: 80 alphanumeric characters. This description is included in the displayed list of available Trading Partners.	
Certificate/Key <i>The fields displayed depend on the type of partner and the selected options.</i>		
Use dual certificates	Click this button to toggle between the use of single or dual certificates. The effect of this button varies according to the context: • If you are creating a local partner and you... ◦ select Use dual certificates: A signing certificate is used to sign what is sent, and an encryption certificate is used to decrypt what is received. ◦ do not select Use dual certificates: A single certificate is used to sign what is sent, and is used to decrypt what is received. • If you are creating a remote partner and you... ◦ select Use dual certificates: An encryption certificate is used to encrypt what is sent, and a signing certificate is used to check the signature of what is received. ◦ do not select Use dual certificates: A single certificate is used to encrypt what is sent, and to check the signature of what is received.	Local and Remote
The following three lines are visible when you do not select the Use dual certificates button.		

Field	Definition	Partner type...
Certificate: Type	Select a single certificate type to handle both outgoing and incoming messages: • None • Secs alias: local Gateway PKI • PassPort PS entity: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • PassPort PS certificate: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Local and Remote
Certificate: Name	Enter the name of the certificate to be used to handle both outgoing and incoming messages. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.	
Certificate: Set key password	Opens a dialog box to enter the password of the certificate key, to be used to handle both outgoing and incoming messages. This parameter is not available if you select SECS ALIAS as the <i>Certificate Type</i>. When the certificate type is PassPort PS Entity, you need to enter the entity's password in this field. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.	
The following six lines are visible when you do select the Use dual certificates button.		

Field	Definition	Partner type...
Signing certificate: Type	<i>This field is visible when you select the Use dual certificates button.</i> Select a certificate type for message signing: • None • Secs alias: local Gateway PKI • PassPort PS entity: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • PassPort PS certificate: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Remote
Signing certificate: Name	<i>This field is visible when you select the Use dual certificates button.</i> Enter the name of the certificate to use for signing. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.	
Signing certificate: Set Entity password	<i>This field is visible when you select the Use dual certificates button.</i> Enter the password for signing certificate key use. This parameter is not available if you select SECS ALIAS as the <i>Signing Certificate Type</i>.	

Field	Definition	Partner type...
Encryption certificate: Type	<i>This field is visible when you select the Use dual certificates button.</i> Select a certificate type for message encryption or decryption: • None • Secs alias: local Gateway PKI • PassPort PS entity: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • PassPort PS certificate: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Remote
Encryption certificate: Name	<i>This field is visible when you select the Use dual certificates button.</i> Enter the name of the certificate to use for the encryption of sent messages. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.	
Encryption certificate: Key password	<i>This field is visible when you select the Use dual certificates button.</i> Enter the password for the use of the certificate to be used for the encryption of sent messages. This parameter is not available if you select SECS ALIAS as the <i>Encryption Certificate Type</i>.	

Field	Definition	Partner type...
Encryption algo	<i>This field is visible when you select any option other than none from the Encryption certificate: Type field.</i> Select the algorithm to use for message and/or acknowledgment encryption: • AES128 (default) • AES256 • DES • 3DES • RC2_40 (for AS2 and RosettaNet only) • RC4 (for AS2 and RosettaNet only) Note: Gateway supports RC4 with 128-bit key size. Gateway does not support RC4 with 40-bit or 64-bit key sizes. Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information.	
Key encryption algo	<i>This field is visible when you select any option other than none from the Encryption certificate: Type field.</i> The only available value for this field is: • RSA (default)	

Field	Definition	Partner type...
Signing algo	For creating or modifying Local Partners only. This field is visible when you select any option other than none from the Certificate: Type field. Select the algorithm to use for signing: • SHA1 with RSA (default) • MD5 with RSA • SHA512 with RSA • SHA256 with RSA Note Restrictions apply to this list in FIPS mode. See the Security Guide for further information. Note The MD5 with RSA signing algorithm is not available unless the configuration option <code>[monitor]enable_md5_hash_algo</code> is set to yes (not recommended).	Local
Use Multiple Certificates	Only if you are using PassPort PS Select this option to use multiple certificates. Refer to the Axway PassPort documentation for details.	Local and Remote
Default parameters for file sending These fields are active only when you are creating or modifying a remote Trading Partner object (Local Partner checkbox unselected)		

Field	Definition	Partner type...
Associated originator Partner	Either: • Enter the originator partner to use when sending a file, or • Select a partner from the drop-down list	Remote
Default outgoing Site	Either: • Enter a default outgoing Site name, or • Select a Site name from the drop-down list The Site must be an existing Site on your local Gateway. When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.	
Remote directory (url complement)	<i>Not available for all types of partner</i> Specify: • For AS2, the URL path of the directory where the remote exchange partner stores received data • For AS3, the directory where the remote partner stores received data Your remote destination exchange partner must provide you with this information. This parameter is only significant when the remote exchange partner is operating as a server.	
Requested services: Encrypt file	Select this option to activate encryption.	
Requested services: Sign file	Select this option to activate message signing.	
Requested services: Compress file	Select this option to activate message compression. If you select this option, select a compression algorithm in the Compression algo field.	

Field	Definition	Partner type...
Requested services: Compression algo	If you select the Compress file option, select a compression algorithm from the drop-down list.	
Requested services: Request receipt	Specify whether or not you want to receive a receipt on reception of a file you send. If you choose to receive a receipt, specify whether the receipt is signed or unsigned. Enter one of the following values: • None = no receipt requested (default) • Signed = request signed receipt • Unsigned = request unsigned receipt If you select either the Signed or Unsigned values for this option, enter a receipt reception address in the Receipt url/email field.	
Requested services: Synchronize receipt	<i>Only for AS2.</i> Specify whether or not you want to synchronize receipt delivery: • selected: your remote partner returns a reception acknowledgment immediately on file reception and unpacking • not selected: your remote partner returns a reception acknowledgment in a protocol session independent of the original transfer session	
Requested services: Receipt url/email	<i>Not available for the S/MIME protocol.</i> If you choose to receive receipts of sent messages, specify the address to which the receiving partner sends the receipt.	
Parameters for file reception		
These fields are active only when you are creating or modifying a remote Trading Partner object (Local Partner checkbox unselected)		

Field	Definition	Partner type...
Force signed	Select this option to specify that you want Gateway to require message signing on reception. If you request signing by selecting the Sign file option above, you should also activate this function.	Remote
Force encrypted	Select this option to specify that you want Gateway to require message encryption as criteria for reception. If you request encryption by selecting the Encrypt file option above, you should also activate this function.	
Force receipt requested	Specify whether or not you want Gateway to require receipt requests on reception. Select one of the following values: • Undefined = Do not require a receipt request (default) • None = Ensure that no receipt is requested. Gateway rejects messages with receipt requests • Signed = Ensure that a signed receipt is requested. Gateway only accepts messages with requests for signed receipts • Unsigned = Ensure that a unsigned receipt is requested. Gateway only accepts messages with requests for unsigned receipts	
Force receipt url/email	If you enter an address as a value for this parameter, Gateway checks to confirm that the receipt return path specified in the incoming message is the same. If the check fails, Gateway rejects the message.	
Advanced parameters...: Local file location for encoding/decoding Click the Advanced... button to display the following two fields. These fields enable you to specify the directory and name format of the file where transfers are processed for encoding (outgoing files) and decoding (incoming files). You may want to specify a secure directory for this file, keeping in mind that the files that are decoded in this directory can be read by malicious intruders. If you do not specify a directory, Gateway creates the files in <Gateway installation directory>\runtime\tmp		

Field	Definition	Partner type...
Directory path	Enter the local physical path of the directory where transferred files are encoded and decoded.	Local and Remote
File name pattern	Enter a specific file name format to enable you to identify the transferred file in the specified out directory.	

Note: If you are using RosettaNet, also refer to [Configuring the RosettaNet environment](#).

Displaying a list of Trading Partner objects

To display a list of all Trading Partner objects stored in the database:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Trading Partners** sub-node.

Gateway displays a list of all existing Trading Partner objects in the right (display) pane.

Displaying a list of Selected Trading Partner objects

To generate a list of selected Trading Partner objects:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.

Gateway displays a context menu.

3. Choose **Select...** from the context menu.

Gateway displays the *Select Trading Partner* window.

4. In the *Select Trading Partner* window, use the data fields to enter criteria for Trading Partner display selection. You can use one or more of the following selection criteria:

- Name
- Format
- Local or Remote partner

5. Click **OK** to accept the display selection criteria.

Gateway regenerates the Trading Partner List display in the right window, taking into account your selection criteria.

Viewing the attributes of a Trading Partner

To view the attributes of an existing Trading Partner via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Trading Partners** sub-node.

Gateway displays all existing Trading Partners in the right (display) pane.
3. In the right pane, right-click to select the Trading Partner that you want to modify.

Gateway displays a context menu.
4. Select **View...** from the context menu.

Gateway opens the *Trading Partner (Read only)* window. This window provides you with a view of all attributes for the selected Trading Partner.
5. When you have finished viewing the information, click **Close** to quit the window.

Modifying a Trading Partner object

To modify and update a Trading Partner object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.

Gateway displays all existing Trading Partners in the right (display) pane.
3. Click the name of the Trading Partner to update. Right-click and select **Modify...** from the context menu.

Gateway displays the *Trading Partner* window. For window details refer to [Creating a Trading Partner](#).
4. Make your changes and click **OK** to confirm.

Note: There are two types of Trading Partners, Local and Remote which are two distinct objects and must be treated as such. If the Remote Partner is active when modifying the Trading Partner and switching to Local Partner occurs, this may result in abnormal behavior of the product.

Deleting a Trading Partner object

To delete a Trading Partner object:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click the **Trading Partners** sub-node.

Gateway displays all existing Trading Partners in the right (display) pane.
3. Click the name of the Trading Partner that you want to delete. Right-click and select **Delete...** from the context menu.

Gateway displays a confirmation dialog box.
4. In the confirmation dialog box, click **Yes** to confirm the delete operation.

Alternatively, click **No** or **Cancel** to abandon the operation.

Gateway deletes the Trading Partner object from the database and removes the entry from the display window.

[Working with Trading Partners \(command line\)](#)

[Overview: Trading Partners](#)

Working with Trading Partner objects (command line)

[peladm create tradepart](#)

[peldsp display tradepart](#)

[peldsp select tradepart](#)

[peladm update tradepart](#)

[peladm delete tradepart](#)

Note: If you are using SWIFTNet, refer to [Configuring the Gateway/SWIFTNet environment](#) for specific parameters.

Creating a Trading Partner: *peladm create_tradepart*

Syntax	<code>peladm create_tradepart</code> [see parameters below]
Description	Use this command to create a new Trading Partner object. You can specify one of two types of Trading Partner: • Local • Remote

Parameters	Parameter name (and abbreviation)	Description	For local or remote
	<code>-name (-n)</code> (<i>mandatory</i>)	For S/MIME or AS1, enter an email address. For other protocols, enter a unique name for the Trading Partner object. The remote business exchange partner uses this name to identify the local partner. Maximum: 128 characters.	Local and remote
	<code>-format (-f)</code> (<i>mandatory</i>)	Enter one of the following protocol formats for exchanges with the remote partner: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET	Local and remote

	<code>-alias (-a)</code> (<i>mandatory</i>)	Enter a name for the Trading Partner object to use for local identification operations. The name must be unique. Gateway stores this alias as part of the data of each Transfer Request. You can use the alias as a substitute for the <code>partner name</code> parameter when depositing Transfer Requests. By default, the alias is the <code>name (-n)</code> value set to uppercase and truncated to 31 characters.	Local and remote
	<code>-certificate_name (-ctfn)</code>	Enter the SECS certificate name to use for the partner.	Remote
	<code>-comments (-cts)</code>	Optionally, enter a free-text description of the Trading Partner object. Maximum: 80 alphanumeric characters.	Local and remote
	<code>-compress (-c)</code>	Specify whether or not you want to apply compression to the data contained in sent messages: • Y = compress sent messages • N = (default)	Remote

	-compress_algo (-ca)	Enter the algorithm to use for message and/or receipt compression. You can only enter the following value: • ZLIB	Remote
	-dir_name (-dn)	Enter: • For AS2, the URL path of the directory where the remote exchange partner stores received data • For AS3, the directory where the remote partner stores received data Your remote destination exchange partner must provide you with this information. This parameter is only significant when the remote exchange partner is operating as a server.	Remote
	-dir_path (-dp)	Enter the local physical path of the file where transferred files are encoded and decoded.	Remote
	-encrypt (-e)	Specify whether or not you want to encrypt the data contained in sent messages: • Y = Encrypt sent messages • N = Do not encrypt (default)	Remote

	-encryption_ algo (-ea)	Enter the algorithm to use for message and/or receipt encryption. One of: • AES128 (default) • AES256 • DES • 3DES • RC2_40 (<i>for AS2 and RosettaNet only</i>) • RC4 (<i>for AS2 and RosettaNet only</i>) Note: Gateway supports RC4 with 128-bit key size. Gateway does not support RC4 with 40-bit or 64-bit key sizes.	Remote
	-encryption_ obj_name (-eon)	Enter the name of the encryption certificate.	Local and remote
	-encryption_ obj_password (-eop)	Enter the password to use with the encryption certificate. Do not use this parameter if you enter SECS_ALIAS for encryption_obj_type.	Local and remote

	<code>-encryption_obj_type (-eot)</code>	Enter the type of encryption certificate to use for encrypting sent messages. One of: • NONE • SECS_ALIAS: local Gateway PKI • XPP_ENTITY: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • XPP_CERTIFICATE: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Local and remote
	<code>-file_component (-fc)</code>	Enter the name of the file where the exchange partner stores received data. Your remote destination exchange partner must provide you with this information. This parameter is only significant when the remote exchange partner is operating as a server.	Remote

	<code>-force_ encrypted (-fe)</code>	Specify whether or not you want Gateway to require message encryption as a criteria for reception: • Y = Ensure received message is encrypted. Gateway rejects unencrypted messages. • N = (default)	Remote

	<pre>-force_receipt_request (-frr)</pre>	Specify whether or not you want Gateway to require receipt requests on reception. Enter one of: • Undef = Do not require a receipt request (default) • None = Ensure that no receipt is requested. Gateway rejects messages with receipt requests • Signed = Ensure that a signed receipt is requested. Gateway only accepts messages with requests for signed receipts. • Unsigned = Ensure that a unsigned receipt is requested. Gateway only accepts messages with requests for unsigned receipts.	Remote
	<pre>-force_receipt_request_to (-frrto)</pre>	If you enter an address as a value for this parameter, Gateway checks to confirm that the receipt return path specified in the incoming message is the same. If the check fails, Gateway rejects the message.	Remote

	<code>-force_signed</code> <code>(-fs)</code>	Specify whether or not you want Gateway to require for message signing on reception: • Y = Check if received messages are signed. Gateway rejects unsigned messages • N = No signing required. Gateway accepts all incoming messages (default)	Remote
	<code>-is_local (-loc)</code>	Specify if the Trading Partner object you are creating is for a local or a remote partner: • Y = local Trading Partner object • N = remote Trading Partner object (default)	Local
	<code>-key_encryption_algo</code> <code>(-kea)</code>	Enter the algorithm to use for key encryption. You can enter only the following value: • RSA (default)	Remote

	<code>-receipt_ request (-rr)</code>	Specify whether or not you want to receive a receipt for the reception of a file you send. If you choose to receive a receipt, specify whether the receipt is signed or unsigned. Enter one of: • None = no receipt requested (default) • Signed = request signed receipt • Unsigned = request unsigned receipt	Remote
	<code>-receipt_ request_ synchronized (- rrs)</code>	Specify whether or not you want to synchronize receipt delivery so that the receipt is generated and delivered in the same protocol session as the original transfer: • Y = Yes, synchronize • N = No, do not synchronize (default)	Remote

	<code>-receipt_ request_to (- rrto)</code>	If you choose to receive receipts of sent messages, specify the address to which the receiving partner sends the receipt. For: • AS1 - The default return address is the email address you entered as the <code>name (-n)</code> value. You can specify an alternative email address in this parameter • AS2 asynchronous mode - Enter a return URL or email address • AS3 - Enter a return URL or email address • SWIFTNet (<i>FileAct, Real time only</i>) - Refer to Configuring the Gateway/SWIFT Net environment For other protocols, Gateway does not use this parameter.	Remote

	<code>-remote_site (-rs)</code>	Default outgoing Site name. The Site must be an existing Site on your local Gateway. When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.	Remote
	<code>-sign (-s)</code>	Specify whether or not you want signing of the data contained in sent messages: • Y = sign sent messages • N = (default)	Remote
	<code>-sign_obj_name (-son)</code>	Enter the name of the signing certificate.	Remote
	<code>-sign_obj_password (-sop)</code>	Enter the password to use with the signing certificate. Do not use this parameter if you enter SECS_ALIAS for <code>sign_obj_type</code> .	Remote

	<code>-sign_obj_type</code> <code>(-sot)</code>	Enter the type of signing certificate to use on sent messages. One of: • NONE • SECS_ALIAS: local Gateway PKI • XPP_ENTITY: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • XPP_CERTIFICATE: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Remote
	<code>-signing_algo</code> <code>(-sa)</code>	Enter the algorithm to use for message and/or receipt signing. One of: • SHA1withRSA (default) • MD5withRSA • SHA256withRSA* • SHA512withRSA* * Available only for AS1, AS2 and AS3.	Local
	<code>-tradepart_org_name</code> <code>(-org)</code>	When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.	Remote

**Example 1:
creating a
remote
partner
for AS2**

The following command creates a remote Trading Partner object with the characteristics:

- name = PartnerRemote1
- format = AS2
- alias = AS2PR1
- default remote target = CYCL_XIP1REC_AS2
- signing = yes
- encryption = yes
- receipt = yes
- receipt URL =

`http://hpx15:8382/user1/transfer?user=user1&password=user1`

- signed receipt forced
- compression = yes
- compression algo = ZLIB
- signing algo = MD5withRSA
- encryption algo = 3DES
- encryption certificate type = XPP_CERTIFICATE
- encryption certificate name = XPPUSER1
- encryption certificate password = xppuser1
- signing object type = XPP_CERTIFICATE
- signing object name = XPPUSER1
- signing object password = xppuser1

```
peladm create_tradepart
  -n PartnerRemote1
  -f AS2
  -a AS2PR1
  -rs CYCL_XIP1REC_AS2
  -s Y
  -e Y
  -rr signed
  -rrto
  'http://hpx15:8382/user1/transfer?user=user1&password=user1'
  -frr signed
  -c Y
  -ca ZLIB
  -sa MD5withRSA
  -ea 3DES
```

	<pre> -eot XPP_CERTIFICATE -eon XPPUSER1 -eop xppuser1 -sot XPP_CERTIFICATE -son XPPUSER1 -sop xppuser1 </pre>
Example 2: creating a local partner for AS2	The following command creates a local Trading Partner object with the characteristics: • name = PartnerLocal1 • format = AS2 • alias = AS2PL1 • local = yes • signing algo = SHA1withRSA • encryption certificate type = XPP_CERTIFICATE • encryption certificate name = GW • signing object type = XPP_CERTIFICATE • signing object name = GW <pre> peladm create_tradepart -n PartnerLocal1 -a AS2PL1 -f AS2 -loc Y -sa SHA1withRSA -eot XPP_CERTIFICATE -eon GW -sot XPP_CERTIFICATE -son GW </pre>

Displaying a Trading Partner: peldsp display_tradepart

Syntax	<code>peldsp display_tradepart {-name} [-format] [-alias]</code>
Description	Use this command to display the characteristics of a selected Trading Partner object.

Parameters	Parameter name (and abbreviation)	Description
	<code>-name (-n)</code> (<i>mandatory</i>)	Enter the name of the Trading Partner object for which you want to display the characteristics.
	<code>-format (-f)</code>	Enter the format of the Trading Partner object for which you want to display the characteristics. Enter one of: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET
	<code>-alias (-a)</code>	Enter the alias of the Trading Partner object for which you want to display the characteristics.

Example

Enter the following command to display the characteristics of a selected AS2-type Trading Partner object:

```
peldsp display_tradepart
```

```
-f AS2
```

```
-n user1_as1@domain.com
```

```
trade_name='user1_as1@domain.com'
```

```
trade_alias='AS2_2'
```

```
trade_format='AS2'
```

```
trade_is_local='N'
```

```
trade_comments=''
```

```
trade_dir_path=''
```

```
trade_file_component=''
```

```
trade_sign_obj_type='XPP_ENTITY'
```

```
trade_sign_obj_name='USER2'
```

```
trade_sign_obj_password=''
```

```
trade_encryption_obj_type='XPP_ENTITY'
```

```
trade_encryption_obj_name='USER2'
```

```
trade_encryption_obj_password=''
```

```
trade_tradepart_org_name='part 1'
```

```
trade_remote_site='GW1'
```

```
trade_dir_name='/'
```

```
trade_encrypt='Y'
```

```
trade_sign='Y'
```

```
trade_compress='Y'
```

```
trade_compress_algo='ZLIB'
```

```
trade_encryption_algo='AES128'
```

```
trade_key_encryption_algo='RSA'
```

```
trade_signing_algo='SHA1withRSA'
```

```
trade_receipt_request='signed'
```

```
trade_receipt_request_to=''
```

```
trade_receipt_request_synchronized='N'
```

```
trade_force_receipt_request='undefined'
```

```
trade_force_signed='N'
```

```
trade_force_encrypted='N'
```

```
trade_force_receipt_request_to=''
```

Selecting a Trading Partner: *peldsp select_tradepart*

Syntax	peldsp select_tradepart {-name} [-alias] [-format] [-is_local]
Description	Use this command to select a given Trading Partner.

Parameters	Parameter name (and abbreviation)	Description
	<code>-format (-f)</code>	Protocol format. Enter one of: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET
	<code>-name (-n)</code> (mandatory)	Enter the unique Trading Partner name.
	<code>-alias (-a)</code>	Enter a name for the Trading Partner object to use as selection criteria. Gateway stores this alias as part of the data of each Transfer Request. You can use the alias as a substitute for the <code>partner name</code> parameter when depositing Transfer Requests. By default, the alias is the <code>name (-n)</code> value set to uppercase and truncated to 31 characters.
	<code>-is_local (-loc)</code>	Enter one of: • Y = local • N = remote (default)
Example	The following example selects the AS1 Trading Partner object named <code>user1_as1@domain.com</code> : <pre> peldsp select_tradepart -n user1_as1@domain.com -f AS1 </pre>	

Modifying a Trading Partner: *peladm update_tradepart*

Syntax	<code>peladm update_tradepart [parameters]</code>
Description	Use this command to modify the characteristics of a Trading Partner.
Parameters	Refer to the parameters for the peladm create_tradepart command.

Example	Enter the following command to modify the Trading Partner object named <i>MyTrade</i>: <pre>peladm update_tradepart -n MyTrade -f AS2 -rs CYCL_XIP1REC_AS2 -s N -e N -c N</pre>

Note: There are two types of Trading Partners, Local and Remote which are two distinct objects and must be treated as such. If the Remote Partner is active when modifying the Trading Partner and switching to Local Partner occurs, this may result in abnormal behavior of the product.

Deleting a Trading Partner: *peladm delete_tradepart*

Syntax	<code>peladm delete_tradepart {-name} [-format] [-alias]</code>	
Description	Use this command to delete a Trading Partner.	
Parameters	Parameter name (and abbreviation)	Description
	<code>-name (-n)</code> (mandatory)	Enter the unique Trading Partner name.
	<code>-format (-f)</code>	Protocol format. Enter one of: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET
	<code>-alias (-a)</code>	Enter an alias for the Trading Partner object you want to delete.
Example	The following example deletes the AS1 Trading Partner object named <i>user1_as1@domain.com</i>: <pre>peladm delete_tradepart -n user1_as1@domain.com -f AS1</pre>	

[Overview: Trading Partners](#)

[Working with Trading Partners \(GUI\)](#)

[Configuring the Gateway/SWIFTNet environment](#)

Axway Gateway: Managing Partners

Working with Trading Partner objects (command line)

[`peladm create tradepart`](#)[`peldsp display tradepart`](#)[`peldsp select tradepart`](#)[`peladm update tradepart`](#)[`peladm delete tradepart`](#)

Note: If you are using SWIFTNet, refer to [Configuring the Gateway/SWIFTNet environment](#) for specific parameters.

Creating a Trading Partner: `peladm create_tradepart`

Syntax	<code>peladm create_tradepart</code> [see parameters below]
Description	Use this command to create a new Trading Partner object. You can specify one of two types of Trading Partner: • Local • Remote

Parameters	Parameter name (and abbreviation)	Description	For local or remote
	<code>-name (-n)</code> (<i>mandatory</i>)	For S/MIME or AS1, enter an email address. For other protocols, enter a unique name for the Trading Partner object. The remote business exchange partner uses this name to identify the local partner. Maximum: 128 characters.	Local and remote
	<code>-format (-f)</code> (<i>mandatory</i>)	Enter one of the following protocol formats for exchanges with the remote partner: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET	Local and remote

	<code>-alias (-a)</code> <i>(mandatory)</i>	Enter a name for the Trading Partner object to use for local identification operations. The name must be unique. Gateway stores this alias as part of the data of each Transfer Request. You can use the alias as a substitute for the <code>partner name</code> parameter when depositing Transfer Requests. By default, the alias is the <code>name (-n)</code> value set to uppercase and truncated to 31 characters.	Local and remote
	<code>-certificate_name (-ctfn)</code>	Enter the SECS certificate name to use for the partner.	Remote
	<code>-comments (-cts)</code>	Optionally, enter a free-text description of the Trading Partner object. Maximum: 80 alphanumeric characters.	Local and remote
	<code>-compress (-c)</code>	Specify whether or not you want to apply compression to the data contained in sent messages: • Y = compress sent messages • N = (default)	Remote

	<code>-compress_algo (-ca)</code>	Enter the algorithm to use for message and/or receipt compression. You can only enter the following value: • ZLIB	Remote
	<code>-dir_name (-dn)</code>	Enter: • For AS2, the URL path of the directory where the remote exchange partner stores received data • For AS3, the directory where the remote partner stores received data Your remote destination exchange partner must provide you with this information. This parameter is only significant when the remote exchange partner is operating as a server.	Remote
	<code>-dir_path (-dp)</code>	Enter the local physical path of the file where transferred files are encoded and decoded.	Remote
	<code>-encrypt (-e)</code>	Specify whether or not you want to encrypt the data contained in sent messages: • Y = Encrypt sent messages • N = Do not encrypt (default)	Remote

	-encryption_ algo (-ea)	Enter the algorithm to use for message and/or receipt encryption. One of: • AES128 (default) • AES256 • DES • 3DES • RC2_40 (<i>for AS2 and RosettaNet only</i>) • RC4 (<i>for AS2 and RosettaNet only</i>) Note: Gateway supports RC4 with 128-bit key size. Gateway does not support RC4 with 40-bit or 64-bit key sizes.	Remote
	-encryption_ obj_name (-eon)	Enter the name of the encryption certificate.	Local and remote
	-encryption_ obj_password (-eop)	Enter the password to use with the encryption certificate. Do not use this parameter if you enter SECS_ALIAS for encryption_obj_type.	Local and remote

	<code>-encryption_obj_type (-eot)</code>	Enter the type of encryption certificate to use for encrypting sent messages. One of: • NONE • SECS_ALIAS: local Gateway PKI • XPP_ENTITY: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • XPP_CERTIFICATE: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Local and remote
	<code>-file_component (-fc)</code>	Enter the name of the file where the exchange partner stores received data. Your remote destination exchange partner must provide you with this information. This parameter is only significant when the remote exchange partner is operating as a server.	Remote

	<code>-force_ encrypted (-fe)</code>	Specify whether or not you want Gateway to require message encryption as a criteria for reception: • Y = Ensure received message is encrypted. Gateway rejects unencrypted messages. • N = (default)	Remote

	<pre>-force_receipt_request (-frr)</pre>	Specify whether or not you want Gateway to require receipt requests on reception. Enter one of: • Undef = Do not require a receipt request (default) • None = Ensure that no receipt is requested. Gateway rejects messages with receipt requests • Signed = Ensure that a signed receipt is requested. Gateway only accepts messages with requests for signed receipts. • Unsigned = Ensure that a unsigned receipt is requested. Gateway only accepts messages with requests for unsigned receipts.	Remote
	<pre>-force_receipt_request_to (-frrto)</pre>	If you enter an address as a value for this parameter, Gateway checks to confirm that the receipt return path specified in the incoming message is the same. If the check fails, Gateway rejects the message.	Remote

	<code>-force_signed (-fs)</code>	Specify whether or not you want Gateway to require for message signing on reception: • Y = Check if received messages are signed. Gateway rejects unsigned messages • N = No signing required. Gateway accepts all incoming messages (default)	Remote
	<code>-is_local (-loc)</code>	Specify if the Trading Partner object you are creating is for a local or a remote partner: • Y = local Trading Partner object • N = remote Trading Partner object (default)	Local
	<code>-key_encryption_algo (-kea)</code>	Enter the algorithm to use for key encryption. You can enter only the following value: • RSA (default)	Remote

	<code>-receipt_ request (-rr)</code>	Specify whether or not you want to receive a receipt for the reception of a file you send. If you choose to receive a receipt, specify whether the receipt is signed or unsigned. Enter one of: • None = no receipt requested (default) • Signed = request signed receipt • Unsigned = request unsigned receipt	Remote
	<code>-receipt_ request_ synchronized (- rrs)</code>	Specify whether or not you want to synchronize receipt delivery so that the receipt is generated and delivered in the same protocol session as the original transfer: • Y = Yes, synchronize • N = No, do not synchronize (default)	Remote

	<code>-receipt_ request_to (- rrto)</code>	If you choose to receive receipts of sent messages, specify the address to which the receiving partner sends the receipt. For: • AS1 - The default return address is the email address you entered as the <code>name (-n)</code> value. You can specify an alternative email address in this parameter • AS2 asynchronous mode - Enter a return URL or email address • AS3 - Enter a return URL or email address • SWIFTNet (<i>FileAct, Real time only</i>) - Refer to Configuring the Gateway/SWIFT Net environment For other protocols, Gateway does not use this parameter.	Remote

	<code>-remote_site (-rs)</code>	Default outgoing Site name. The Site must be an existing Site on your local Gateway. When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.	Remote
	<code>-sign (-s)</code>	Specify whether or not you want signing of the data contained in sent messages: • Y = sign sent messages • N = (default)	Remote
	<code>-sign_obj_name (-son)</code>	Enter the name of the signing certificate.	Remote
	<code>-sign_obj_password (-sop)</code>	Enter the password to use with the signing certificate. Do not use this parameter if you enter SECS_ALIAS for <code>sign_obj_type</code> .	Remote

	<code>-sign_obj_type</code> <code>(-sot)</code>	Enter the type of signing certificate to use on sent messages. One of: • NONE • SECS_ALIAS: local Gateway PKI • XPP_ENTITY: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details. • XPP_CERTIFICATE: for use with the PassPort PS server. Refer to the Axway PassPort documentation for details.	Remote
	<code>-signing_algo</code> <code>(-sa)</code>	Enter the algorithm to use for message and/or receipt signing. One of: • SHA1withRSA (default) • MD5withRSA • SHA256withRSA* • SHA512withRSA* * Available only for AS1, AS2 and AS3.	Local
	<code>-tradepart_org_name</code> <code>(-org)</code>	When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.	Remote

**Example 1:
creating a remote partner for AS2**

The following command creates a remote Trading Partner object with the characteristics:

- name = PartnerRemote1
- format = AS2
- alias = AS2PR1
- default remote target = CYCL_XIP1REC_AS2
- signing = yes
- encryption = yes
- receipt = yes
- receipt URL =

`http://hpx15:8382/user1/transfer?user=user1&password=user1`

- signed receipt forced
- compression = yes
- compression algo = ZLIB
- signing algo = MD5withRSA
- encryption algo = 3DES
- encryption certificate type = XPP_CERTIFICATE
- encryption certificate name = XPPUSER1
- encryption certificate password = xppuser1
- signing object type = XPP_CERTIFICATE
- signing object name = XPPUSER1
- signing object password = xppuser1

`peladm create_tradepart`

```
-n PartnerRemote1
-f AS2
-a AS2PR1
-rs CYCL_XIP1REC_AS2
-s Y
-e Y
-rr signed
-rrto
```

`'http://hpx15:8382/user1/transfer?user=user1&password=user1'`

```
-frr signed
-c Y
-ca ZLIB
-sa MD5withRSA
-ea 3DES
```

	<pre> -eot XPP_CERTIFICATE -eon XPPUSER1 -eop xppuser1 -sot XPP_CERTIFICATE -son XPPUSER1 -sop xppuser1 </pre>
Example 2: creating a local partner for AS2	The following command creates a local Trading Partner object with the characteristics: • name = PartnerLocal1 • format = AS2 • alias = AS2PL1 • local = yes • signing algo = SHA1withRSA • encryption certificate type = XPP_CERTIFICATE • encryption certificate name = GW • signing object type = XPP_CERTIFICATE • signing object name = GW <pre> peladm create_tradepart -n PartnerLocal1 -a AS2PL1 -f AS2 -loc Y -sa SHA1withRSA -eot XPP_CERTIFICATE -eon GW -sot XPP_CERTIFICATE -son GW </pre>

Displaying a Trading Partner: peldsp display_tradepart

Syntax	<code>peldsp display_tradepart {-name} [-format] [-alias]</code>
Description	Use this command to display the characteristics of a selected Trading Partner object.

Parameters	Parameter name (and abbreviation)	Description
	<code>-name (-n)</code> (<i>mandatory</i>)	Enter the name of the Trading Partner object for which you want to display the characteristics.
	<code>-format (-f)</code>	Enter the format of the Trading Partner object for which you want to display the characteristics. Enter one of: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET
	<code>-alias (-a)</code>	Enter the alias of the Trading Partner object for which you want to display the characteristics.

Example

Enter the following command to display the characteristics of a selected AS2-type Trading Partner object:

```
peldsp display_tradepart
```

```
-f AS2
```

```
-n user1_as1@domain.com
```

```
trade_name='user1_as1@domain.com'
```

```
trade_alias='AS2_2'
```

```
trade_format='AS2'
```

```
trade_is_local='N'
```

```
trade_comments=''
```

```
trade_dir_path=''
```

```
trade_file_component=''
```

```
trade_sign_obj_type='XPP_ENTITY'
```

```
trade_sign_obj_name='USER2'
```

```
trade_sign_obj_password=''
```

```
trade_encryption_obj_type='XPP_ENTITY'
```

```
trade_encryption_obj_name='USER2'
```

```
trade_encryption_obj_password=''
```

```
trade_tradepart_org_name='part 1'
```

```
trade_remote_site='GW1'
```

```
trade_dir_name='/'
```

```
trade_encrypt='Y'
```

```
trade_sign='Y'
```

```
trade_compress='Y'
```

```
trade_compress_algo='ZLIB'
```

```
trade_encryption_algo='AES128'
```

```
trade_key_encryption_algo='RSA'
```

```
trade_signing_algo='SHA1withRSA'
```

```
trade_receipt_request='signed'
```

```
trade_receipt_request_to=''
```

```
trade_receipt_request_synchronized='N'
```

```
trade_force_receipt_request='undefined'
```

```
trade_force_signed='N'
```

```
trade_force_encrypted='N'
```

```
trade_force_receipt_request_to=''
```

Selecting a Trading Partner: *peldsp select_tradepart*

Syntax	peldsp select_tradepart {-name} [-alias] [-format] [-is_local]
Description	Use this command to select a given Trading Partner.

Parameters	Parameter name (and abbreviation)	Description
	<code>-format (-f)</code>	Protocol format. Enter one of: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET
	<code>-name (-n)</code> (mandatory)	Enter the unique Trading Partner name.
	<code>-alias (-a)</code>	Enter a name for the Trading Partner object to use as selection criteria. Gateway stores this alias as part of the data of each Transfer Request. You can use the alias as a substitute for the <code>partner name</code> parameter when depositing Transfer Requests. By default, the alias is the <code>name (-n)</code> value set to uppercase and truncated to 31 characters.
	<code>-is_local (-loc)</code>	Enter one of: • Y = local • N = remote (default)
Example	The following example selects the AS1 Trading Partner object named <code>user1_as1@domain.com</code> : <pre> peldsp select_tradepart -n user1_as1@domain.com -f AS1 </pre>	

Modifying a Trading Partner: `peladm update_tradepart`

Syntax	<code>peladm update_tradepart [parameters]</code>
Description	Use this command to modify the characteristics of a Trading Partner.
Parameters	Refer to the parameters for the peladm create tradepart command.

Example	Enter the following command to modify the Trading Partner object named <i>MyTrade</i>: <pre>peladm update_tradepart -n MyTrade -f AS2 -rs CYCL_XIP1REC_AS2 -s N -e N -c N</pre>

Note: There are two types of Trading Partners, Local and Remote which are two distinct objects and must be treated as such. If the Remote Partner is active when modifying the Trading Partner and switching to Local Partner occurs, this may result in abnormal behavior of the product.

Deleting a Trading Partner: peladm delete_tradepart

Syntax	peladm delete_tradepart {-name} [-format] [-alias]	
Description	Use this command to delete a Trading Partner.	
Parameters	Parameter name (and abbreviation)	Description
	-name (-n) (<i>mandatory</i>)	Enter the unique Trading Partner name.
	-format (-f)	Protocol format. Enter one of: • AS1 • AS2 • AS3 • SMIME • RN • CMS (for OFTP) • SWIFTNET
	-alias (-a)	Enter an alias for the Trading Partner object you want to delete.
Example	The following example deletes the AS1 Trading Partner object named <i>user1_as1@domain.com</i>: <pre>peladm delete_tradepart -n user1_as1@domain.com -f AS1</pre>	

[Overview: Trading Partners](#)

[Working with Trading Partners \(GUI\)](#)

[Configuring the Gateway/SWIFTNet environment](#)

Axway Gateway: Managing Events

Managing events on Gateway

This topic introduces Events, Decision Rules and Rule Tables in Gateway.

[About Event Management on Gateway](#)

[Basic example](#)

[Types of triggering event and conditions](#)

[Types of triggered processes](#)

[Rule Table exits](#)

About Event Management on Gateway

You can configure Gateway to execute tasks triggered by different types of event and platform conditions. To do this, you create **Decision Rule** and **Rule Table** objects.

Decision Rule objects

In a Decision Rule object, you define one or more events that you want to trigger a specific action. You also specify the action that is carried out when the event occurs. You can create as many Decision Rules as you require. You place Decision Rules in Rule Tables.

There are four categories of Decision Rules and Rule Tables, as described in [Types of triggering event and conditions](#).

Rule Table objects

A Rule Table object is an ordered list that contains one or more Decision Rules. For each Decision Rule that it contains, the Rule Table displays the following parameters:

- Order
- Decision Rule name
- Action to be executed
- Condition parameters

You arrange Decision Rules in a Rule Table in the order of preferred execution priority. When Gateway handles a transfer event or condition and evaluates the table, it evaluates the table entries from top to bottom in the order of priority. It selects the first Decision Rule that presents a complete conditional match.

Additionally, in a Rule Table object you can:

- Define a default execution type for the Table. If no Decision Rule matches the Transfer parameters, Gateway applies the default execution type.
- For Transfers, restrict processing to a limited set of Transfers, depending on the following conditions:
 - Transfer type: TRANS, MSG ...
 - Transfer mode: Responder, Initiator...
 - Transfer direction: IN, OUT...
 - Transfer state: ENDED, ACKED...
- Set the log level to *No Log*, *Short* or *Full*.

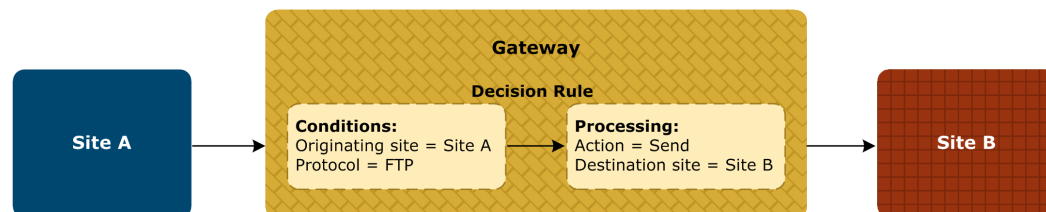
There are four categories of Decision Rules and corresponding Rule Tables, as described in [Types of triggering event and conditions](#).

You can define more than one Rule Table. You can define multiple Rule Tables in the same category or combine one or more Rule Tables of different categories.

Basic example

As a basic example, you can link the reception event of an FTP transfer from a specified site (Site A in the illustration below) to a forwarding process that redirects the file to a designated server (Site B in the illustration below).

You define the triggering conditions and resulting processing for the transfer in a Decision Rule that you create and place in a Rule Table.



Types of triggering event and conditions

You can link many different types of events and conditions to execution processes. Gateway separates these event and condition types into four categories, each with its own set of Decision Rules and Rule Tables. The four different categories are described in the following table.

Category	Use	Example
Transfer Change State	Trigger a specified action using one or more transfer attribute values as the triggering criteria.	Create a Decision Rule specifying: • Transfer Direction: Incoming • Transfer Type: FTP • Originating Site: SiteA To this condition, link an action forwarding the file to a specified server using an outgoing transfer definition model. When Gateway detects an incoming FTP transfer originating from SiteA, it initiates the Transfer Request. For a detailed example of routing using Transfer State Change Decision Rules and Rule Tables, refer to Usage example: Routing.

Category	Use	Example
Directory Scanning	Trigger a specified action using the detection of a specific file or file type in a directory of the Gateway host machine as the triggering criteria.	Create a Decision Rule specifying as a condition that a file named <code>example.txt</code> is present in the directory <code>C:\x.fb_poll*</code>. To this condition, link an action sending the file to a specified monitor via a Transfer Request. When Gateway detects the specified file in the specified directory, it initiates the Transfer Request. For a detailed example of a Transfer triggered by the presence of a file in a specified folder, refer to Usage example: Directory Scanning.

Category	Use	Example
Scheduling	Trigger a specified action at a specific time. Note There is an application architecture issue that affects all time zones when adjusting for DST one hour back: the scheduled events do not run at all during the 1-hour period between the clocks being moved back and the time from which the clocks were originally moved back. (For example: in Brazil, after the time changes from 00:00 to 23:00, between 23:00 and 00:00 - 1-hour period- no scheduled events will run). After this 1-hour period, the events will resume running without requiring any intervention.	Create a Decision Rule specifying that the file <code>todays_log</code> is archived every day at 12 a.m. For a detailed example of the scheduling of a repeated task, refer to Usage example: Scheduling.

Category	Use	Example
XMS	Trigger a process linking an Axway Messaging transfer to a transfer via Gateway.	Link an Axway Messaging connector to an XMS-type Decision Rule. When Gateway detects a transfer over the connector it applies the appropriate Model to the transfer. Refer to: Axway Messaging connector.

In the main window of the GUI, each of the categories is represented by a separate node containing a specific set of menus and windows.

Types of triggered processes

Decision Rules and Rule Tables link the triggering events to Gateway actions. There is a wide variety of actions that Gateway can execute. For example, when triggered by a Decision Rule, Gateway can:

- Execute a batch file or Perl script
- **Caution** Please note that care should be taken when **writing event scripts**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide > Security best practices > [Identified threats and possible mitigations](#)* section for additional information.
- Deposit a Transfer Request
- Purge a Mailbox or archive a Log file
- Apply a Model for a message transfer to an Axway Messaging server (XMS category only)
- ...

Rule Table exits

Gateway automatically associates three categories of Rule Table with the following exit functions that are called during processing.

Rule Table category	Associated exit
Transfer Change State	exitmisc.c

Rule Table category	Associated exit
Directory Scanning	exitfs.c
Scheduling	exittemp.c

Gateway triggers these exits immediately before it scans the corresponding table for matching.

You can modify these Rule Table related exit functions and then locally recompile the functions in order to integrate the modifications into Gateway. This enables you to add additional control over whether or not Gateway analyzes a specific Rule Table. Alternatively you may choose to modify exits in order to apply reusable user variables to Transfer Models or to triggering conditions, or to apply alternate handling processes via APIs.

[Overview: Event management](#)

[Transfer states in Gateway](#)

[Working with Rule Tables and Decision Rules](#)

[Creating an Axway Messaging connector](#)

[Defining Decision Rule conditions](#)

[Working with Decision Rules \(command line\)](#)

[Working with Rule Tables \(command line\)](#)

[Decision Rules and Rule Tables: Parameters List](#)

[Command syntax for Scheduling Event Models \(date syntax\)](#)

[Usage example: Routing](#)

[Usage example: Directory Scanning](#)

[Usage example: Scheduling](#)

[User exits and Rule Tables](#)

Axway Gateway: Managing Events

Working with Rule Tables and Decision Rules

[Creating a Rule Table](#)

[Creating a Decision Rule](#)

[Viewing a Rule Table or a Decision Rule](#)

[Modifying a Rule Table or a Decision Rule](#)

[Deleting a Rule Table or a Decision Rule](#)

[Copying a Rule Table or a Decision Rule](#)

[Refreshing the Decision Rule/Rule Table list display](#)

[Selecting one or more Decision Rules/Rule Tables from a displayed list](#)

Creating a Rule Table

To create a new Rule Table:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. To create:
 - A Scheduling, Directory Scanning or Transfer Change State type Rule Table, right-click the sub-node that represents the corresponding category. A context menu appears. Select **New** then **Rule Table...**
 - An XMS-type Rule Table, you must first [create an Axway Messaging connector](#). After creating the connector, the connector name appears in the Directory Tree structure, listed under the **Event Management >XMS** sub-node. Right-click the connector name for which you want to create the Rule Table. A context menu appears. Select **New** then **Rule Table...**

Gateway displays the *New Rule Table* window.

This window comprises three tabs. The names and the contents of the tabs vary depending on the Rule Table type you are creating. Click the tab names in the following table to see explanations for completing the tabs.

Rule Table type	Tabs
Scheduling	General
	Scheduling Event
	Linked rules
Directory Scanning	General
	File Event
	Linked rules

Rule Table type	Tabs
Transfer Change State	General
	Event
	Linked rules
XMS	General
	XMS Event
	Linked rules

- After completing the tabs, click **OK**.
Alternatively, click **Close** to abandon the creation.

Creating a Decision Rule

To create a new Decision Rule:

- In the left pane of the GUI main window, click  to expand the **Event Management** node.
Gateway expands the node to display the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

- To create:
 - A Scheduling, Directory Scanning or Transfer Change State type Decision Rule, right-click the sub-node that represents the corresponding category. A context menu appears. Select **New** then **Decision Rule...**
 - An XMS-type Decision Rule, you must first [create the related Axway Messaging connector](#) in the GUI.
After creating the connector, the connector name appears in the Directory Tree structure, listed under the **Event Management >XMS** sub-node. Right-click the connector name for which you want to create the Decision Rule. A context menu appears. Select **New** then **Decision Rule...**

Gateway displays the *New Decision Rule* window.

- Complete the *New Decision Rule* window using the following table.

Decision Rule window

Element	Description
Name	Enter a unique name for the new Decision Rule.
Linked	Click the check box to activate or deactivate the Decision Rule.
Type	<i>You cannot modify this field.</i> This field displays the Decision Rule type.
Property list	<i>For Axway Messaging and JMS only</i> Select a Property List to use with this Decision Rule from the drop-down list. The Property List you select in this field determines the contents of the Argument or Name selection box below. An XMS-type Decision Rule operates over an Axway Messaging connector. This connector is associated with one or more Property Lists that define attributes handled over the connector. Note: If you create a Decision Rule and later modify it, it is not possible to modify this field (grayed out).
Comments	<i>Optional</i> Enter a free-text description for the Decision Rule. Maximum: 80 alphanumeric characters. This description appears in the comment column of the Decision Rule list display pane.
Conditions	This pane displays the conditions list. This is a list of expressions that contain the conditions for the triggering of a Decision Rule. The expressions displayed are the results of the information you enter in this section. All JMS properties are preceded by <code>property_</code>.
Argument	Select an argument to use in the triggering criteria expression for your Decision Rule. For JMS: select jms property, then complete the Name field.
Name	<i>JMS only</i> Enter a JMS property name. If you have completed the Property list field above, select a property from the drop-down list.
Operator	Select an operator to use in the triggering criteria expression for your Decision Rule.

Element	Description
Value	Enter the value you want to assign to the Argument in your triggering criteria expression. To use a symbolic variable, precede the variable by the & character. For example, you can enter file name as a symbolic variable: <pre>appli = &(file_name)</pre> When processing the expression, Gateway replaces the symbolic variable with the file name that it takes from the concerned event.
Add	Click Add to insert the current contents of the Argument, Name (<i>JMS only</i>), Operator and Value fields to the conditions list.
Update	To change an entry in the conditions list: 1. Highlight the entry. 2. Modify the values displayed in the Argument, Name (<i>JMS only</i>), Operator and Value fields. 3. Click Update to transfer the new values to the entry in the list.
Delete	To delete an entry in the conditions list, highlight the entry and click Delete.
Clear	Click Clear to remove any values that currently appear in the Argument, Name (<i>JMS only</i>), Operator and Value fields.
Processing (if all conditions match)	

Element	Description
Execution type	Select the processing type to apply to the event: • Model: redirect the Transfer to another transfer whose parameters are defined in a Transfer Model. If you select this value, you must complete the Model field. Additionally, it is recommended that you select the Link input and output transfers option, located to the right of the Model field. • XIB: redirect the Transfer to Axway Integrator or AMTrix for processing • To XMS: redirect the Transfer to an Axway Messaging server queue via an Axway Messaging connector • Mailbox cleaning (<i>for Scheduling-type Rule Tables only</i>): clean the Mailbox via an associated Purge Model • Perl Script: apply a Perl script • Batch command: execute a batch command or script • No action: implement no routing
Model	This field is displayed if you select either <i>Model</i> or <i>To XMS</i> as the Execution type. • For <i>Model</i> execution types: Enter the name of the Model to apply or select a Model from the drop-down list. Keep in mind that the Model name must correspond to a General Model. • For <i>To XMS</i> execution types: Enter the name of an XMS-type Model.
XMS connector out	This field is displayed if you select <i>To XMS</i> as the Execution type. Select the name of the Axway Messaging connector.
Link input and output transfers	<i>Not for all types of Decision Rule.</i> This field is displayed if you select <i>Model</i> as the Execution type. Select this option to create a logical link between the incoming and the outgoing Transfers.
Purge Model	This field is displayed if you select <i>Mailbox cleaning</i> as the Execution type for a Scheduling-type Decision Rule. Select the name of the Purge Model that you want to associate with the execution type.

Element	Description
Perl script	This field is displayed if you select <i>Perl Script</i> as the Execution type. Enter the script to apply to the current Transfer. The variable <code>p_perldll</code> must contain the location of the Perl library on the current machine. For more information, refer to the readme in the <code>run_time\scripts\perl</code> folder.
Batch command	This field is displayed if you select <i>Batch command</i> as the Execution type. Enter the command to apply to the current Transfer. Alternatively, enter the path and file name of the script to execute. Refer to Executing scripts as command lines from Decision Rules.

Viewing a Rule Table or a Decision Rule

To view a Rule Table or a Decision Rule:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. Click the sub-node that corresponds to the type of Rule Table or Decision Rule you want to view. For example, to view a Scheduling type Rule Table, click the **Scheduling** sub-node.

In the right pane, Gateway displays a list of the Rule Tables and Decision Rules that exist for the sub-node category.

3. In the right pane, right-click the name of the Rule Table or Decision Rule you want to view.
4. Select **View...** in the context menu.

Gateway displays a read-only version of the *Rule Table* or *Decision Rule* window, including all relevant tabs.

5. Use this window as necessary to view your Rule Table or Decision Rule configuration.
6. When finished, click **Close**.

Modifying a Rule Table or a Decision Rule

To modify an existing Rule Table or a Decision Rule:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. Click the sub-node that corresponds to the type of Rule Table or Decision Rule you want to modify. For example, to modify a Scheduling type Rule Table, click the **Scheduling** sub-node.

In the right pane, Gateway displays a list of the Rule Tables and Decision Rules that exist for the sub-node category.

3. In the right pane, right-click the name of the Rule Table or Decision Rule you want to modify.
4. Select **Modify...** in the context menu.

Gateway displays a *Rule Table* or *Decision Rule* editing window, including all relevant tabs.

5. Use this window to modify your Rule Table or Decision Rule.
6. When finished, click **OK**.

Deleting a Rule Table or a Decision Rule

To delete an existing Rule Table or a Decision Rule:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. Click the sub-node that corresponds to the type of Rule Table or Decision Rule you want to delete.

For example, to delete a Scheduling type Rule Table, click the **Scheduling** sub-node.

In the right pane, Gateway displays a list of the Rule Tables and Decision Rules that exist for the sub-node category.

3. In the right pane, right-click the name of the Rule Table or Decision Rule you want to delete.
4. Select **Delete** in the context menu.

Gateway displays a dialog box enabling you to confirm or abandon the delete process.

5. Click **Yes** to confirm deletion.

Gateway closes the dialog box and deletes the Rule Table or Decision Rule. The line representing the Rule Table or Decision Rule is removed from the Decision Rule/ Rule Table list display in the right pane.

Copying a Rule Table or a Decision Rule

To make copy of an existing Rule Table or a Decision Rule in the same display list as the original Rule Table or Decision Rule:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node.
Gateway expands the node to display the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. Click the sub-node that corresponds to the type of Rule Table or Decision Rule you want to copy.
For example, to copy a Scheduling type Rule Table, click the **Scheduling** sub-node.

In the right pane, Gateway displays a list of the Rule Tables and Decision Rules that exist for the sub-node category.

3. In the right pane, right-click the name of the Rule Table or Decision Rule you want to copy.
4. Select **Copy...** in the context menu.

Gateway displays a *Copying* dialog box.

5. In the *Copying* dialog box, enter a unique name for the copy of the Rule Table or Decision Rule.
Then click **OK** to confirm the copy process.

Gateway closes the dialog box and creates a copy of the Rule Table or Decision Rule. It adds a new line representing the new Rule Table or Decision Rule to the Decision Rule/ Rule Table list display in the right pane. The parameters of the copy are identical the original, except for the new name. To modify parameters of the copy, use the [Modify...](#) command.

Refreshing the Decision Rule/Rule Table list display

To refresh the display of Rule Tables and/or Decision Rules in the display pane:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. In the Directory Tree structure, right-click the sub-node that corresponds to the type of Rule Table and/or Decision Rule list you want to refresh.

For example, to refresh the view of a Scheduling type list of Rule Tables and Decision Rules, right-click the **Scheduling** sub-node.

Gateway displays a context menu.

3. In the context menu, select **Refresh**.

Gateway refreshes the list of Rule Tables and Decision Rules displayed in the right pane.

Selecting one or more Decision Rules / Rule Tables from a displayed list

To filter the display of Rule Tables and/or Decision Rules in the display pane according to specific criteria:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. Right-click the sub-node that corresponds to the type of Rule Table and/or Decision Rule list you want to filter.

For example, to filter a Scheduling-type list of Rule Tables and Decision Rules, right-click the **Scheduling** sub-node.

Gateway displays a context menu.

3. In the context menu, select **Select....**

Gateway displays the *Select Rule Tables/Decision rules* window.

4. In this window, enter values in the fields that you want to use as filtering criteria for a new display list. You can enter multiple criteria, and you can combine filtering of Rule Tables and Decision Rules. The following table lists the fields you can use to filter the display.

Field	Description
Rule tables selection	
Name	Enter a Rule Table name to filter Rule Tables in the list display.
Execution	Select an execution type to filter Rule Tables in the list display.
Decision rules selection	
Name	Enter a Decision Rule name to filter Decision Rules in the list display.
Comments	Enter a comment to filter Decision Rules in the list display. The comment must be the complete text of the comment field.
Execution	Select an execution type to filter Decision Rules in the list display.

5. After completing the fields as required, click **OK**.

Gateway refreshes the list display in the right pane, filtering entries according to the specified criteria.

Axway Gateway: Managing Events

Creating an Axway Messaging connector

[About Axway Messaging connectors](#)

[Creating an Axway Messaging IN connector](#)

[Creating an Axway Messaging OUT connector](#)

About Axway Messaging connectors

Before you create an XMS-type Decision Rule or XMS-type Rule Table, you need to create an Axway Messaging connector in the GUI. You can create either an *IN connector* or an *OUT connector*, as described in the sections below.

After you create a connector, it is visible as a node in the left pane Directory tree structure of the Gateway main window, under the **Event Management** node. The connector is listed under either the **XMS Connector In** or **XMS Connector Out** sub-node.

Creating an Axway Messaging IN connector

To create a new Axway Messaging IN connector:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node.
Gateway expands the node to display the following sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS
 Each sub-node represents a [Decision Rule / Rule Table category](#).
2. Click  to expand the **XMS** sub-node.
3. Right-click **ConnectorIn**, then select **New** from the context menu.
Gateway displays the *XMS Connector In* window.
4. Complete the fields of the *XMS Connector In* window using the information in the following table.
5. Click **OK** to confirm the data you have entered.
Gateway:
 - closes the *XMS Connector In* window
 - creates the connector
 - displays a new entry in the Directory tree structure with the assigned connector name
 - displays a new line entry in the *Connector In* list in the main window

XMS Connector In window

Element	Description
Name	Enter a unique name for the connector.
Type	<i>You cannot modify this field.</i> This field displays the value <i>In</i> , indicating that you are creating an IN type connector.
Active	Click the check box to activate or deactivate the connector.
XMS (Axway Messaging) server	
Name	Enter the name of the Axway Messaging server to which you are connecting.
Port	Enter the Axway Messaging server access port number.
Login	
Identifier	Enter the login identifier that Gateway uses to log in to the Axway Messaging server.

Element	Description
Set password	This button opens a dialog box to enter the password that you want to associate with the Identifier.
Messages	
Type	Select one of the values: • XMS message = the connector handles Axway Messaging messages • XMS notif error = the connector handles Axway Messaging error and acknowledgment messages
Standard size	Enter the maximum message size to be handled.
Queue name	Enter the name of the Axway Messaging queue in which the Gateway reads messages.
Reception directory	Enter the directory in which Gateway stores Axway Messaging data content before redirecting it.
Log level	Select one of the Axway Messaging logging level values: • No log = logging deactivated • Short = short logs • Full = detailed logs
Backup message	Select this option if you want to save a copy of messages that were filtered and not handled by Decision Rules and Rule Tables. Messages are saved to the Axway Messaging file <code>XMSReceptionQueue bck</code> .

Creating an Axway Messaging OUT connector

To create a new Axway Messaging OUT connector:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule / Rule Table category](#).

2. Click  to expand the **XMS** sub-node.
3. Right-click **ConnectorOut**, then select **New** from the context menu. Gateway displays the *XMS Connector Out* window.

4. Complete the fields of the *XMS Connector Out* window using the information in the following table.
5. Click **OK** to confirm the data you have entered. Gateway:
 - closes the *XMS Connector Out* window
 - creates the connector
 - displays a new entry in the Directory tree structure with the assigned connector name
 - displays a new line entry in the *Connector Out* list in the main window

XMS Connector Out window

Element	Description
Name	Enter a unique name for the connector.
Type	<i>You cannot modify this field.</i> This field displays the value <i>Out</i> , indicating that you are creating an OUT type connector.
Active	Click the check box to activate or deactivate the connector.
XMS (Axway Messaging) server	
Name	Enter the name of the Axway Messaging server to which you are connecting.
Port	Enter the Axway Messaging server access port number.
Login	
Identifier	Enter the login identifier that Gateway uses to log in to the Axway Messaging server.
Set password	This button opens a dialog box to enter the password that you want to associate with the Identifier.
Messages	
Compression	Select this option if you want incoming files transferred to Axway Messaging to be compressed. If you do not want to use the message compression function, leave this box unselected.

Element	Description
Persistent	Select this option if you want the Axway Messaging server to operate in persistent mode. Persistent mode: Until the receiving application consumes the message, the receiving server stores a copy of the message in the Axway Messaging Log. If the connection between the receiving application and the receiving server fails, the receiving server does not lose the message. Non-persistent mode: The receiving server <i>does not</i> store a copy of the message in the Axway Messaging Log. If the connection between the receiving application and the receiving server fails, the receiving server loses the message.
Priority	Select one of the following values to indicate the priority an application uses for getting messages from a source: • High • Normal • Low
Log level	Select one of the Axway Messaging logging level values: • No log = logging deactivated • Short = short logs • Full = detailed logs

Axway Gateway: Managing Events

Defining Decision Rule conditions

[About Decision Rule conditions](#)

[Condition parameters](#)

[Operators](#)

[SMTP values in Decision Rule conditions](#)

[Completing the Value field](#)

[Using symbolic variables](#)

[Using user functions](#)

[Case-insensitive conditions](#)

About Decision Rule conditions

Each Decision Rule condition that you define is an expression that comprises:

- **Argument:** To complete the Argument component of an expression, click the arrow on the **Argument** selection box and then select an argument from the drop-down list. The arguments displayed vary according to the type of Decision Rule you are creating or modifying, and, in the case of Transfers, the type of Transfer. The list contents vary to take into account, for example:
 - The CGate selected on input
 - The use of a VFD
 - A transfer via an Axway Messaging connector

Argument values in Decision Rules can comprise a combination of string values, user functions and [symbolic variables](#). User functions can also be called inside one another, as long as the returned values are compatible with the argument expected in the upper function.

This substitution mechanism applies to:

- Model names, when the selected **Execution type** is **Model**
- Directory paths and file patterns in the Trading Partner configuration

Note: The number of functions you can nest is limited.

The complete list of possible values for non-XMS (Axway Messaging) events is provided in [Condition parameters](#) below. In the case of Axway Messaging, the attribute list is derived dynamically from the processed message. It is therefore not possible to provide all potential elements of the list in this documentation.

- **Operator:** To complete the Operator component of an expression, click the arrow on the **Operator** selection box and then select an operator from the drop-down list that is displayed. The available operators (=, >, < and so on) vary according to the type of Argument you selected. The complete list of operators is given in [Operators](#) below.
- **Value:** A value of variable length. This value can be a symbolic variable.

Example

Argument	Operator	Value
appli	=	FTP_A

Note: A condition is identified within a given Decision Rule via the name *Argument* of the input argument to test. This means that a condition involving a given input argument can only be used once in a given Decision Rule condition.

Condition parameters

Each time you enter a value, Gateway checks that the string entered is syntactically correct and that it corresponds to the data type of the field. For example, it checks that a string only contains numeric characters for a field of the type integer. This type checking is only applicable to static strings, that is, that do not contain symbolic variables.

The following table lists the parameters you can use in Decision Rule conditions.

Note: Parameters for use specifically with SWIFTNet are listed in a separate table.

Transfer Change State Decision Rule condition parameters

Parameter	Real type	Description
acknowledgement_option	Select from: - TO_ACK - NOT_TO_ACK	option for the acknowledgment of the transfer
appli	Allocation unit	Application name
appli_user_param1		User variables that relate to the Application
appli_user_param2		
block_size	Integer	Block size
called_address		Output of CGate or CGateGroup
called_SAP		Output of CGate or CGateGroup
cgate_user_param1		User variables that relate to the selected CGate
cgate_user_param2		
content_id		X.420 parameter
content_type	String[12]	Mime content type of the attached file So far, only available for EDIINT and S/MIME files.
data_code	Integer	Data character set
dest_alias	String[25]	Destination

Parameter	Real type	Description
destination	String[25]	Protocol destination name
dir_path	String[128]	Directory path
direction	Select from: • In • Out	Direction. Since Decision Rule processing can apply to both input and output Transfers, it is highly recommended that you set this parameter to specify whether to apply the Rule to the incoming or outgoing Transfer. If you do not set the direction parameter, the Decision Rule will loop continuously on both the input and output Transfer.
dir_name	String[254]	VFD deposit directory
f_bytes	Integer	Number of file data bytes transferred
file_component	String[127]	File name
file_label	String[80]	File description
file_name	String[128]	Logical file name
file_org		File organization
file_size	Integer	File size in octets
file_type	String[17]	File type (protocol)
full_state		Full transfer state value
group		
http_host_name		CGate and CGateGroup output

Parameter	Real type	Description
integer_user_param1	Integer	User variables set by the C or Perl exit
integer_user_param2		
integer_user_param3		
integer_user_param4		
integer_user_param5		
ipmid_user_id	String	X.420 parameter
last_end_reason	Integer	Last end reason for transfer
local_agent	String[32]	Local Site
message_id	String[256]	Mime Message-Id of the attached file Presently, only available for EDIINT and S/MIME files.
mail_xpriority	Integer	SMTP/POP3 X-Priority
mode	Select from: • Initiator • Responder • Both	Mode
mts_id	String	X.420 parameter
multi_nb_ok	Integer	The number of successful child transfers of a Multi-transfer. Refer to Transferring multiple files with FTP on page 1091 for further information.

Parameter	Real type	Description
multi_nb_total	Integer	The total number of child transfers of a Multi transfer. Refer to Transferring multiple files with FTP on page 1091 for further information.
multi_state		State of a Multi transfer. Refer to Transferring multiple files with FTP on page 1091 for further information.
newline_convention (-nlc)	Select from: • U(nix) • W(indows)	SFTP only. Line feed convention for transfer, depending on remote system type
org_alias	String[25]	Origin
originator	String[32]	Protocol name originator
p1.originator		X.420 parameter
p1.recipient		X.420 parameter
p2.ipmId.user		X.420 parameter
p2.bodyPartType		X.420 parameter
p2.importance		X.420 parameter
p2.sensitivity		X.420 parameter
padding		Padding character code
param1	String[41]	User parameter no. 1
param2	String[41]	User parameter no. 2
protocol		Protocol
rcv_appli		Receiver Application name
rcv_msg		Received message
rcv_text		Receiver free text
rcv_user		Receiver user name

Parameter	Real type	Description
rec_count	Integer	Number of records
rec_fmt		Record format
rec_len	Integer	Record length
receipt_nack_msg	String[512]	Receipt negative or warning description Only available for EDIINT and S/MIME files.
receipt_reply_by_xfer		Local identifier for the receipt sent for the current transfer
receipt_reply_to_xfer		Local identifier for the message that the current receipt acknowledges
receipt_request		Receipt request option used for transfer
receipt_request_sync		Receipt requested synchronously
receipt_request_to	String[256]	Receipt return address
receipt_sent_mode	Select from: - MANUAL - AUTOMATIC	Receipt send mode
receipt_subtype	Select from: - WARNING - ERROR - FAILURE	Receipt sub type
receipt_type	Select from: - SUCCESS - FAILURE	Receipt type
recipient_receipt_notif_request	Select from: - None - NRN - RN_NRN	X.420 parameter

Parameter	Real type	Description
recipient_reply_request	Y or N	X.420 parameter
remote_agent	String[32]	Remote Site
route_state	Select from: • FAILED • NOT_ROUTED • PROCESSING • TO_ROUTE • ROUTED	
signed_eerp	Y or N	Is signed EERP required? (for OFTP R2.0)
sap		TCP/IP local connection port
site_user_param1		User variables that relate to the Local Site
site_user_param2		
smtp_bcc		<i>Bcc</i> header field of the received SMTP message, when a single addressee is specified.
smtp_cc		<i>Cc</i> header field of the received SMTP message, when a single addressee is specified.
smtp_from		<i>From</i> header field of the received SMTP message.
smtp_subject		<i>Subject</i> header field of the received SMTP message.
smtp_to		<i>To</i> header field of the received SMTP message, when a single addressee is specified.
snd_appli		Sender Application name
snd_text		Sender free text
snd_user		Sender user name

Parameter	Real type	Description
state		
string_user_param1		
string_user_param2		
string_user_param3	String	User variables set by the C or Perl exit
string_user_param4		
string_user_param5		
text_file		
time_of_last_access		
tls_cipher_suite		TLS cipher suite
tls_client_auth		Client authentication
tls_server_auth		Server authentication
tls_sprof		Security profile
tmp_dea_id	String	Trading partner management DEA identifier
trade_compress		File is compressed
trade_compress_algo	ZLIB	Compression algo
trade_delivery_mode	Select from: • RT • SNF	Indicates the delivery mode of the file or message
trade_dest	String[128]	Destination trading partner name
trade_dest_alias	String[31]	Destination trading partner alias
trade_encrypt		File is encrypted

Parameter	Real type	Description
trade_encrypt_dn	String[100]	Name used for encryption in Initiator mode.
trade_encryption_algo		File encryption algorithm
trade_file_encoded		File is in trade format or not
trade_format		Trading format used
trade_key_encryption_algo	RSA	Key encryption algo
trade_org	String[128]	Originator trading partner name
trade_org_alias	String[31]	Originator trading partner alias
trade_receipt_content_type	String[256]	Content type returned in receipt (S/MIME only)
trade_receipt_message_id	String[256]	Message ID acknowledged by the receipt
trade_receipt_mic	String[40]	MIC of the receipt
trade_receipt_mic_algo		MIC calculation algorithm requested for the receipt
trade_receipt_recipient	String[128]	Recipient acknowledged by the receipt
trade_receipt_request_type	String[30]	Type of acknowledgement request.
trade_receipt_sign_algo		Signing algorithm requested for the receipt
trade_receipt_signature	String[1024]	Signature returned in receipt (S/MIME only)
trade_remote_destination		Destination partner is a remote partner
trade_request_type	String[30]	Type of request
trade_sign		File is signed

Parameter	Real type	Description
trade_sign_algo	Select from: - MD5withRSA - SHA1withRSA	Signing algo
trade_sign_dn	String[100]	Name used for signing in Initiator mode. Otherwise the securityDN (supplied by SNL) is used.
trade_state		Trading process state
type		Request type (value in the condition: TRANS, EERP and so on)
ua_rn_id	String	
unprocessed_dir_path		Directory path of the unprocessed file (clear file that is not yet encapsulated in EDIINT)
unprocessed_file_component		File name of the unprocessed file
username		
user_processed		Yes or No
user_state		
virtual_file_desc	free text	Virtual file description (for OFTP R2.0)
x_bytes	Integer	Number of bytes transferred
x_data_code		Transferred data code
x_rec_fmt		Transfer format
x_rec_len	Integer	Transfer record length
xfer_data_code		Transfer data code

Parameter	Real type	Description
xfer_duplicated	Y or N	Used to filter duplicate incoming SwiftNet FileAct RealTime transfers. The "xfer_duplicated" argument is linked with the "duplicated from xfer" transfer parameter: "xfer_duplicated=Y" means that "duplicated from xfer" is set in the transfer. Note: The "xfer_duplicate=Y" does not necessarily mean that the transfer is a duplicate. For example, a put at disposal permanent transfer will generate outgoing child transfers with "duplicate from xfer" pointing to the permanent put at disposal transfer.
xfer_ident	String[48]	
xfer_rec_fmt		Transfer record format
xfer_rec_len	Integer	Transfer record length
yday	Integer	Day of the current year of the transfer

Additional Decision Rule condition parameters for SWIFTNet

The following Decision Rule condition parameters are specifically for use with SWIFTNet:

Parameter	Real type	Description
header_info	Y or N	Has Header Info?
sw_add_params	String	Additional SwiftNet parameters file
sign_list	Y or N	Has Signature List?

Parameter	Real type	Description
sw_auth_indicator		Refer to SWIFTNet File and Message Copy .
sw_copy_indicator		Refer to SWIFTNet File and Message Copy .
sw_copy_state		Refer to SWIFTNet File and Message Copy .
sw_copy_type		Refer to SWIFTNet File and Message Copy .
sw_message_id	String[40]	Overwrite the message id with specified value.
sw_pd_indication		Make the responder aware that the incoming transfer may be a duplicate of another transfer. Refer to Possible Duplicate management for SWIFTNet .
trade_non_repudiation		Request SWIFT to keep proof of sending.
trade_notif_queue	String[30]	Reception queue for Store-and-Forward delivery notifications.
sw_retrieved_message	Y or N	Is a Retrieved Message?
sw_retrieved_message_direction	Select from: - received - sent	Retrieved Message direction
sw_third_party_dn	String	Third-Party DN
trade_service	String[30]	SWIFT business service name.

Operators

This section describes the set of operators that you can use in Decision Rule expressions. The operators available depend on the type of argument that you set in the **Argument** field.

Since SMTP fields typically contain a list of string values, the way Gateway handles Decision Rule expressions that contain SMTP arguments is slightly different. Refer to [SMTP values in Decision Rule conditions](#).

Operator	Meaning	Value field type
=	equals	- string: the value can contain wildcard characters "*" and "?" - enumerated - numeric
#	does not equal	- string: the value can contain wildcard characters "*" and "?" - enumerated - numeric
<	less than	Comparison operators: - string: compare alphabetic order - numeric
>	greater than	
=<	equal to or less than	
>=	greater than or equal to	
= NULL	is null	
# NULL	is not null	
< >	includes a set of values	- string - enumerated - numeric
> <	excludes a set of values	- string - enumerated - numeric
[]	ArgumentIn must be greater than or equal to the first value in the list and less than or equal to the second value	numeric: the list must include two values separated by a comma (,)

Operator	Meaning	Value field type
] [	ArgumentIn must be less than the first value or greater than the second value in the list	numeric: the list must include two values separated by a comma (,)
exist	<i>JMS only</i> exists	
not exist	<i>JMS only</i> does not exist	

SMTP values in Decision Rule conditions

Since SMTP fields typically contain a list of string values, the condition is valid (or invalid in the case of a negative operator), if *one* of the SMTP values corresponds to the value of the expression.

The following table describes how Gateway processes Decision Rule conditions, where the `smtp_to` field in the incoming Transfer contains the values *Smith*, *Jones* and *Black*.

Decision Rule condition	Valid	Description
smtp_to = Smith	YES	Smith is one of the values in the incoming Transfer smtp_to field.
smtp_to = Lloyd	NO	Lloyd is not a value listed in the incoming Transfer smtp_to field.
smtp_to # Lloyd	YES	No name in the smtp_to field of the incoming Transfer is equal to Lloyd.
smtp_to # Smith	NO	Smith is one of the names in the incoming Transfer smtp_to field.
smtp_to <> Smith,Lloyd,Swanson	YES	One of the values in the incoming Transfer corresponds to one of the values in the Decision Rule expression (Smith).

Decision Rule condition	Valid	Description
smtp_to <> Lloyd,Swanson,Kye	NO	No smtp_to value in the incoming Transfer corresponds to any of the values in the Decision Rule expression.
smtp_to >< Lloyd,Swanson,Kye	YES	None of the names specified corresponds to any value in the smtp_to field in the incoming Transfer.
smtp_to >< Jones,Lloyd,Kye	NO	Jones is one of the smtp_to values in the incoming Transfer.

Scheduling Decision Rule condition parameters

Parameter	Real type	Description
integer_user_param1	Integer	User variables set by the C or Perl exit
integer_user_param2		
integer_user_param3		
integer_user_param4		
integer_user_param5		
string_user_param1	String	User variables set by the C or Perl exit
string_user_param2		
string_user_param3		
string_user_param4		
string_user_param5		

Directory Scanning Decision Rule condition parameters

Parameter	Real type	Description
file_name	String[255]	File name
file_size	Integer	File size in bytes
integer_user_param1	Integer	User variables set by the C or Perl Exit
integer_user_param2		
integer_user_param3		
integer_user_param4		
integer_user_param5		
string_user_param1	String	User variables set by the C or Perl Exit
string_user_param2		
string_user_param3		
string_user_param4		
string_user_param5		
last_modification	Datetime	Time of the last data change in the monitored folder

Completing the Value field

To complete the **Value** field, you can:

- Enter a value directly
- Select a defined object, such as a Site, from the drop-down list
- Select a symbolic variable from the drop-down list
- Use user functions

Using symbolic variables

To retrieve a value from the current Transfer Request, you can use *symbolic variables*. Gateway replaces the symbolic variable with the corresponding value in the current Transfer.

The syntax is **&(Variable_name)**

For example, if you set the condition: **Destination # &(Originator)**, Gateway:

- Identifies the originator Site in the current Transfer
- Replaces the variable **&(Originator)** with this value
- Checks that the destination Site is not the same as the originator Site

In particular, it is recommended that you use symbolic variables in the Model definition if you set the **Execution type** to **Model**. For example:

- file_component = &(dir_path)/&(file_component)
- application = &(appli)
- destination_alias = &(dest_alias)

To retrieve the system date and time, use the following symbolic variables:

- **&(DATE)**: to retrieve the system date, for example: 20070121
- **&(TIME)**: to retrieve the system time, for example: 101402

Symbolic variables for Transfer Change State Decision Rules

You can use the following values as symbolic variables for Transfer Change State Decision Rules:

- acknowledgment_option
- alloc_count
- alloc_size
- alloc_unit
- append
- appli
- application
- appli_user_param1
- appli_user_param2

- attach_extract_file
- block_size
- called_addr
- cgate_user_param1
- cgate_user_param2
- compression
- content_id
- content_type
- counts_0
- counts_1
- counts_2
- counts_3
- counts_4
- counts_5
- counts_6
- counts_7
- counts_8
- counts_9
- create_date
- data_code
- DATE
- date_begin
- date_create
- date_end
- date_to_begin
- date_to_end
- destination
- dest_alias
- direction
- dir_name
- dir_path
- dir_type
- dn_request
- dup_from_xfer
- etb5_auth_type

- etb5_ciph_type
- etb5_id_bank
- etb5_id_bank_sec
- etb5_id_client
- etb5_id_client_sec
- etb5_memo_type
- etb5_paraf_used
- etb5_rsign1_used
- etb5_rsign2_used
- etb5_seal_type
- etb5_sign_type
- etb5_userarea
- exch_id
- extract_date
- file_ciphored
- file_component
- file_label
- file_name
- file_org
- file_size
- file_sys_dep_param
- file_type
- ftp_command
- full_state
- f_bytes
- generate_xfb_subject
- group
- header_info
- http_host_name
- ident
- integer_user_param1
- integer_user_param2
- integer_user_param3
- integer_user_param4
- integer_user_param5

- interval
- ipmid_user_id
- key_len
- key_offset
- last_end_diag
- last_end_err
- last_end_reason
- local_agent
- local_ident
- mail_xpriority
- master
- max_alloc_size
- max_file
- max_files
- max_request
- max_requests
- message_id
- mode
- model
- msg1
- msg2
- mts_id
- multi_nb_ok
- multi_nb_total
- multi_state
- nack_action
- nb_canceled
- nb_ended
- nb_frozen
- nb_progressing_servicing
- nb_suspended
- nb_total
- nb_to_begin
- newline_convention
- next_xfer

- org_alias
- originator
- padding
- param1
- param2
- path_name
- permanent
- prev_xfer
- priority
- protocol
- proto_spec_str
- purge_option
- rcv_appli
- rcv_msg
- rcv_text
- rcv_user
- receipt_nack_msg
- receipt_reply_by_xfer
- receipt_reply_to_xfer
- receipt_request
- receipt_request_sync
- receipt_request_to
- receipt_sent_mode
- receipt_subtype
- receipt_type
- recipient_receipt_notif_request
- recipient_reply_request
- rec_count
- rec_fmt
- rec_fmt
- rec_len
- rec_len
- remote_agent
- reply_by_xfer
- reply_to_xfer

- request_method
- response_code
- response_request_sync
- response_to_xfer
- retry_count
- retry_count_max
- route_from_xfer
- route_state
- route_to_xfer
- sap
- select_req
- sentinel_transfer_filter
- sentinel_xferident
- sentinel_xferobjectname
- sequence
- sign_list
- site_user_param1
- site_user_param2
- smtp_bcc
- smtp_cc
- smtp_from
- smtp_subject
- smtp_to
- snd_appli
- snd_msg
- snd_text
- snd_user
- state
- string_user_param1=
- string_user_param2=
- string_user_param3=
- string_user_param4=
- string_user_param5=
- sw_add_params
- sw_auth_by_xfer

- sw_auth_indicator
- sw_auth_to_xfer
- sw_copy_indicator
- sw_copy_stateone
- sw_copy_typeone
- sw_message_id
- sw_payload_type
- sw_pd_indication
- sw_retrieved_message
- sw_retrieved_message_direction
- sw_third_party_dn
- sys_dep
- text_file
- TIME
- tls_cipher_suite
- tls_client_auth
- tls_server_auth
- tls_sprof
- total_files
- total_requests
- tpm_dea_id
- tpm_local_dea_id
- trade_compress
- trade_compress_algo
- trade_delivery_mode
- trade_dest
- trade_dest_alias
- trade_encrypt
- trade_encryption_algo
- trade_encrypt_dn
- trade_fdacenter
- trade_fdasubmissiontype
- trade_file_encoded
- trade_format
- trade_key_encryption_algo

- trade_non_repudiation
- trade_notif_queue
- trade_org
- trade_org_alias
- trade_processing_error
- trade_receipt_content_type
- trade_receipt_message_id
- trade_receipt_mic
- trade_receipt_mic_algo
- trade_receipt_recipient
- trade_receipt_request_type
- trade_receipt_signature
- trade_receipt_sign_algo
- trade_remote_destination
- trade_request_type
- trade_rn_business_code
- trade_rn_document_name
- trade_rn_document_type
- trade_rn_message_id
- trade_rn_mic
- trade_rn_process_id
- trade_rn_transaction_id
- trade_rn_version
- trade_service
- trade_sign
- trade_sign_algo
- trade_sign_dn
- trade_state
- truncating_allowed
- type
- ua_rn_id
- unique_filename
- uniq_filename
- unprocessed_dir_path
- unprocessed_file_component

- username
- user_processed
- user_state
- xfer_duplicated
- xfer_ident
- xfer_rec_fmt
- xfer_rec_len
- xntf_filter
- xntf_profile
- x_bytes
- x_data_code
- yday

Symbolic variables for Scheduling Decision Rules

You can use the following values as symbolic variables for Scheduling Decision Rules:

- rule_table_id
- event_occur_time
- DATE
- TIME
- Yday
- integer_user_param1
- integer_user_param2
- integer_user_param3
- integer_user_param4
- integer_user_param5
- string_user_param1
- string_user_param2
- string_user_param3
- string_user_param4
- string_user_param5

Symbolic variables for Directory Scanning Decision Rules

You can use the following values as symbolic variables for Directory Scanning Decision Rules:

- DATE
- TIME
- yday
- dir_name
- dir_path
- file_component
- file_name
- file_size
- integer_user_param1
- integer_user_param2
- integer_user_param3
- integer_user_param4
- integer_user_param5
- last_modification
- rule_table_id
- string_user_param1
- string_user_param2
- string_user_param3
- string_user_param4
- string_user_param5

Using user functions

You can use the following user functions to complete the Decision Rule **Value** field:

- [strrep\(\)](#)
- [substr\(\)](#)
- [substring\(\)](#)
- [sepstring\(\)](#)
- [uppercase\(\)](#)
- [lowercase\(\)](#)

These functions are described in the following tables:

strrep()

Purpose This function enables you to replace a substring of a source string with another substring. All the occurrences of the substring in the source string are replaced.

Syntax `strrep(source, old_substring, new_substring)`

where:

- *source* = the source data to be searched
source must be in the format `&(var_name)` where *var_name* is a symbolic variable name representing a field in the transfer record. There must be no embedded space in the expression `&(var_name)`. At run time, `&(var_name)` is substituted by its actual data.
- *old_substring* = substring to be replaced from the initial source string (one or more characters)
- *new_substring* = substring replacing all the occurrences (one or more characters)

Example In the following examples, `&(var)` has the value: `e:/program files/axway/Gateway/v613/bin/test.exe`

Enter:	The function returns:
<code>strrep(&(var), "/", "\\")</code>	<code>e:\program files\axway\Gateway\v613\bin\test .exe</code>
<code>strrep(&(var), '!', '@')</code>	<code>e:/program files/axway/Gateway/v613/bin/test .exe</code>
<code>strrep(&(var), ' .exe', ' .ini')</code>	<code>e:/program files/axway/Gateway/v613/bin/test .ini</code>
<code>strrep(&(var), "/", "//*")</code>	<code>e://program files//axway//Gateway//v613//bin/ /test.exe</code>
<code>strrep(&(var), "program files", "")</code>	<code>e://axway/Gateway/v613/bin/test.e xe</code>

substr ()

Purpose This function enables you to extract a sub-section of a source string. The function extracts a substring of the variable that you specify in the *source* parameter, starting from the position that you indicate via the *from* parameter and stopping at the position that you indicate via the *to* parameter.

substr()**Syntax** `substr(source, from, to)`

where:

- *source* = the source data to be searched
source must be in the format `&(var_name)` where *var_name* is a symbolic variable name representing a field in the transfer record. There must be no embedded space in the expression `&(var_name)`. At run time, `&(var_name)` is substituted by its actual data.
- *from* = the starting position in the source from which to extract the substring
- *to* = the end position in the source for the substring extraction

The *from* and *to* expressions must take one of the following two formats:

- A numeric expression: *n*, *-n*, *+n*, ***
n denotes the position relative to the beginning of the *source* (Relative to 1)
-n denotes the position relative to the end of the *source* (-1 is the last position of the source, -2 is next before last, and so on)
+n denotes the length of the characters to be copied, starting from a previously evaluated position. It has meaning only in the *to* expression. In the *from* expression, it is equivalent to *n*
*** denotes the end of the *source*
- A string expression: *+n'string-to-search'*
'*string-to-search*' comprises one or more characters denoting the string-pattern to search. It must be enclosed between a pair of single quotes (') or double quotes (").
n is optional. If present, it denotes the next *n*th occurrence of *string-to-search* within the *source*. Otherwise the first occurrence is assumed. If *n* is ***, then the last occurrence is searched.
+ is optional. If present, the *string-to-search* is also included in the substring returned, otherwise it is not included.

In the *from* expression, the search starts from the beginning of the source. In the *to* expression, the search starts at the last position located in the *from* expression.

substr()

Examples In the following examples, `&(var)` has the value:

`e:/program files/axway/Gateway/v613/bin/test.exe`

Enter:	The function returns:
<code>substr(&(var), 1, *)</code>	<code>e:/program files/axway/Gateway/v613/bin/test.exe</code>
<code>substr(&(var), 1, 2)</code>	<code>e:</code>
<code>substr(&(var), +2('/', +'/')</code>	<code>/axway/</code>
<code>substr(&(var), *"/", *)</code>	<code>test.exe</code>
<code>substr(&(var), '/bin/', *)</code>	<code>test.exe</code>
<code>substr(&(var), -3, *)</code>	<code>exe</code>
<code>substr(&(var), -8, -5)</code>	<code>test</code>

substring()

Purpose This function extracts a substring of the variable name that you specify, starting from the position that you indicate via the `startIndex` parameter and stopping at the position that you indicate via the `stopIndex` parameter.

Syntax `substring(&(variableName), startIndex, stopIndex)`
 or
`substring(&(variableName), startIndex)`

substring()

Examples If the Destination in the current Transfer is FTP_DEST, the function:

```
substring(&(destination), 2, 5)
```

returns: TP_D

If the Destination in the current Transfer is FTP_DEST, the function:

```
substring(&(destination), 2)
```

returns: TP_DEST

sepstring()

Purpose This function extracts a substring from the variable name that you specify. sepstring extracts all, from *n*th occurrence of the specified separator up to the end.

Syntax `sepstring(&(variableName), separator, nb_occurrence)`

Example If the Destination in the current Transfer is /run_time/etc/file.txt, the function:

```
sepstring(&(destination), '/', 3)
```

 returns: file.txt
 The function extracts all characters following the third separator /.
Note: The separator parameter must be enclosed in single quotes (').

uppercase()

Purpose This function converts the string that you specify to upper case characters.

Syntax `uppercase(&(variableName))`

Example If the Destination in the current Transfer is ftp_dest, the function:

```
uppercase(&(destination))
```

 returns: FTP_DEST

<code>lowercase()</code>	
Purpose	This function converts the string that you specify to lower case characters.
Syntax	<code>lowercase(&(variableName))</code>
Example	If the Destination in the current Transfer is FTP_DEST, the function: <pre>lowercase(&(destination))</pre> returns: ftp_dest

Case-insensitive conditions

You can define case-insensitive conditions in Decision Rules.

Example

You may want to reject any incoming files that are potentially dangerous. The type of file concerned is executable and of the type: *.bat, *.bin, *.cmd, *.dll, *.exe and so on. However, you cannot be sure if the file name is in lowercase, uppercase or mixed.

This example enables you to define a Decision Rule condition that selects all files with a name of the format *.exe, *.bin, *.bat, including: filename.EXE, filename.Exe and filename.eXe.

Create a Decision Rule condition with the following parameters:

Argument	Operator	Value
file_name	< >	*.[eE][xX][eE], *.[bB][iI][nN], *.[bB][aA][tT]

You can then call a script to delete these files.

[Overview: Event management](#)

[Working with Rule Tables and Decision Rules](#)

[Working with Decision Rules \(command line\)](#)

[Working with Rule Tables \(command line\)](#)

[Decision Rules and Rule Tables: Parameters List](#)

[Command syntax for Scheduling Event Models](#) (date syntax)

[Usage example: Routing](#)

[Usage example: Directory Scanning](#)

[Usage example: Scheduling](#)[User exits and Rule Tables](#)

Axway Gateway: Managing Events

New Rule Table: General tab

When you create a Rule Table, Gateway displays the **General** tab in the *New Rule Table* window.

Element	Description
Name	Enter a unique name for the new Rule Table.
Type	<i>You cannot modify this field.</i> This field displays the Rule Table type.
Comments	<i>Optional</i> Enter a free-text description for the Rule Table. Maximum: 80 alphanumeric characters.
Active	Select this option to set the Rule Table to <i>active</i> status.
Exclusive table	<i>Not for Transfer Change State Rule Tables.</i> By default, Gateway analyzes the Decision Rules in a Rule Table in their listed order, stopping at the first Rule that matches a complete set of conditions. • Select this option, if you want Gateway to stop the analysis of the table list after processing the first Rule that matches a complete set of conditions (default). • Disable this option if you want Gateway to continue analyzing the table list after processing a matching Rule. This enables you to execute more than one task on the occurrence of a single event.
Exit called	Some Rule Tables are inherently associated with exit processes that are automatically called during Rule processing. Select this option to disable all exits inherent in Rule processing.

Element	Description
Default execution type	The default execution type to apply to an event if no defined Decision Rule matches the current parameters. Select one of the following: • Model: Redirect the Transfer to another transfer whose parameters are defined in a Transfer Model. If you select this option, you must complete the Model name field. Additionally, it is recommended that you select the Link input and output transfers option. • XIB (<i>only for Scheduling-type Rule Tables and Transfer Change State Rule Tables</i>): Redirect the Transfer to AMTrix/ Axway Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to Purging for transfers routed from Gateway to Integrator. • To XMS: Redirect the Transfer to an Axway Messaging server queue via an Axway Messaging connector. • Mailbox cleaning (<i>only for Scheduling-type Rule Tables</i>): Clean the Mailbox via an associated Purge Model. • Perl Script: Apply a Perl script. • Batch command: Execute a batch command or script. • No action: Implement no routing.
Model name	This field is displayed if you select <i>Model</i> or <i>To XMS</i> in Default execution type. Enter the name of the Model to apply or select a Model from the drop-down list. The Model name must correspond to a General Model.
XMS connector out	This field is displayed if you select <i>To XMS</i> in Default execution type. Enter the name of the Axway Messaging out connector that handles the Transfer.
Link input and output transfers	<i>Not for all types of Rule Table.</i> This field is displayed if you select <i>Model</i> in Default execution type. Select this option to create a logical link between the incoming and the outgoing Transfers.

Element	Description
Purge Model	This field is displayed if you select <i>Mailbox cleaning</i> in Default execution type . Select the Purge Model you want to apply for the purge operation.
Perl script	This field is displayed if you select <i>Perl Script</i> in Default execution type . Enter the script to apply to the current Transfer.
Batch command	This field is displayed if you select <i>Batch command</i> in Default execution type . Enter the command to apply to the current Transfer. Alternatively, enter the path and file name of the script to execute. Refer to Executing scripts as command lines from Decision Rules .
Log level	Select one of the following: • No log: no log recording • Short: normal log recording • Full: detailed log recording

Axway Gateway: Managing Events

New Rule Table: General tab

When you create a Rule Table, Gateway displays the **General** tab in the *New Rule Table* window.

Element	Description
Name	Enter a unique name for the new Rule Table.
Type	<i>You cannot modify this field.</i> This field displays the Rule Table type.
Comments	<i>Optional</i> Enter a free-text description for the Rule Table. Maximum: 80 alphanumeric characters.
Active	Select this option to set the Rule Table to <i>active</i> status.

Element	Description
Exclusive table	<i>Not for Transfer Change State Rule Tables.</i> By default, Gateway analyzes the Decision Rules in a Rule Table in their listed order, stopping at the first Rule that matches a complete set of conditions. • Select this option, if you want Gateway to stop the analysis of the table list after processing the first Rule that matches a complete set of conditions (default). • Disable this option if you want Gateway to continue analyzing the table list after processing a matching Rule. This enables you to execute more than one task on the occurrence of a single event.
Exit called	Some Rule Tables are inherently associated with exit processes that are automatically called during Rule processing. Select this option to disable all exits inherent in Rule processing.
Default execution type	The default execution type to apply to an event if no defined Decision Rule matches the current parameters. Select one of the following: • Model: Redirect the Transfer to another transfer whose parameters are defined in a Transfer Model. If you select this option, you must complete the Model name field. Additionally, it is recommended that you select the Link input and output transfers option. • XIB (<i>only for Scheduling-type Rule Tables and Transfer Change State Rule Tables</i>): Redirect the Transfer to AMTriX/ Axway Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to Purging for transfers routed from Gateway to Integrator. • To XMS: Redirect the Transfer to an Axway Messaging server queue via an Axway Messaging connector. • Mailbox cleaning (<i>only for Scheduling-type Rule Tables</i>): Clean the Mailbox via an associated Purge Model. • Perl Script: Apply a Perl script. • Batch command: Execute a batch command or script. • No action: Implement no routing.

Element	Description
Model name	This field is displayed if you select <i>Model</i> or <i>To XMS</i> in Default execution type. Enter the name of the Model to apply or select a Model from the drop-down list. The Model name must correspond to a General Model.
XMS connector out	This field is displayed if you select <i>To XMS</i> in Default execution type. Enter the name of the Axway Messaging out connector that handles the Transfer.
Link input and output transfers	<i>Not for all types of Rule Table.</i> This field is displayed if you select <i>Model</i> in Default execution type. Select this option to create a logical link between the incoming and the outgoing Transfers.
Purge Model	This field is displayed if you select <i>Mailbox cleaning</i> in Default execution type. Select the Purge Model you want to apply for the purge operation.
Perl script	This field is displayed if you select <i>Perl Script</i> in Default execution type. Enter the script to apply to the current Transfer.
Batch command	This field is displayed if you select <i>Batch command</i> in Default execution type. Enter the command to apply to the current Transfer. Alternatively, enter the path and file name of the script to execute. Refer to Executing scripts as command lines from Decision Rules.
Log level	Select one of the following: • No log: no log recording • Short: normal log recording • Full: detailed log recording

New Rule Table: Linked rules tab

When you create any of the four types of Rule Table, Gateway displays the **Linked rules** tab in the *New Rule Table* window.

Use the **Linked rules** tab to add or remove [Decision Rules from the Rule Table](#), and to manage the order in which Gateway analyzes the Decision Rules.

Element	Description
Linked Rules	This frame contains the list of Decision Rules for inclusion in the Rule Table. Select the Decision Rules you want to include from the <i>Available Rules</i> frame.
View rule	This button appears when you select a Decision Rule in either the <i>Available Rules</i> frame or the <i>Linked Rules</i> frame. Click this button to view information about the selected rule.
Link	Select a Decision Rule in the <i>Available Rules</i> frame, then click the Link button. Gateway removes the Decision Rule from the <i>Available Rules</i> frame (linking it to the Rule Table), and places it in the <i>Linked Rules</i> frame. Note. Up to 749 Decision Rules for 64-bit Gateway Server installations, and up to 833 Decision Rules for 32-bit Gateway Server installations can be linked to a Rule Table. If more Decision Rules are linked to a single Rule Table, the product's functionality can be impacted – for example, Gateway Navigator could be disconnected with no indication that the limit was exceeded. Recommendation: Instead of linking more than the specified maximum number of Decision Rules, use more than one rule table. In this case, you must set the rule engine to allow evaluation spanning multiple tables, with the <code>[monitor] continue_ruletable_scan</code> configuration parameter.
Unlink	Select a Decision Rule in the <i>Linked Rules</i> frame, then click the Unlink button. Gateway removes the Decision Rule from the <i>Linked Rules</i> frame (unlinking it from the Rule Table), and places it in the <i>Available Rules</i> frame.

Element	Description
Up and down arrows	These buttons are active when you select a Decision Rule in the <i>Linked Rules</i> frame. Use these buttons to modify the position of the selected Decision Rule in the list. The list order determines the priority in which Gateway analyzes Decision Rules for conditional matching.
Available Rules	This frame displays all Decision Rules that are available for linking to the Rule Table. Click any Decision Rule listed in this frame to select it. Then use the View rule and/or Link buttons as described above.
Selected rule	This display bar displays the name of the Decision Rule you select in the <i>Linked Rules</i> or <i>Available Rules</i> frame.

Axway Gateway: Managing Events

New Rule Table: Scheduling Event tab

When you create or update a Scheduling-type Rule Table, Gateway displays the **Scheduling Event** tab in the *New Rule Table* window. The **Scheduling Event** tab, in turn, links you to the [New calendar](#) window.

Use the **Scheduling Event** tab to define one or more sets of dates that can trigger a Gateway action.

Scheduling Event tab

Use the **Scheduling Event** tab to specify the date(s) and time(s) a process is triggered.

Element	Description
Calendars	The Calendars frame contains a list of the scheduling calendars that have been created to trigger events. You create scheduling calendars in the New calendar window.
Dates	For any line (calendar) you select in the Calendars frame, the Dates frame displays all triggering dates for the selected calendar.
Hours	For any line (calendar) you select in the Calendars frame, the Hours frame displays all triggering times for the selected calendar.

Element	Description
New	Click New to open the New calendar window , enabling you to schedule an event time.
Update	Click Update to open the New calendar window and modify a selected calendar.
Delete	Select the name of a scheduling calendar in the Calendars frame, and then click Delete to remove the selected calendar.
Delay	In the event that your Gateway server is out of service during a period when events are scheduled, this section enables you to specify the handling of the un-triggered events on server restoration.
Type	Select one of the following values: • Until next delay (default) – when Gateway restarts, it executes the next scheduled action at the normal scheduled time, ignoring programmed actions that were not executed when the server was out of service. • Infinite delay – when Gateway restarts, it executes all actions that were scheduled but not executed due to server incapacity. • Special value – when, on restart, the delay value you specify in the Value field is exceeded, the programmed action is abandoned.
Value	This field is active when you select <i>Special value</i> in the Type field above. Enter a value to specify the maximum time within which a scheduled action can be executed on server restart. If this delay is exceeded, the programmed action is abandoned. Example: You schedule an action to be triggered at 12:30 p.m. on Friday and you enter a Special value delay of one hour. If the server is out of service from 12:00 p.m. to 1:00 p.m. on Friday, the action is triggered on server restart because restart occurs within the specified delay. However, if the server is out of service from 12:00 p.m. to 2:00 p.m. on Friday, the action is abandoned because the allowed delay value is exceeded.

New calendar window

When you click the **New** button or the **Update** button of the **Scheduling Event** tab, Gateway opens the *New calendar* window.

Use the *New calendar* window to create or modify the times and dates when a Gateway action is triggered.

Section	Description
Calendar pane	Use the calendars displayed in the Calendar pane to select dates on which events are triggered. - To display different months, click the left and right arrows situated to the left and right of the month names. - To select a date, click the date in any calendar. When you select the date, Gateway displays the date in the Dates pane. - To select additional dates, click on more dates. Gateway adds the additional selections to the display in the Dates pane. - To display the calendar of the month that contains today's date, click the Today button. - To remove a selected date, use the Delete button of the Dates pane.
Dates	The Dates pane displays the results of actions you perform in the Calendar pane and Selected Date sections of the New calendar window. Use the Dates pane to view and verify your scheduling data. - To add a line to the scheduling display, create a date in the Calendar pane and Selected Date section. - To remove a line from the scheduling display, click the line to select it and then click Delete. button situated immediately below the pane. - To modify a line of the scheduling display, select the line and then use the tools of the Selected Date section to change the line contents. - To associate one or more time of day with a date, click to select the date, and then use the tools of Selected Hour section to create time triggers. When you are satisfied with the scheduling data, click the OK button at the bottom of the New calendar window. Gateway closes the New calendar window and creates a new entry in the Calendars list of the Scheduling Event tab main window.

Section	Description
Selected Date	The Selected Date section contains elements that enable you to define and edit the date or series of dates that trigger a Gateway action. The Selected Date section contains the following elements: Selected Date data entry box: To work in this box, first click the radio button located immediately to the left of the box. You can: - Enter a date directly in this box. Be sure to respect the syntax for dates. For syntax information see Scheduling Event Model command syntax. - Enter a date by using the interactive calendars in the Calendar pane. After you enter a date via the a calendar, you can edit the date in the data entry box. - Enter a date by using the Date Syntax boxes, located below the Selected Date data entry box. When you use the Data Syntax boxes to create a date and then click Add or Apply, Gateway enters the data in the data entry box. - Edit existing dates by selecting an entry in the Selected Date pane and then editing the time in the Selected Date data entry box. Date Syntax boxes: The Date Syntax boxes help you define dates by providing you with a series of choices that ensure a syntactically correct date definition. To work with these boxes, first click the radio button located immediately to the left of the first box. When you begin creating a new date, only one Date Syntax box is visible. When you work with this box, more boxes appear depending on the date syntax you are creating. To create a date, follow this procedure: - Click the first Date Syntax box to display the drop-down list. From the list, select the beginning of a date expression. The list comprises the following choices: add remove keep and When you make a selection from the drop-down list,

Section	Description
	a second Date Syntax box appears to the right of the first box. - Click the second box to display the drop-down list. Choose a second element of your date definition expression from the list. When you make a selection from the drop-down list, a third Date Syntax box appears to the right of the second box. - Select elements from selection boxes as long as new boxes appear. When your date selection syntax is complete, the selected item in the last box terminates with a period, indicating the end of the expression. - Click Add to accept the date definition and add it to the Selected Date pane. Apply button: Click Apply to update a selected entry in the Selected Date pane after you edit it in the Selected Date data entry box. Add button: Click Add to add the time currently displayed in the Selected Date entry box to the Selected Date pane. Clear button: Click clear to abandon the creation of a date selection expression and clear all data from the Date Syntax boxes.
Hours (applied to all dates)	The Hours pane displays the results of actions you perform in the Hour section of the <i>New calendar</i> window. Use the Hours pane to view and verify your triggering time data. You can associate multiple triggering times with a single entry in the Dates pane. - To add a triggering time to the Hours display pane, create a time trigger in the Selected Hour section. The triggering times you create apply to every date in the Selected Date pane. - To remove a triggering time from the Hours display pane, click the line to select it, then click the Delete button situated immediately below the pane. - To modify a triggering time in the Hours display pane, click the line to select it, then use the tools of the Selected Hour section to change the line contents.

Section	Description
Selected Hour	The Selected Hour section contains elements that enable you to define a time of day when Gateway triggers an action. The triggering times you define in this section apply to every date in the Selected Date pane. The Selected Hour section contains the following elements: • Selected Hour data entry box: You can: ◦ Enter trigger times directly in this box. Be sure to respect the syntax for time data. Refer to the syntax examples in the Examples of hour syntax section. For additional syntax information see Scheduling Event Model command syntax. ◦ Enter times by using the Time-selection bar. ◦ Edit existing trigger times by selecting an entry in the Selected Hour pane and then editing the time in the Selected Hour data entry box. • Apply button: Click Apply to update a selected entry in the Selected Hour pane after you edit it in the Selected Hour data entry box. • Add button: Click Add to add the time currently displayed in the Selected Hour data entry box to the Selected Hour pane.
Time-selection bar	Use the <i>Time-selection bar</i> to enter triggering times in the Selected Hour data entry box of the Selected Hour section. To select a time of day that triggers an event, double-click anywhere on the Time selection bar. A pointer appears on the bar. Now click-drag the pointer on the bar to the time of your choice. You can choose an hour and minute, with a granularity limit of 5 minutes. For example, using the selection bar you can choose 11:35 or 11:40 but not minutes between these times. For greater precision, choose a time using the selection bar, then edit the result directly in the Selected Hour data entry box.
Examples of hour syntax	This pane displays examples of the correct syntax for time trigger entries. Refer to Scheduling Event Model command syntax.
OK	After creating date and time combinations that you want to use as triggering events, select OK to validate. Gateway closes the New calendar window and creates a new entry in the Calendars list of the Scheduling Event tab main window.

Section	Description
Cancel	To abandon the creation of a triggering calendar, click Cancel . Gateway closes the New calendar window and without changing the Scheduling Event tab.

Axway Gateway: Managing Events

New Rule Table: File Event tab

When you create a Directory Scanning type Rule Table, Gateway displays the **File Event** tab in the *New Rule Table* window.

Use the **File Event** tab to specify the file that Gateway supervises for file triggering events.

Element	Description
Watch directory	Enter the path to the directory you want Gateway to supervise (or poll) to determine the presence of a file.
Watch period	Enter the time cycle (in seconds) in which you want polling to reoccur.
Work directory	Enter the path to the work directory. When Gateway polls the Watch Directory and locates a file, it transfers the file to the Work directory before beginning file processing.
Optimized cleaning	Select this option to enable Gateway to periodically delete a part of the contents of the Work Directory contents. The Optimized cleaning function includes security controls to avoid deletion of critical files. However, in cases of extremely sensitive files, it is prudent to avoid activating this option.

Axway Gateway: Managing Events

New Rule Table: Event tab

When you create a Transfer Change State type Rule Table, Gateway displays the **Event** tab in the *New Rule Table* window.

Use the **Event** tab to restrict processing to a limited set of Transfers, depending on conditions you specify.

Element	Description
Restrict processing	Select this check box if you want to create restriction criteria for the application of transfer Decision Rules. When you select this box, Gateway activates the criteria text boxes below.
Transfer state	Enter one or more transfer states for which you want to allow processing. For example: ENDED, ACKED, ...
Transfer mode	Enter the mode for which you want to allow processing: • Responder or • Initiator
Transfer direction	Enter the Transfer direction for which you want to allow processing: • IN or • OUT
Transfer type	Enter one or more transfer types for which you want to allow processing. For example: MSG, TRANS, ...
Transfer protocol	Enter one or more transfer protocols for which you want to allow processing. For example: PEL, ODETTE, PHSE...

Note Multiple values for a condition must be separated by a comma with no space between them.

Axway Gateway: Managing Events

New Rule Table: XMS Event tab

When you create an XMS-type Rule Table, Gateway displays the **XMS Event** tab in the *New Rule Table* window.

Use the **XMS Event** tab to associate a Property List with the Rule Table.

Element	Description
XMS connector	<i>You cannot modify this field.</i> Displays the Axway Messaging connector for which you are creating a Rule Table.
Property list	Enter the name of an existing Property List that you want to associate with the Rule Table. Alternatively, select a Property List from the drop-down list.

Axway Gateway: Managing Events

Working with Decision Rules (command line)

[`peladm create decisionrule`](#)

[`peladm update decisionrule`](#)

[`peladm delete decisionrule`](#)

[`peldsp display decisionrule`](#)

[`peldsp select decisionrule`](#)

Creating a Decision Rule: `peladm create_decisionrule`

Syntax	<code>peladm create_decisionrule</code> [see parameters below]
Description	Use this command to create a new Decision Rule. Specify at least one condition for the Decision Rule. You can specify conditions in two ways: - Use the parameter <code>cond_arg (-ca)</code> to list a set of conditions separated by the semi-colon character. - Use the parameter <code>-import_cond (-ic)</code> to specify a file to import, that contains a list of conditions. Each line in the file contains an argument name, an operator and a value.

Parameters	<code>name (-n)</code> <i>(Mandatory)</i>	Enter a unique name for the Decision Rule.
	<code>type_of_decisionrules</code> <code>(-type)</code>	Enter one of the following Decision Rule types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS
	<code>comments (-cts)</code>	Enter a free text description of the Decision Rule you are creating.

`execution_type`
(-et)

The type of process to be triggered by a Decision Rule. Enter one of the following values:

- **MODEL** = the execution is defined by a Model. Specify the Model in the `model (-ml)` parameter below.
- **AMTRIX** = redirect the Transfer to AMTrix for processing
- **PERL** Refer to the readme in the following folder:
 - On UNIX <Gateway installation directory>/run_time/scripts/perl/
 - On Windows <Gateway installation directory>\run_time\scripts\perl\
- **COMMAND** = the execution type is a command line or script specified in the `command_line` parameter. Refer to [Executing scripts as command lines from Decision Rules](#).
- **TO_XMS**
- **NONE** (default)
- **LOG_CLEANING**
- **MAILBOX_CLEANING**
- **STAT_CLEANING**
- **XIB** = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).
- **DEFAULT** = route the transfer using a Model inside an Application. This option is only

- **MODEL** = the execution is defined by a Model. Specify the Model in the `model (-ml)` parameter below.
- **AMTRIX** = redirect the Transfer to AMTriX for processing
- **PERL** Refer to the readme in the following folder:
 - On UNIX
`<Gateway installation director y>/run_time/scripts/perl/`
 - On Windows
`<Gateway installation director y>\run_time\scripts\perl\`
- **COMMAND** = the execution type is a command line or script specified in the `command_line`

parameter.
Refer to
[Executing
scripts as
command lines
from Decision
Rules.](#)

- **TO_XMS**
- **NONE** (default)
- **LOG_**
CLEANING
- **MAILBOX_**
CLEANING
- **STAT_**
CLEANING
- **XIB** = redirect
the Transfer to
Integrator for
processing. The
payload file for a
Transfer which
was routed to
Integrator is not
deleted by the
purge operation.
For more
information,
refer to [Purging
for transfers
routed from
Gateway to
Integrator.](#)
- **DEFAULT** =
route the
transfer using a
Model inside an
Application. This
option is only
available via
online
commands. The
option exists for
reasons of

	compatibility with earlier product versions.
<code>link_ transfers (-lt)</code>	Use this parameter if you enter the value <i>MODEL</i> for the <code>execution_type</code> parameter. Enter one of the following values: • Y (default) = Yes - create a logical link between the incoming and the outgoing Transfers • N = No - no link between the incoming and the outgoing Transfers
<code>model (-ml)</code>	If you entered the value <i>MODEL</i> or <i>TO_ XMS</i> for the <code>execution_type</code> parameter, enter a Model name.
<code>purge_model (-pm)</code>	If you entered the value <i>MAILBOX_ CLEANING</i> for the <code>execution_type</code> parameter, enter the name of an existing Purge Model to define the purge operation on the Mailbox.

<code>perl_script (-ps)</code>	If you entered the value <i>PERL</i> for the <code>execution_type</code> parameter, enter the Perl script name and access path.
<code>command_ line (-cl)</code>	If you entered the value <i>COMMAND</i> for the <code>execution_type</code> parameter, enter the execution command line. The command line can include symbolic variables. Alternatively, you can enter the path and file name of a script to execute. Refer to Executing scripts as command lines from Decision Rules.
<code>cond_arg (- ca)</code>	Enter one or more conditional arguments in the format <i>condition1;condition 2</i> For example: <pre>cond "destination = FTPC; direction = I, protocol<>PHSE, FTP; appli = & (file_name) "</pre>
<code>prop_list_ name (-pln)</code>	<i>Only for JMS or Axway Messaging</i> Enter a Property List name.

xms_ connector_ out_name (- xcn)	Use this parameter if you are creating an XMS-type Decision Rule. Enter an Axway Messaging connector name.

import_cond (-ic)	Enter the name of the file that contains the execution conditions.

Usage rules**Execution type parameters:**

If you set `execution_type` to:

- *COMMAND*, then enter a value for the `command_line` parameter. You can enter a command or the path and file name of a script. Refer to [Executing scripts as command lines from Decision Rules](#).
 - *MODEL*, then enter a value for the `model` parameter.
 - *PERL*, then enter a value for the `perl_script` parameter.
-

Example 1

The following example creates a Decision Rule to handle incoming files from the FTP client FTPC.

```
peladm create_decisionrule
  -name TABFTP
  -type XFER_CHANGE_STATE
  -execution_type Model
  -ml MODELFTP
  -cond_arg "destination = FTPC; direction =
I"
  -protocol<>PHSE,FTP; appli = &(file_name) "
```

Gateway checks the following conditions:

- Destination = FTPC
- Direction = Input
- Protocol = either PeSIT HS E or FTP
- Application = file name of the current Transfer

If all conditions correspond to the current Transfer parameters, Gateway submits a new Transfer Request and applies the Model MODELFTP. This Model contains the parameters required to route the Transfer.

Example 2

The following example creates a Decision Rule for routing to a PeSIT client:

```
peladm create_decisionrule
  -name TABPHSE
  -type XFER_CHANGE_STATE
  -execution_type DEFAULT
  -cond_arg "destination = *PHSE; rec_
len>=1200; originator #FTPC"
```

Gateway checks the following conditions:

- Destination = *PHSE (a destination Site whose name ends with PHSE)
- Record length = greater than or equal to 1200
- Originator = any originator Site that is not FTTPC

If all the conditions correspond to the current Transfer parameters, Gateway submits a new Transfer Request by using the execution type *DEFAULT*.

Updating Decision Rules: peladm update_decisionrule

Syntax

`peladm update_decisionrule` [see parameters below]

Description

Use this command to modify an existing Decision Rule.

You can:

- Add a new condition to a Decision Rule
 - Modify an existing condition
 - Delete an existing condition
 - Modify the name of the condition import file
 - Modify the rule type
 - Link or un-link Transfers incoming and outgoing Transfers
 - Modify the routing behavior for this Rule
 - Modify the Model referenced in this Rule
 - Modify the Perl script referenced in this Rule
 - Modify the command line referenced in this Rule
-

Parameters	<code>name (-n)</code> <i>(Mandatory)</i>	Enter a unique name for the Decision Rule.
	<code>type_of_decisionrules (-type)</code>	Enter one of the following Decision Rule types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS
	<code>comments (-cts)</code>	Edit the free text description of the Decision Rule.

`execution_type`
(-et)

The type of process to be triggered by a Decision Rule. Enter one of the following values:

- **MODEL** = the execution is defined by a Model. Specify the Model in the `model` (-ml) parameter below.
 - **AMTRIX** = redirect the Transfer to AMTrix for processing
 - **PERL**
 - **COMMAND** = the execution type is a command line or script specified in the `command_line` parameter. Refer to [Executing scripts as command lines from Decision Rules](#).
 - **TO_XMS**
 - **NONE** (default)
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).
 - **DEFAULT** = route the transfer using a Model inside an Application. This option is only available via online commands. The option exists for reasons of compatibility with earlier product versions.
-

link_transfers (-lt)	Use this parameter if you enter the value <i>MODEL</i> for the <code>execution_type</code> parameter. Enter one of the following values: • Y (default) = Yes - create a logical link between the incoming and the outgoing Transfers • N = No - no link between the incoming and the outgoing Transfers
model (-ml)	If you entered the value <i>MODEL</i> or <i>TO_XMS</i> for the <code>execution_type</code> parameter, enter a Model name.
purge_model (-pm)	If you entered the value <i>MAILBOX_CLEANING</i> for the <code>execution_type</code> parameter, enter the name of an existing Purge Model to define the purge operation on the Mailbox.
perl_script (-ps)	If you entered the value <i>PERL</i> for the <code>execution_type</code> parameter, enter the Perl script name and access path.
command_line (-cl)	If you entered the value <i>COMMAND</i> for the <code>execution_type</code> parameter, enter the execution command line.
prop_list_name (-pln)	<i>Only for JMS or Axway Messaging</i> Enter a Property List name.
xms_connector_ out_name (-xcn)	Use this parameter if you are creating an XMS-type Decision Rule. Enter an Axway Messaging connector name.
add_cond (-a)	Add one or more conditions.
update_cond (-u)	Update on or more existing conditions.
delete_cond (-d)	Specify a condition to delete.
import_cond (-ic)	Specify a file the file to import that contains the execution conditions.

Usage rules**Execution type parameters:**

If you set `execution_type` to:

- *COMMAND*, then enter a value for the `command_line` parameter.
- *MODEL*, then enter a value for the `model` parameter.
- *PERL*, then enter a value for the `perl_script` parameter.

Syntax**Examples**

Adding a condition:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-add_cond ['""argument_name operator
value""';']*]
```

Modifying a condition:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-update_cond ['""argument_name operator
value""';']*]
```

Deleting a condition:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-update_cond ['""argument_name operator
value""';']*]
```

Modifying routing behavior:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-execution_type { MODEL | AMTRIX | PERL
| COMMAND |
    TO_XMS | DEFAULT | NONE | LOG_CLEANING |
    MAILBOX_CLEANING | STAT_CLEANING | XIB
}]
```

Modifying the Model:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-model model name]
```

Modifying the Perl script:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-perl_script script access path]
```

Modifying the online command:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-command_line online command]
```

Example The following example updates the Decision Rule *MyRule* and sets the processing type to use the Model *MyModel*:

```
peladm update_decisionrule
    -name MyRule
    -execution_type Model
    -model MyModel
```

Deleting a Decision Rule: `peladm delete_decisionrule`

Syntax `peladm delete_decisionrule [-name] [-type_of_decisionrules]`

Description Use this command to delete an existing Decision Rule. Repeat the command for each Decision Rule you want to delete.

Parameters `-name (-n)` Enter the name of the Decision Rule you want to delete.

`-type_of_decisionrules (-type)` Enter the type of Decision Rule you want to delete. Enter one of the following values:

- **XFER_CHANGE_STATE**
- **TEMPORAL**
- **FILE_STORING**
- **XMS**

Notes If the `peladm delete_decisionrule` command produces an error with a message like this:

The decision rule that wants to be deleted (peladm delete_decisionrule -n MY_DECISION_RULE -type MY_TYPE) is linked to a Rule Table (MY_RULETABLE), therefore it cannot be deleted.

In this case, the solution is to first unlink the decision rule from the MY_RULETABLE table:

```
peladm update_ruletable -n MY_RULETABLE -
type MY_TYPE -d MY_DECISION_RULE
```

Example Enter the command:

```
peladm delete_decisionrule -n
MyDecisionRule
```

Gateway deletes the Decision Rule with the name *MyDecisionRule*.

Displaying Decision Rule properties: `peldsp display_decisionrule`

Syntax	<code>peldsp display_decisionrule [-name] [-type_of_decisionrules]</code>
Description	Use this command to display the properties of a specified Decision Rule.
Parameters	<code>-name (-n)</code>: Enter the name of the Decision Rule whose properties you want to display <code>-type_of_decision_rules (-t)</code>: Enter the type of Decision Rule whose properties you want to display. Enter one of the following values: • XFER_CHANGE_STATE • TEMPORAL • FILE_STORING • XMS
Example	Enter the command: <pre>peldsp display_decisionrule -n MyDecisionRule</pre> Gateway returns the properties of the Decision Rule with the name <i>MyDecisionRule</i>: <pre>rd_name MyDecisionRule rd_type 'XFER_CHANGE_STATE' rd_comments "" rd_execution_type = MODEL rd_link_transfers = Y rd_perl_script = rd_cond_1 = protocol<> FTP, PHSE</pre>

Displaying all Decision Rule names: `peldsp select_decisionrule`

Syntax	<code>peldsp select_decisionrule</code>
Description	Use this command to display the names of all Decision Rules.

Parameters	Enter the command with no parameters to display the names of all Decision Rules. To list Decision Rules according to a given parameter, enter the command followed by the parameter, for example <code>-name</code> or <code>-type TEMPORAL</code>.
Example	Enter the command: <pre>peldsp select_decisionrule</pre> Gateway returns the names of all Decision Rules existing on the server.

Axway Gateway: Managing Events

Working with Decision Rules (command line)

[`peladm create\_decisionrule`](#)

[`peladm update\_decisionrule`](#)

[`peladm delete\_decisionrule`](#)

[`peldsp display\_decisionrule`](#)

[`peldsp select\_decisionrule`](#)

Creating a Decision Rule: `peladm create_decisionrule`

Syntax	<code>peladm create_decisionrule</code> [see parameters below]
Description	Use this command to create a new Decision Rule. Specify at least one condition for the Decision Rule. You can specify conditions in two ways: • Use the parameter <code>cond_arg</code> (<code>-ca</code>) to list a set of conditions separated by the semi-colon character. • Use the parameter <code>-import_cond</code> (<code>-ic</code>) to specify a file to import, that contains a list of conditions. Each line in the file contains an argument name, an operator and a value.

Parameters	<code>name (-n)</code> <i>(Mandatory)</i>	Enter a unique name for the Decision Rule.
	<code>type_of_decisionrules</code> <code>(-type)</code>	Enter one of the following Decision Rule types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS
	<code>comments (-cts)</code>	Enter a free text description of the Decision Rule you are creating.

`execution_type`
(-et)

The type of process to be triggered by a Decision Rule. Enter one of the following values:

- **MODEL** = the execution is defined by a Model. Specify the Model in the `model (-ml)` parameter below.
 - **AMTRIX** = redirect the Transfer to AMTrix for processing
 - **PERL** Refer to the readme in the following folder:
 - On UNIX <Gateway installation directory>/run_time/scripts/perl/
 - On Windows <Gateway installation directory>\run_time\scripts\perl\
 - **COMMAND** = the execution type is a command line or script specified in the `command_line` parameter. Refer to [Executing scripts as command lines from Decision Rules](#).
 - **TO_XMS**
 - **NONE** (default)
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).
 - **DEFAULT** = route the transfer using a Model inside an Application. This option is only
-

- **MODEL** = the execution is defined by a Model. Specify the Model in the `model (-ml)` parameter below.
- **AMTRIX** = redirect the Transfer to AMTriX for processing
- **PERL** Refer to the readme in the following folder:
 - On UNIX
`<Gateway installation director y>/run_time/scripts/perl/`
 - On Windows
`<Gateway installation director y>\run_time\scripts\perl\`
- **COMMAND** = the execution type is a command line or script specified in the `command_line`

parameter.
Refer to
[Executing
scripts as
command lines
from Decision
Rules.](#)

- **TO_XMS**
- **NONE** (default)
- **LOG_**
CLEANING
- **MAILBOX_**
CLEANING
- **STAT_**
CLEANING
- **XIB** = redirect
the Transfer to
Integrator for
processing. The
payload file for a
Transfer which
was routed to
Integrator is not
deleted by the
purge operation.
For more
information,
refer to [Purging
for transfers
routed from
Gateway to
Integrator.](#)
- **DEFAULT** =
route the
transfer using a
Model inside an
Application. This
option is only
available via
online
commands. The
option exists for
reasons of

	compatibility with earlier product versions.
<code>link_ transfers (-lt)</code>	Use this parameter if you enter the value <i>MODEL</i> for the <code>execution_type</code> parameter. Enter one of the following values: • Y (default) = Yes - create a logical link between the incoming and the outgoing Transfers • N = No - no link between the incoming and the outgoing Transfers
<code>model (-ml)</code>	If you entered the value <i>MODEL</i> or <i>TO_ XMS</i> for the <code>execution_type</code> parameter, enter a Model name.
<code>purge_model (-pm)</code>	If you entered the value <i>MAILBOX_ CLEANING</i> for the <code>execution_type</code> parameter, enter the name of an existing Purge Model to define the purge operation on the Mailbox.

<code>perl_script (-ps)</code>	If you entered the value <i>PERL</i> for the <code>execution_type</code> parameter, enter the Perl script name and access path.

<code>command_ line (-cl)</code>	If you entered the value <i>COMMAND</i> for the <code>execution_type</code> parameter, enter the execution command line. The command line can include symbolic variables. Alternatively, you can enter the path and file name of a script to execute. Refer to Executing scripts as command lines from Decision Rules .

<code>cond_arg (- ca)</code>	Enter one or more conditional arguments in the format <i>condition1;condition 2</i> For example: <code>cond "destination = FTPC; direction = I, protocol<>PHSE, FTP; appli = & (file_name) "</code>

<code>prop_list_ name (-pln)</code>	<i>Only for JMS or Axway Messaging</i> Enter a Property List name.

xms_ connector_ out_name (- xcn)	Use this parameter if you are creating an XMS-type Decision Rule. Enter an Axway Messaging connector name.

import_cond (-ic)	Enter the name of the file that contains the execution conditions.

Usage rules**Execution type parameters:**

If you set `execution_type` to:

- *COMMAND*, then enter a value for the `command_line` parameter. You can enter a command or the path and file name of a script. Refer to [Executing scripts as command lines from Decision Rules](#).
 - *MODEL*, then enter a value for the `model` parameter.
 - *PERL*, then enter a value for the `perl_script` parameter.
-

Example 1

The following example creates a Decision Rule to handle incoming files from the FTP client FTPC.

```
peladm create_decisionrule
  -name TABFTP
  -type XFER_CHANGE_STATE
  -execution_type Model
  -ml MODELFTP
  -cond_arg "destination = FTPC; direction =
I"
  -protocol<>PHSE,FTP; appli = &(file_name)"
```

Gateway checks the following conditions:

- Destination = FTPC
- Direction = Input
- Protocol = either PeSIT HS E or FTP
- Application = file name of the current Transfer

If all conditions correspond to the current Transfer parameters, Gateway submits a new Transfer Request and applies the Model MODELFTP. This Model contains the parameters required to route the Transfer.

Example 2

The following example creates a Decision Rule for routing to a PeSIT client:

```
peladm create_decisionrule
  -name TABPHSE
  -type XFER_CHANGE_STATE
  -execution_type DEFAULT
  -cond_arg "destination = *PHSE; rec_
len>=1200; originator #FTPC"
```

Gateway checks the following conditions:

- Destination = *PHSE (a destination Site whose name ends with PHSE)
- Record length = greater than or equal to 1200
- Originator = any originator Site that is not FTPC

If all the conditions correspond to the current Transfer parameters, Gateway submits a new Transfer Request by using the execution type *DEFAULT*.

Updating Decision Rules: peladm update_decisionrule

Syntax

`peladm update_decisionrule` [see parameters below]

Description

Use this command to modify an existing Decision Rule.

You can:

- Add a new condition to a Decision Rule
 - Modify an existing condition
 - Delete an existing condition
 - Modify the name of the condition import file
 - Modify the rule type
 - Link or un-link Transfers incoming and outgoing Transfers
 - Modify the routing behavior for this Rule
 - Modify the Model referenced in this Rule
 - Modify the Perl script referenced in this Rule
 - Modify the command line referenced in this Rule
-

Parameters	<code>name (-n)</code> <i>(Mandatory)</i>	Enter a unique name for the Decision Rule.
	<code>type_of_decisionrules (-type)</code>	Enter one of the following Decision Rule types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS
	<code>comments (-cts)</code>	Edit the free text description of the Decision Rule.

`execution_type`
(-et)

The type of process to be triggered by a Decision Rule. Enter one of the following values:

- **MODEL** = the execution is defined by a Model. Specify the Model in the `model` (-ml) parameter below.
 - **AMTRIX** = redirect the Transfer to AMTrix for processing
 - **PERL**
 - **COMMAND** = the execution type is a command line or script specified in the `command_line` parameter. Refer to [Executing scripts as command lines from Decision Rules](#).
 - **TO_XMS**
 - **NONE** (default)
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).
 - **DEFAULT** = route the transfer using a Model inside an Application. This option is only available via online commands. The option exists for reasons of compatibility with earlier product versions.
-

link_transfers (-lt)	Use this parameter if you enter the value <i>MODEL</i> for the <code>execution_type</code> parameter. Enter one of the following values: • Y (default) = Yes - create a logical link between the incoming and the outgoing Transfers • N = No - no link between the incoming and the outgoing Transfers
model (-ml)	If you entered the value <i>MODEL</i> or <i>TO_XMS</i> for the <code>execution_type</code> parameter, enter a Model name.
purge_model (-pm)	If you entered the value <i>MAILBOX_CLEANING</i> for the <code>execution_type</code> parameter, enter the name of an existing Purge Model to define the purge operation on the Mailbox.
perl_script (-ps)	If you entered the value <i>PERL</i> for the <code>execution_type</code> parameter, enter the Perl script name and access path.
command_line (-cl)	If you entered the value <i>COMMAND</i> for the <code>execution_type</code> parameter, enter the execution command line.
prop_list_name (-pln)	<i>Only for JMS or Axway Messaging</i> Enter a Property List name.
xms_connector_ out_name (-xcn)	Use this parameter if you are creating an XMS-type Decision Rule. Enter an Axway Messaging connector name.
add_cond (-a)	Add one or more conditions.
update_cond (-u)	Update on or more existing conditions.
delete_cond (-d)	Specify a condition to delete.
import_cond (-ic)	Specify a file the file to import that contains the execution conditions.

Usage rules**Execution type parameters:**

If you set `execution_type` to:

- *COMMAND*, then enter a value for the `command_line` parameter.
- *MODEL*, then enter a value for the `model` parameter.
- *PERL*, then enter a value for the `perl_script` parameter.

Syntax**Examples**

Adding a condition:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-add_cond ['""argument_name operator
value""';']*]
```

Modifying a condition:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-update_cond ['""argument_name operator
value""';']*]
```

Deleting a condition:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-update_cond ['""argument_name operator
value""';']*]
```

Modifying routing behavior:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-execution_type { MODEL | AMTRIX | PERL
| COMMAND |
    TO_XMS | DEFAULT | NONE | LOG_CLEANING |
    MAILBOX_CLEANING | STAT_CLEANING | XIB
}]
```

Modifying the Model:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-model model name]
```

Modifying the Perl script:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-perl_script script access path]
```

Modifying the online command:

```
peladm update_decisionrule
    {-name <Decision rule name>}
    [-command_line online command]
```

Example

The following example updates the Decision Rule *MyRule* and sets the processing type to use the Model *MyModel*:

```
peladm update_decisionrule
    -name MyRule
    -execution_type Model
    -model MyModel
```

Deleting a Decision Rule: *peladm delete_decisionrule*

Syntax

```
peladm delete_decisionrule [-name] [-type_of_
decisionrules]
```

Description

Use this command to delete an existing Decision Rule. Repeat the command for each Decision Rule you want to delete.

Parameters

<code>-name (-n)</code>	Enter the name of the Decision Rule you want to delete.
<code>-type_of_ decisionrules (-type)</code>	Enter the type of Decision Rule you want to delete. Enter one of the following values: • XFER_CHANGE_STATE • TEMPORAL • FILE_STORING • XMS

Notes

If the `peladm delete_decisionrule` command produces an error with a message like this:

The decision rule that wants to be deleted (peladm delete_decisionrule -n MY_DECISION_RULE -type MY_TYPE) is linked to a Rule Table (MY_RULETABLE), therefore it cannot be deleted.

In this case, the solution is to first unlink the decision rule from the MY_RULETABLE table:

```
peladm update_ruletable -n MY_RULETABLE -
type MY_TYPE -d MY_DECISION_RULE
```

Example

Enter the command:

```
peladm delete_decisionrule -n
MyDecisionRule
```

Gateway deletes the Decision Rule with the name *MyDecisionRule*.

Displaying Decision Rule properties: *peldsp display_decisionrule*

Syntax	<code>peldsp display_decisionrule [-name] [-type_of_decisionrules]</code>
Description	Use this command to display the properties of a specified Decision Rule.
Parameters	<code>-name (-n)</code>: Enter the name of the Decision Rule whose properties you want to display <code>-type_of_decision_rules (-t)</code>: Enter the type of Decision Rule whose properties you want to display. Enter one of the following values: • XFER_CHANGE_STATE • TEMPORAL • FILE_STORING • XMS
Example	Enter the command: <pre>peldsp display_decisionrule -n MyDecisionRule</pre> Gateway returns the properties of the Decision Rule with the name <i>MyDecisionRule</i>: <pre>rd_name MyDecisionRule rd_type 'XFER_CHANGE_STATE' rd_comments "" rd_execution_type = MODEL rd_link_transfers = Y rd_perl_script = rd_cond_1 = protocol<> FTP, PHSE</pre>

Displaying all Decision Rule names: *peldsp select_decisionrule*

Syntax	<code>peldsp select_decisionrule</code>
Description	Use this command to display the names of all Decision Rules.

Parameters	Enter the command with no parameters to display the names of all Decision Rules. To list Decision Rules according to a given parameter, enter the command followed by the parameter, for example <code>-name</code> or <code>-type TEMPORAL</code> .

Example	Enter the command: <code>peldsp select_decisionrule</code> Gateway returns the names of all Decision Rules existing on the server.

Axway Gateway: Managing Events

Working with Rule Tables (command line)

[`peladm create ruletable`](#)[`peladm update ruletable`](#)[`peladm delete ruletable`](#)[`peldsp display ruletable`](#)[`peldsp select ruletable`](#)

Creating a Rule Table: `peladm create_ruletable`

Syntax	<code>peladm create_ruletable</code> [see parameters below]

Description	Use this command to create a new Rule Table.

Parameters	<i>Mandatory parameter</i>	
	<code>-name (-n)</code>	Enter a unique name for the Rule Table you are creating.
	<code>-status (-stat)</code>	Enter one of the following states: • ACTIVE (default) • INACTIVE
	<code>-type_of_rules_table (-type)</code>	Enter one of the following Rule Table types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS

```
-default_execution_  
type (-det)
```

The type of process to be triggered if no process is triggered by an associated Decision Rule. Enter one of the following values:

- **MODEL** = the default execution is defined by a Model. Specify the Model in the `default_model (-dm)` parameter below.
 - **AMTRIX** = redirect the Transfer to AMTrix for processing
 - **PERL**
 - **COMMAND** = the default execution is a command line or script specified in the `default_command_line` parameter. Refer to [Executing scripts as command lines from Decision Rules](#).
 - **TO_XMS**
 - **NONE** (default)
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).
 - **DEFAULT** = route the transfer using a Model inside an Application. This option is only available via online commands. The option exists for reasons of compatibility with earlier product versions.
-

<code>-link_transfers (-lt)</code>	Use this parameter if you enter the value <i>MODEL</i> for the <code>default_execution_type</code> parameter. Enter one of the following values: • Y (default) = Yes. Create a logical link between the incoming and the outgoing Transfers. • N = No. No link between the incoming and the outgoing Transfers When you enter Y: • The input Transfer Request parameter <code>route_to_xfer</code> is set to the output Transfer identifier • The output Transfer Request parameter <code>route_from_xfer</code> is set to the input Transfer identifier
<code>-default_model (-dm)</code>	If you entered the value <i>MODEL</i> for the <code>default_execution_type</code> parameter, enter a Model name.
<code>-default_purge_model (-dpm)</code>	If you entered the value <i>MAILBOX_CLEANING</i> for the <code>default_execution_type</code> parameter, enter the name of an existing Purge Model to define the purge operation on the Mailbox.
<code>-default_perl_script (-dps)</code>	If you entered the value <i>PERL</i> for the <code>default_execution_type</code> parameter, enter the Perl script name and access path.

<code>-default_command_line (-dcl)</code>	If you entered the value COMMAND for the <code>default_execution_type</code> parameter, enter the execution command line, or alternatively, enter the path and file name to a script. Refer to Executing scripts as command lines from Decision Rules.
<code>-add_decisionrule (-a)</code>	Add one or more Decision Rules.
<code>-restrict_processing (-rp)</code>	Select whether or not to activate restriction criteria parameters for Decision Rules processing. Enter one of the following values: • Y = Yes • N (default) = No If you enter the Y (yes) value, you must also enter at least one value for one of the following parameters: • <code>restrict_processing_on_state</code> • <code>restrict_processing_on_direction</code> • <code>restrict_processing_on_type</code> • <code>restrict_processing_on_mode</code> • <code>restrict_processing_on_protocol</code>
<code>-restrict_processing_on_state (-rpos)</code>	You can enter one or more of the following values if you entered the value Y (yes) for the <code>restrict_processing (-rp)</code> parameter: • ENDED • CANCELED • ACKNOWLEDGED

`-restrict_
processing_on_
direction (-rpod)`

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **IN**
- **OUT**

`-restrict_
processing_on_type
(-rpot)`

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **TRANS**
- **LOTS**
- **POLL**
- **LIST**
- **EERP**
- **SELECT**
- **MSG**
- **DIR**

`-restrict_
processing_on_mode
(-rpom)`

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **INITIATOR**
 - **RESPONDER**
-

`-restrict_
processing_on_
protocol (-rpop)`

You can enter one or more of the following values if you entered the value Y (yes) for the `restrict_processing (-rp)` parameter:

- **PEL**
- **ODETTE**
- **PHSE**
- **PHSD**
- **FTP**
- **HTTP**
- **FTP_HTTP**
- **POP3**
- **SFTP**
- **SMTP**
- **AS2**
- **AS3**
- **AS1_POP3**
- **AS1_SMTP**
- **RN_HTTP**
- **RN_POP3**
- **RN_SMTP**
- **SWIFTNET**
- **JMS**
- **X420**
- **X400**

`-log_level (-ll)`

To define the logging level, enter one of the following values:

- **DEFAULT_LEVEL** (default)
 - **0** = none
 - **1** = short
 - **2** = full
-

```
-is_rules_table_  
exclusive (-excl)
```

By default, Gateway analyzes the Decision Rules in a Rule Table in their listed order, stopping at the first Rule that matches a complete set of conditions.

Enter one of the following values:

- **Y** = (default)
- **N** = No - enables Gateway to continue analyzing the table list after processing a matching Rule

```
-is_user_exits_  
activated (-exit)
```

Rule Tables are inherently associated with exit processes that are automatically called during Rule processing.

Enter one of the following values:

- **Y** (default) = Yes - enable all inherent exits
- **N** = No - disable all inherent exits

```
-temporal_event_  
model (-tem)
```

If you are creating a Scheduling-type Rule Table (`-type = TEMPORAL`), and you set the `default_execution_type` to *MODEL*, use this parameter to specify the event Model.

Note: This parameter requires a [specific syntax](#).

`-execution_delay (-
delay)`

Use this parameter if you are creating a Scheduling-type Rule Table.

If your Gateway server is out of service during a period when events are scheduled, this parameter enables you to specify the handling of the un-triggered events on server restoration.

- **UNTIL_NEXT** (default) = when the Gateway server restarts, it executes the next scheduled action at the normal scheduled time, ignoring programmed actions that were not executed when the server was out of service.
- **INFINITE** = when the Gateway server restarts, it executes all actions that were scheduled but not executed due to server incapacity.
- **1 ... 2147483647** = enter a value to specify the maximum time within which a scheduled action can be executed on server restart. If this delay is exceeded, the programmed action is abandoned.

Example: You schedule an action to be triggered at 12:30 p.m. Friday and enter a **Special value** delay of one hour. If the server is out of service from 12:00 p.m. to 1:00 p.m. Friday, the action is triggered on server restart because restart occurs within the specified delay. However, if the server is out of service from 12:00 p.m. to 2:00 p.m. Friday, the action is abandoned because the allowed delay value is

If your Gateway server is out of service during a period when events are scheduled, this parameter enables you to specify the handling of the un-triggered events on server restoration.

- **UNTIL_NEXT**
(default) = when the Gateway server restarts, it executes the next scheduled action at the normal scheduled time, ignoring programmed actions that were not executed when the server was out of service.
- **INFINITE** = when the Gateway server restarts, it executes all actions that were scheduled but not executed due to server incapacity.
- **1 ... 2147483647** = enter a value to specify the maximum time within which a scheduled action can be executed on server restart. If this delay is exceeded, the programmed action is abandoned.

Example: You schedule an action to be triggered at 12:30 p.m. Friday and enter a **Special value** delay of one hour. If the server is out of service from 12:00

	p.m. to 1:00 p.m. Friday, the action is triggered on server restart because restart occurs within the specified delay. However, if the server is out of service from 12:00 p.m. to 2:00 p.m. Friday, the action is abandoned because the allowed delay value is exceeded.
<code>-watched_directory (-watchdir)</code>	Use this parameter if you are creating a Directory Scanning type Rule Table. Enter a string which defines the watched directory.
<code>-watch_period (-wp)</code>	Use this parameter if you are creating a Directory Scanning type Rule Table. Enter a positive integer which defines the watch period {1 ... 2147483647 } in seconds. Default = 10.
<code>-work_directory (-workdir)</code>	Use this parameter if you are creating a Directory Scanning type Rule Table. Enter a string which defines the work directory.
<code>-is_optimized_cleaning (-optclean)</code>	Use this parameter if you are creating a Directory Scanning type Rule Table. • Y (default) = Yes - optimized cleaning is activated • N = No - optimized cleaning is not activated

<code>-prop_list_ name (-pln)</code>	<i>Only for JMS or Axway Messaging</i> Enter a Property List name.
<code>-xms_ connector_ name (-xcn)</code>	Use this parameter if you are creating an XMS-type Rule Table. Enter an Axway Messaging connector name.

Usage rules**Execution type parameters:**

If you set `default_execution_type` to:

- *COMMAND*, then enter a value for the `default_command_line` parameter.
- *MODEL*, then enter a value for the `default_model` parameter.
- *PERL*, then enter a value for the `default_perl_script` parameter.

Restricted processing parameters

To limit XFER_CHANGE_STATE Decision Rule processing to a limited set of Transfers, you must set the parameter `restrict_processing` to **Y** (yes). Gateway then restricts processing based one or more conditions that you set for the following parameters:

- `restrict_processing_on_direction`: restrict processing to either **Input** or **Output** Transfers
- `restrict_processing_on_mode`: restrict processing to either **Initiator** or **Responder** mode Transfers
- `restrict_processing_on_state`: restrict processing to either **Ended**, **Canceled** or **Acknowledged** Transfers
- `restrict_processing_on_type`: restrict processing to Transfers of the specified type (TRANS, POLL, LOT,...)
- `restrict_processing_on_protocol`: restrict processing to Transfers of the specified protocol (PEL, ODETTE, PHSE,...)

If multiple values are specified for a parameter, these must be separated by a comma with no extra spaces.

Modifying a Rule Table: *peladm update_ruletable*

Syntax

`peladm update_ruletable` [see parameters below]

Description	Use this command to set or modify the parameters of an existing Rule Table.

Parameters	<i>Mandatory parameter</i> <code>-name (-n)</code>	Enter the name of the Rule Table you want to update.
	<code>-status (-stat)</code>	Enter one of the following states: • ACTIVE (default) • INACTIVE
	<code>-type_of_rules_table (-type)</code>	Enter one of the following Rule Table types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS

`-default_
execution_type
(-det)`

Enter the type of process to be triggered if no process is triggered by an associated Decision Rule. Enter one of the following values:

- **MODEL** = the default execution is defined by a Model. Specify the Model in the `default_model (-dm)` parameter below.
 - **AMTRIX** = redirect the Transfer to AMTriX for processing
 - **PERL**
 - **COMMAND** = the default execution is a command line or script specified in the `default_command_line` parameter. Refer to [Executing scripts as command lines from Decision Rules](#).
 - **TO_XMS**
 - **NONE** (default)
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to [Purging for transfers routed from Gateway to Integrator](#).
 - **DEFAULT** = route the transfer using a Model inside an Application. This option is only available via online commands. The option exists for reasons of compatibility with earlier product versions.
-

<code>-link_transfers</code> (-lt)	Use this parameter if you enter the value <i>MODEL</i> for the <code>default_execution_type</code> parameter. Enter one of the following values: • Y (default) = Yes - create a logical link between the incoming and the outgoing Transfers • N = No - no link between the incoming and the outgoing Transfers
<code>-default_model</code> (-dm)	If you entered the value <i>MODEL</i> for the <code>default_execution_type</code> parameter, enter a Model name.
<code>-default_perl_script</code> (-dps)	If you entered the value <i>PERL</i> for the <code>default_execution_type</code> parameter, enter the Perl script name and path.
<code>-default_command_line</code> (-dcl)	If you entered the value <i>COMMAND</i> for the <code>default_execution_type</code> parameter, enter the execution command line.
<code>-add_decisionrule</code> (-a)	Enter the name(s) of the Decision Rule (s) you want to add.
<code>-delete_decisionrule</code> (-d)	Enter the name(s) of the Decision Rule (s) you want to delete.
<code>-delete_all_decisionrules</code> (-dad)	Delete all Decision Rules. Enter one of the following values: • Y = Yes • N = No
<code>-move_up</code> (-mu)	Enter the name(s) of the Decision Rule (s) you want to move upwards in the evaluation order list.
<code>-move_down</code> (-md)	Enter the name(s) of the Decision Rule (s) you want to move downwards in the evaluation order list.

```
-restrict_  
processing (-  
rp)
```

Select whether or not to activate restriction criteria parameters for Decision Rules processing.

Enter one of the following values:

- **Y** = Yes
- **N** (default) = No

If you enter the **Y** (yes) value, you must also enter at least one value for one of the following parameters:

- `restrict_processing_on_state`
- `restrict_processing_on_direction`
- `restrict_processing_on_type`
- `restrict_processing_on_mode`
- `restrict_processing_on_protocol`

```
-restrict_  
processing_on_  
state (-rpos)
```

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **ENDED**
- **CANCELED**
- **ACKNOWLEDGED**

```
-restrict_  
processing_on_  
direction (-  
rpod)
```

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **IN**
 - **OUT**
-

`-restrict_
processing_on_
type (-rpot)`

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **TRANS**
- **LOTS**
- **POLL**
- **LIST**
- **EERP**
- **SELECT**
- **MSG**
- **DIR**

`-restrict_
processing_on_
mode (-rpom)`

You can enter one or more of the following values if you entered the value **Y** (yes) for the `restrict_processing (-rp)` parameter:

- **INITIATOR**
 - **RESPONDER**
-

`-restrict_
processing_on_
protocol (-
rpop)`

You can enter one or more of the following values if you entered the value Y (yes) for the `restrict_processing (-rp)` parameter:

- **PEL**
- **ODETTE**
- **PHSE**
- **PHSD**
- **FTP**
- **HTTP**
- **FTP_HTTP**
- **POP3**
- **SFTP**
- **SMTP**
- **AS2**
- **AS3**
- **AS1_POP3**
- **AS1_SMTP**
- **RN_HTTP**
- **RN_POP3**
- **RN_SMTP**
- **SWIFTNET**
- **JMS**
- **X420**
- **X400**

`-log_level (-
ll)`

To define the logging level, enter one of the following values:

- **DEFAULT_LEVEL** (default)
 - **0 ... 2**
-

```
-is_rules_  
table_exclusive  
(-excl)
```

By default, Gateway analyzes the Decision Rules in a Rule Table in their listed order, stopping at the first Rule that matches a complete set of conditions.

Enter one of the following values:

- **Y** = (default)
- **N** = No - enables Gateway to continue analyzing the table list after processing a matching Rule

```
-is_user_exits_  
activated (-  
exit)
```

Rule Tables are inherently associated with exit processes that are automatically called during Rule processing.

Enter one of the following values:

- **Y** (default) = Yes - enable all inherent exits
- **N** = No - disable all inherent exits

```
-temporal_  
event_model (-  
tem)
```

If you are creating a Scheduling-type Rule Table (`-type = TEMPORAL`), and you set the default execution type to *MODEL*, use this parameter to specify the event Model.

Note: This parameter requires a [specific syntax](#).

`-execution_
delay (-delay)`

Use this parameter if you are creating a Scheduling-type Rule Table.

If your Gateway server is out of service during a period when events are scheduled, this parameter enables you to specify the handling of the un-triggered events on server restoration.

- **UNTIL_NEXT** (default) = when the Gateway server restarts, it executes the next scheduled action at the normal scheduled time, ignoring programmed actions that were not executed when the server was out of service.
- **INFINITE** = when the Gateway server restarts, it executes all actions that were scheduled but not executed due to server incapacity.
- **1 ... 2147483647** = enter a value to specify the maximum time within which a scheduled action can be executed on server restart. If this delay is exceeded, the programmed action is abandoned.

Example: You schedule an action to be triggered at 12:30 p.m. on Friday and enter a **Special value** delay of one hour. If the server is out of service from 12:00 p.m. to 1:00 p.m. on Friday, the action is triggered on server restart because restart occurs within the specified delay.

However, if the server is out of service from 12:00 p.m. to 2:00 p.m. on Friday, the action is abandoned because the allowed delay value is exceeded.

`-watched_
directory (-
watchdir)`

Use this parameter if you are creating a Directory Scanning type Rule Table.

Enter a string which defines the watched directory.

<code>-watch_period</code> (<code>-wp</code>)	Use this parameter if you are creating a Directory Scanning type Rule Table. Enter a positive integer which defines the watch period { 1 ... 2147483647 } in seconds. Default = 10.
<code>-work_directory</code> (<code>-workdir</code>)	Use this parameter if you are creating a Directory Scanning type Rule Table. Enter a string which defines the work directory.
<code>-is_optimized_cleaning</code> (<code>-optclean</code>)	Use this parameter if you are creating a Directory Scanning type Rule Table. • Y (default)= Yes - optimized cleaning is activated • N = No - optimized cleaning is not activated
<code>-prop_list_name</code> (<code>-pln</code>)	<i>Only for JMS or Axway Messaging</i> Enter a Property List name.
<code>-restrict_processing_on_mode</code> (<code>-rpom</code>)	Enter one of the following: • INITIATOR • RESPONDER
<code>-xms_connector_name</code> (<code>-xcn</code>)	Use this parameter if you are creating an XMS-type Rule Table. Enter an Axway Messaging connector name.

Usage rules**Execution type parameters:**

If you set `default_execution_type` to:

- *COMMAND*, then enter a value for the `default_command_line` parameter.
- *MODEL*, then enter a value for the `default_model` parameter.
- *PERL*, then enter a value for the `default_perl_script` parameter.

Restricted processing parameters:

To limit Decision Rule processing to a limited set of Transfers, you must set the parameter `restrict_processing` to **Y** (yes). Gateway then restricts processing based on one or more conditions that you set for the following parameters:

- `restrict_processing_on_direction`: restrict processing to either **Input** or **Output** Transfers
- `restrict_processing_on_mode`: restrict processing to either **Initiator** or **Responder** mode Transfers
- `restrict_processing_on_state`: restrict processing to either **Ended**, **Canceled** or **Acknowledged** Transfers
- `restrict_processing_on_type`: restrict processing to Transfers of the specified type (TRANS, POLL, LOT,...)
- `restrict_processing_on_protocol`: restrict processing to Transfers of the specified protocol (PEL, ODETTE, PHSE,...)

If multiple values are specified for a parameter, these must be separated by a comma with no extra spaces.

Example 1

The following example defines the order of Decision Rules in the list.

```
peladm update_ruletable -a TABFTP
peladm update_ruletable -a "TABPHSE,
TABHTTP"
```

Gateway adds the Decision Rule TABFTP to the list, followed by TABPHSE and TABHTTP. Gateway evaluates the list of Decision Rules in this order: TABFTP > TABPHSE > TABHTTP.

Example 2

Use the parameters `move_up` and `move_down` to modify the order of Decision Rules in the list. For example, the command:

```
peladm update_ruletable -move_up TABPHSE
```

moves Decision Rule TABPHSE up one position in the list.

Example 3 The following example sets the default execution type for the whole Rule Table. In addition it uses the `restrict_processing` parameter to restrict processing to Transfers with Ended state and of the type TRANS.

```
peladm update_ruletable -default_execution_type Default
                        -restrict_processing Y
                        -restrict_processing_on_state E
                        -restrict_processing_on_type TRANS
```

Deleting a Rule Table: `peladm delete_ruletable`

Syntax	<code>peladm delete_ruletable [-name]</code>	
Description	Use this command to delete a Rule Table. Repeat the command for each Rule Table you want to delete.	
Parameter	<code>-name (-n)</code>	Enter the name of the Rule Table you want to delete.
	<code>-type_of_rules_table (-type)</code>	Enter the type of Rule Table you want to delete. Enter one of: • XFER_CHANGE_STATE • TEMPORAL • FILE_STORING • XMS
Example	Enter the command: <pre>peladm delete_ruletable -n MyRuleTable</pre> Gateway deletes the Rule Table with the name <i>MyRuleTable</i> .	

Displaying Rule Table properties: `peldsp display_ruletable`

Syntax	<code>peldsp display_ruletable [-name] [-type_of_rules_table]</code>
Description	Use this command to display the properties of a specified Rule Table.

Parameters	<code>-name (-n)</code> : Enter the name of the Rule Table whose properties you want to display.
	<code>-type_of_rules_table (-type)</code> : Enter the type of Rule Table for which you want to display the properties.
	Enter one of:
	• XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS
Example	Enter the command: <pre>peldsp display_ruletable -n MyRuleTable</pre> Gateway returns the properties of the Rule Table with the name <i>MyRuleTable</i> .

Displaying all Rule Table names: peldsp select_ruletable

Syntax	<pre>peldsp select_ruletable [-name] [-default_ execution_type] [-default_model] [-default_ perl_script] [-type_of_rules_table] [-xms_ connector_name]</pre>
Description	Use this command without parameters to display the names of all Rule Tables. Alternatively, add parameters as filtering criteria to restrict the return results.

Parameters		
	<code>-name (-n)</code>	Rule Table name.
	<code>-default_execution_type (-det)</code>	Type of process to be triggered if no process is triggered by an associated Decision Rule. Enter one of: • MODEL • AMTRIX • PERL • COMMAND • TO_XMS • NONE (default) • LOG_CLEANING • MAILBOX_CLEANING • STAT_CLEANING • XIB • DEFAULT
	<code>-default_model (-dm)</code>	If you entered the value <i>MODEL</i> for the <code>default_execution_type</code> parameter, you can add a Model name.
	<code>-default_perl_script (-dps)</code>	If you entered the value <i>PERL</i> for the <code>default_execution_type</code> parameter, you can enter the Perl script name and path.
	<code>-type_of_rules_table (-type)</code>	Enter one of the following types: • XFER_CHANGE_STATE (default) • TEMPORAL • FILE_STORING • XMS
	<code>-xms_connector_name (-xcn)</code>	Use this parameter if you are displaying an XMS-type Rule Table. Enter an Axway Messaging connector name.

Example

Enter the command:

```
peldsp select_ruletable
```

Gateway returns the names of all Rule Tables existing on the server.

Axway Gateway: Managing Events

Decision Rules and Rule Tables: Parameters List

The following table lists all Decision Rule and Rule Table command parameters in alphabetical order. The corresponding field name in the Gateway GUI is displayed in ***italics***

Parameter	Meaning	Values
<code>add_cond (-a)</code> <i>(None)</i>	Modifying Decision Rule Add one or more conditions to an existing Decision Rule	Condition expression that comprises: • argument name • operator • value
<code>add_decisionrule (-a)</code> <i>(None)</i>	Rule Tables Add a Decision Rule to the list	Decision Rule name
<code>cond_arg (-ca)</code> <i>Argument</i>	Creating Decision Rule Define one or more conditions for the Decision Rule	Condition expression that comprises: • argument name • operator • value
<code>command_line (-cl)</code> <i>Batch command</i>	Command line to execute. Can include symbolic variables.	Path of the command. For example: <code>/bin/ksh/p_home_dir /run_time/ftpconf.sh</code>
<code>comments (-cts)</code> <i>Comments</i>	Free-text description	Maximum: 80 alphanumeric characters

Parameter	Meaning	Values
<code>default_ command_line (-dcl)</code> <i>Batch command</i>	Rule Tables Define default command to execute if no Decision Rule matches the Transfer parameters. Set this parameter if you set the <code>default_ execution_type</code> to <i>Command</i> .	Path of the command

Parameter	Meaning	Values
default_ execution_type (-det) Default execution type	Rule Tables Define default processing to use if no Decision Rule matches the current Transfer parameters.	One of the following values: • MODEL = the default execution is defined by a Model. Specify the Model in the <code>default_model (-dm)</code> parameter below. • AMTRIX = redirect the Transfer to AMTriX for processing • PERL • COMMAND = the default execution is a command line or script specified in the <code>default_command_line</code> parameter. Refer to Executing scripts as command lines from Decision Rules. • TO_XMS • NONE (default) • LOG_CLEANING • MAILBOX_CLEANING • STAT_CLEANING • XIB = redirect the Transfer to Integrator for processing. The payload file for a Transfer which was routed to Integrator is not deleted by the purge operation. For more information, refer to Purging for transfers routed from Gateway to Integrator. • DEFAULT (<i>only via command line</i>) = route the transfer using a Model inside an Application. This option exists for reasons of compatibility with earlier product versions.

Parameter	Meaning	Values
default_model (-dm) Model	Rule Tables Define default Model to use if no Decision Rule matches the current Transfer parameters. Set this parameter if you set <code>default_execution_type</code> to <i>Model</i> .	Name of a General Model
default_perl_script (-ds) Perl script	Rule Tables Define default Perl script to use if no Decision Rule matches the current Transfer parameters. Set this parameter if you set <code>default_execution_type</code> to <i>Perl</i> .	Path of the Perl script
delete_cond (-d) (None)	Modifying Decision Rule Delete Decision Rule condition	Condition expression that comprises: • argument name • operator • value
delete_all_decisionrules (-dad)		One of the following values: • Y (yes) (default) • N (no)
delete_decisionrule (-d) (None)	Rule Tables Disable Decision Rule in the list	Decision Rule name
execution_delay (-delay)	A positive integer which defines the execution delay	One of the following values: • UNTIL_NEXT (default) • INFINITE • 1 ... 2147483647

Parameter	Meaning	Values
execution_type (-et) Execution type	Decision Rule The type of process to be triggered by a Decision Rule.	One of the following values: • MODEL • AMTRIX • PERL • COMMAND • TO_XMS • NONE • LOG_CLEANING • MAILBOX_CLEANING • STAT_CLEANING • XIB (for Integrator) • DEFAULT (<i>only via command line</i>) = route the transfer using a Model inside an Application. This option exists for reasons of compatibility with earlier product versions.
import_cond (ic) (None)	Creating Decision Rule Import a text file that defines list of conditions to apply in this Decision Rule	Full file name and path that contains set of conditions to import.
is_optimized_cleaning (-optclean) Optimized cleaning	If this boolean is set, the Optimized cleaning function is activated	One of the following values: • Y (yes) (default) • N (no)
is_rules_table_exclusive (-excl)	If this boolean is set, the Rule Table is exclusive	One of the following values: • Y (yes) (default) • N (no)
is_user_exits_activated (-exit)	If this boolean is set, the user exits are called	One of the following values: • Y (yes) (default) • N (no)

Parameter	Meaning	Values
link_transfers (-lt) Link input and output transfers	Decision Rule Link output Transfer Request to input Transfer. If you set this value to Yes : - the input Transfer Request parameter <code>route_to_xfer</code> is set to the output Transfer identifier - the output Transfer Request parameter <code>route_from_xfer</code> is set to the input Transfer identifier	One of the following values: Y (yes) (default) N (no)
log_level (-ll) (None)	Rule Tables Defines the log level for error messages	One of the following values: 0 = minimum log level 1 = medium (reports errors regarding Decision Rule failures and so on) 2 = detailed (includes information regarding symbolic variables)

Parameter	Meaning	Values
model (-ml) Model	Creating Decision Rule General Model to use for routing definition	General Model name. Maximum: 26 alphanumeric characters. The Model name that you specify must have a corresponding General Model that includes the dest_alias definition. Either the General Model or the associated Protocol Model must include the following parameters: • <code>file_component</code> • <code>appli</code> You can use symbolic variables if you want to retrieve the value for these parameters from the incoming Transfer.
move_down (-md) (None)	Updating Rule Table Move Decision Rule down one place in the list	Decision Rule name
move_up (-mu) (None)	Updating Rule Table Move Decision Rule up one place in the list	Decision Rule name
name (-n) Name	Decision Rule or Rule Table Decision Rule or Rule Table name	Decision Rule or Rule Table name. Maximum: 26 alphanumeric characters.
perl_script (-ps) Perl script	Decision Rule Perl script to use to define routing	Path of Perl script. Maximum: 128 characters. For example, in UNIX: <code>/home/users/utill/scripts/my_script.pl</code>
prop_list_name (-pln)	Name of Property List	

Parameter	Meaning	Values
restrict_processing (-rp) Restrict processing	Rule Tables Restricts processing to Transfers that conform to the conditions that you define for the following parameters: • restrict_processing_on_state • restrict_processing_on_direction • restrict_processing_on_type • restrict_processing_on_mode • restrict_processing_on_protocol	One of the following values: • Y (yes) • N (no)
restrict_processing_on_direction (-rpod) Transfer direction	Rule Tables Restricts processing to either incoming or outgoing Transfers	One of the following values: • I (In) • O (Out)
restrict_processing_on_mode (-rpom) Transfer mode	Rule Tables Restricts processing to Transfer in either Initiator or Responder mode	One of the following values: • I (Initiator) • R (Responder)
restrict_processing_on_state (-rpos) Transfer state	Rule Tables Restricts processing to Transfers of the specified state	One of the following values: • E (Ended) • C (Canceled) • A (Acknowledged)

Parameter	Meaning	Values
restrict_ processing_on_ type (-rpot) Transfer type	Rule Tables Restricts processing to Transfers of the specified type	One of the following values: • TRANS • POLL • LOTS • LIST • EERP • DIR • SELECT • MSG
restrict_ processing_on_ protocol (- rpot) Transfer protocol	Rule Tables Restricts processing to Transfers of the specified protocol(s)	One of the following values: • PEL • ODETTE • PHSE • PHSD • FTP • HTTP • FTP_HTTP • POP3 • SFTP • SMTP • AS2 • AS3 • AS1_POP3 • AS1_SMTP • RN_HTTP • RN_POP3 • RN_SMTP • SWIFTNET • JMS • X420
temporal_ event_model (- tem)	Temporal event model with the specific syntax	

Parameter	Meaning	Values
<code>type_of_decisionrules (-type)</code>	Selects type of Decision Rules	One of the following values: • XFER_CHANGE_STATE • TEMPORAL • FILE_STORING • XMS
<code>type_of_rules_table (-type)</code>	Selects type of Rule Table	One of the following values: • XFER_CHANGE_STATE • TEMPORAL • FILE_STORING • XMS
<code>update_cond (-u)</code>	Update one or more conditions for the Decision Rule	Condition expression that comprises: • argument name • operator • value
<code>watched_directory (-watchdir)</code> Watch directory	Defines the watched directory	String
<code>watch_period (-wp)</code> Watch period	Defines the watch period	Positive integer
<code>work_directory (-workdir)</code> Work directory	Defines the work directory	String
<code>xms_connector_name (-xcn)</code>	Name of Axway Messaging connector	
<code>mail_xpriority (-mailxprio)</code>	SMTP protocol only (with or without S/MIME) - the value for X-Priority mail header (SEPA mail standard)	Integer, between 1 and 5

[Command syntax for Scheduling Event Models](#) (date syntax)

Command syntax for Scheduling Event Models

[Introduction](#)

[Elements of the TEM: Descriptors and calendars](#)

[Assembling the TEM](#)

Introduction

When you create or update a Scheduling-type Rule Table using command lines, use the *Temporal Event Model (TEM) syntax* to set a value for the `temporal_event_model` (`-tem`) parameter. TEM enables you to describe one or more instances in time.

Elements of the TEM: Descriptors and calendars

A TEM consists of one or more calendars that are made up of descriptors. Descriptors specify a date or time:

- [Simple date descriptor](#)
- [Complex date descriptor](#)
- [Simple time descriptor](#)
- [Complex time descriptor](#)

You combine descriptors to create calendars and you combine [calendars](#) to create TEMs.

All the examples provided in this topic consist of a single line. In most instances you can force line breaks, for example to facilitate error searches.

Complete syntax

The complete syntax for a TEM object is a date descriptor followed by a time descriptor:

datedescriptor timedescriptor

Note omitting the time descriptor may (in some cases) lead to error messages

The rest of this topic describes different forms of date and time descriptors.

Simple date descriptor

The simple date descriptor enables you to describe a day or set of days. It comprises four fields enclosed between brackets, the fields separated by semicolons.

To apply a time offset, you add characters after the closing brackets. For more details, refer to [Time offset](#).

Syntax: [**daysOfWeek**;**daysOfMonth**;**months**;**year**]

Note On UNIX systems, semicolons must be escaped, syntax becomes :
[**daysOfWeek** **daysOfMonth** **months** **year**]

where:

- **daysOfWeek** = a number (or numbers) representing one or more weekdays as follows:

1 = Monday	5 = Friday
2 = Tuesday	6 = Saturday
3 = Wednesday	7 = Sunday
4 = Thursday	

Enter single days as a single integer. For example, for Saturday enter: 6

Enter intervals of days as integers separated by hyphens. For example, for Monday to Wednesday enter: 1-3

To enter all days of the week, enter: *

To enter a periodic interval enter the period length after the slash symbol /.

Examples:

1-6/2 = every second day between the Monday and the Saturday beginning on Monday

*/3 = every 3rd day of the week beginning on Monday

- **daysOfMonth** = a number (or numbers) between 1 and 31 representing a day or days of the month.

Enter single dates as a single integer. For example, for the 5th day of the month enter: 5

Enter intervals of consecutive dates as two integers separated by hyphens. For example, for the 11th to the 15th enter: 11-15

To enter all dates of the category, enter: *

To enter a periodic interval enter the period length after the slash symbol /.

Examples:

1-6/2 = every second date between the 1st and the 6th beginning on the 1st

*/3 = every 3rd date of the month beginning on the 1st

- **months** = a number (or numbers) representing a month (or months) as follows:

1 = January	7 = July
2 = February	8 = August
3 = March	9 = September
4 = April	10 = October
5 = May	11 = November
6 = June	12 = December

Enter single months as a single integer. For example, for March enter: 3

Enter intervals of consecutive months as two integers separated by hyphens. For example, for September to December: 9-12

To enter all months of the year, enter: *

To enter a periodic interval enter the period length after the slash symbol /.

Examples:

1-6/2 = every second month between January and June beginning with January

* / 3 = every 3rd month of the period beginning with January

- **year** = one or more years

Use the format...	To indicate...	Example
yyyy	A single year	2009
yyyy-yyyy	An interval of consecutive years	2000-2009
*	All years	*
<yyyy	Before the given year	<2009
>yyyy	After the given year	>2009

To indicate the relation of a given year to a leap year, follow the year with the upright slash symbol (/) followed by a whole number (expressed in base 10) representing a binary four bit mask.

When the bit equals 1, the corresponding years are selected, as follows:

- bit 1 = leap year
- bit 2 = leap year + 1 year
- bit 3 = leap year + 2 years
- bit 4 = leap year + 3 years

To calculate the bit mask from binary to base 10:

Binary	Equivalent base 10
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10

1010	11
1011	12
1100	12
1101	13
1110	14
1111	15

Example 1

2006-2030 | 8 = *all leap years between 2006 and 2030*

Explanation: 8 to the base 10 is expressed as 1000 in binary. The first bit (reading left to right) of 1000 is set to the value 1, so all leap years are selected.

Example 2

2007-2025 | 9 = *all leap years and all years preceding leap years between 2007 and 2025.*

Explanation: 9 to the base 10 is expressed as 1001 in binary. The first bit and the fourth bit (reading left to right) of 1001 are set to the value 1, so all leap years and all leap years +3 are selected.

Time offset

You can offset the time expressed within the brackets of a TEM expression by a number of days.

- To offset the expression later in time by **d** number of days, use the syntax:

`[expression]>>d`

- To offset the expression earlier in time by **d** number of days, use the syntax:

`[expression]<<d`

Example:

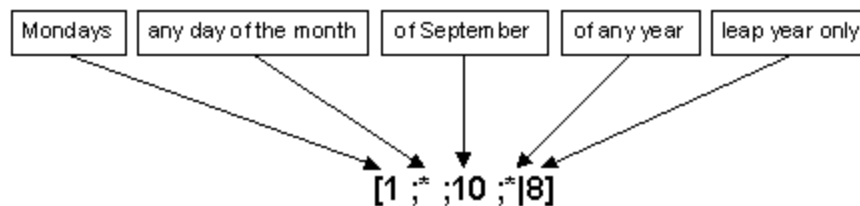
`[* ; 1-20/4 ; 10-12, 2 ; <2010 | 12] <<1`

indicates the day before the days described by the expression inside the brackets.

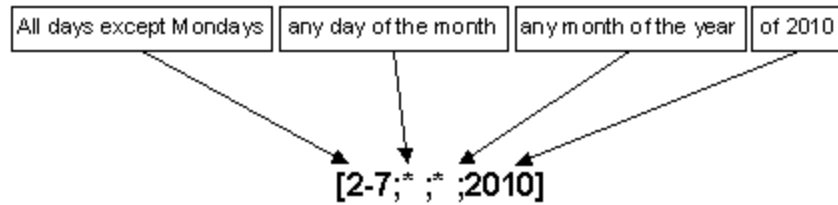
Commented examples

A date descriptor expression describes all of the dates that satisfy all of the elements contained in the expression. The following examples illustrate how to apply the syntax.

Example 1:



Example 2:



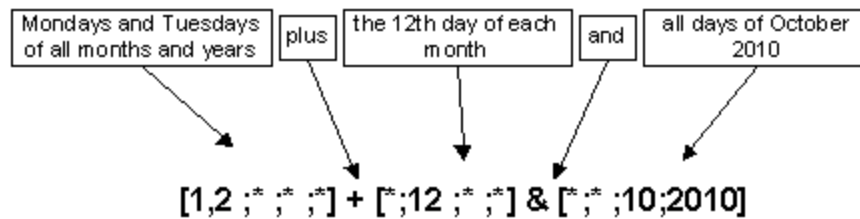
Complex date descriptor

You can use the following operators to link two or more simple date descriptors into a complex date descriptor.

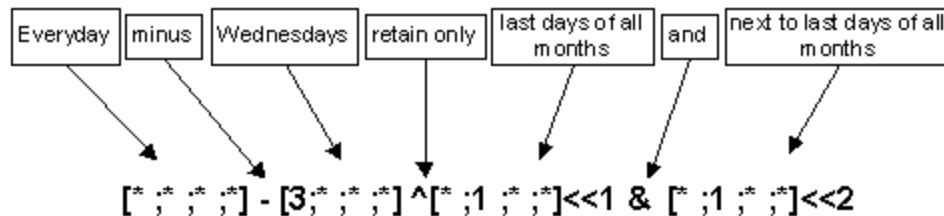
Operator	Description
+	add
-	remove
^	keep only
&	and

Commented examples

Example 1:



Example 2:



Simple time descriptor

The simple time descriptor enables you to describe a time of day or set of times of days. It comprises two fields separated by the letter **H** in the format:

HoursHMinutes

Hour and Minute field syntax

Hour and minutes fields obey the same syntactic rules. To complete the fields you can enter:

- A single hour or minute as a single integer. For example, for 3 a.m. enter: 3
- Intervals of consecutive hours or minutes as two integers separated by hyphens. For example, for 5 a.m. to 7 a.m. enter: 5-7
- All times of a category. Enter: *
- A periodic interval. Enter the period length after the slash symbol /.

Examples:

$1-6/2$ = hour or minute 1, 3, 5

$*/3$ = in the hour field indicates the hours 0, 3, 6, 9, 12, 15, 18, 21

Complex time descriptor

You can link two or more simple time descriptors into a complex date descriptor. To link simple time descriptors separate them with semicolons.

Example

`15H12 ; 12-16H30`

indicates 15:12 and 12:30, 13:30, 14:30, 15:30, 16:30.

Calendar

A calendar is an association of a complex date descriptor and a complex time descriptor. You can create any logically valid combination of elements.

Example 1

`[*;*;*;*]16H15`

indicates every day at 16:15.

Example 2

`[1;*;*;*]-[*;*;10;*] 16H30,50`

indicates every Monday except in October at 16:30 and 16:50.

Assembling the TEM

A TEM is a calendar or a combination of calendars. To combine calendars, create consecutive chains of calendar descriptors.

Example

```
[*;*;*;*]16H15 [1;*;*;*]-[*;*;10;*] 16H30,50
```

indicates every day at 16:15, and every Monday with the exception of Mondays in October at 16:30 and 16:50.

Axway Gateway: Managing Events

Executing scripts as command lines from Decision Rules

Caution Please note that care should be taken when **using shell scripts**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

[UNIX platforms](#)

[Windows platforms](#)

[Advanced options](#)

UNIX platforms

Executing UNIX shell scripts

On UNIX platforms you can configure Gateway to execute a shell script as a result of a Decision Rule analysis. Do one of the following:

- In a Rule Table:
 - Enter **COMMAND** as the value for the `default_execution_type` parameter
 - Enter the path and name of the shell script to execute in the `default_command_line` parameter
- In a Decision Rule:
 - Enter **COMMAND** as the value for the `execution_type` parameter
 - Enter the path and name of the shell script to execute in the `command_line` parameter

When the execution criteria for the Decision Rule are satisfied, Gateway executes the specified shell script.

Using environment variables

Gateway enables you to retrieve transfer characteristics through the use of environment variables.

To retrieve the value of any Gateway transfer attribute, you prefix the name of the attribute in the script with `x_`. The list of available attribute variables is the same as the list of attributes for Decision Rule execution criteria and for Models.

For example, to use the values of the following transfer attributes in the shell script:

- `file_name`
- `originator`
- `destination`

...in the shell script, you retrieve the values by listing them in the following syntax:

```
$x_file_name, $x_originator, $x_destination
```

Retrieving all variables

If you want to retrieve a list of every variable available for a transfer in the script, you can use a command such as:

```
env | grep x_
```

All variable names and their values are printed in the `Q0000x.out` router trace file.

Specifying arguments

Additionally, you can specify arguments in the script command line. For example, you can enter the following shell script path and name:

```
<Gateway installation directory>/run_time/scripts/sample.sh var1 var2
```

`var1` and `var2` can be retrieved by using the syntax `$1`, `$2` as usual in the script.

Sample shell script

A sample `xpr` shell script (`xpr.sh` for UNIX, `xpr.bat` for Windows) is provided with Gateway. These files are located in `<Gateway installation directory>/runtime/etc`.

Windows platforms

On Windows platforms you can retrieve variables in the usual way using batch files. In the batch file, use the syntax `%x_param%` or `%1`, `%2` for parameters.

Advanced options

In the [Advanced options of the configuration menu](#), set the `started_command_user_control` option to **yes** if you want to control the maximum number of simultaneous activated scripts. This value is determined by the `job_max` parameter (that is only taken into account by Gateway if the `started_command_user_control` parameter is set to **yes**).

Decision Rule usage example: Routing

[Model-based routing via the GUI](#)

[Configuring Gateway for routing](#)

[Results](#)

Model-based routing via the GUI

This topic provides an example of how to specify the routing behavior of Gateway.

The example describes how to use the GUI to route a file from an Axway Transfer product to an FTP client via Gateway, using a Decision Rule that applies a Model as the execution type.

The resulting series of processes is summarized by the following schema:



Configuring Gateway for routing

To configure your Gateway for routing you execute the following main tasks:

- [Create the Local and Remote Sites](#)
- [Create the execution Model](#)
- [Create the Decision Rule](#)
- [Create the Rule Table and link the Decision Rule](#)

The following sections provide details about how to accomplish these tasks.

Creating Local and Remote Sites

Create the following Sites:

- In Gateway:
 - Local Site = GATEWAY
 - Remote Site = TRANSFER
 - FTP Template Site = TFTP
- In Axway Transfer:
 - Local Site = TRANSFER
 - Remote Site = GATEWAY

Creating the execution Model

In this example, the Decision Rule specifies a Model as the execution type for the received Transfer. Create a General Model and, optionally, a corresponding Protocol Model with the same name. The General Model is mandatory for routing, because you can only define the Transfer destination in the General Model and not in a Protocol Model. For more information, refer to [About Models](#).

1. Create a General Model with the following properties:
 - Model name: MODEL_FTP
 - Protocol: GENERAL
 - Destination alias: TFTP
2. Create a Protocol Model with the following properties:
 - Model name: MODEL_FTP
 - Protocol: FTP
 - Destination: FTPCLIENT
 - File component: '&(dir_path)/&(file_component)'
 - Application name: '&(appli)'

The file component and application parameters are mandatory. You can define these parameters in either the General or the Protocol Model. In this example, the definition uses symbolic variables (&(appli)) to retrieve the corresponding parameter values from the incoming Transfer. Using symbolic variables not only simplifies the definition but guarantees transfer coherence.

Creating the Decision Rule

To define a Decision Rule that applies Model-based routing, create a Transfer Change State type Decision Rule as follows:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node.

Gateway expands the node to display the following four sub-nodes:

 - Scheduling
 - Directory Scanning
 - Transfer Change State

- XMS

Each sub-node represents a [Decision Rule/Rule Table category](#).

- Right-click the **Transfer Change State** sub-node. A context menu appears. Select **New > Decision Rule**.

Gateway displays a *New Decision Rule* window.

- In the *New Decision Rule* window, enter the following data.

For the parameter/condition...	Enter the value...	Description
Name	Rule1	Unique name for the Decision Rule.
Active	Click the check box	Activates the Decision Rule.
Comments	(optional descriptive text)	Free text description of the Decision Rule. Enter up to 80 alphanumeric characters. This description appears in the comment column of the Decision Rule list display pane.
Conditions		
1st condition	protocol = PeSIT	The protocol used in the incoming Transfer must be PeSIT.
2nd condition	destination#& (originator)	The destination must be different from the originator Site in the incoming Transfer.
3rd condition	rec_len[]150,3000	The record length must be greater than or equal to 150 and less than or equal to 3000.
4th condition	direction = I	The direction must be Input. Specifying the direction parameter prevents the Decision Rule from looping on input and output Transfers.
5th condition	file_size][10,50	The file size must not be between the values 10 and 50.

For the parameter/ condition...	Enter the value...	Description
6th condition	state = ended	The state of the incoming Transfer is <i>ended</i>
Processing (if all conditions match)		
Execution type	Model	Execute Model-based routing on output.
Model	MODEL_FTP	Name of the Model to use to execute routing. This Model name must correspond to the name of a General Model and can optionally correspond to a Protocol Model name. The General Model must include the <code>destination_alias</code> parameter. Either the Protocol or the General Model must include the following parameters: • <code>file_component</code> • <code>appli</code>
Link input and output transfers	Click the check box	Creates a logical link between the incoming and the outgoing Transfers.

4. When you have finished entering the values, click **OK** to accept.

Creating the Rule Table

To use an existing Decision Rule, you link the Decision Rule to a Rule Table. For this example we create a Transfer Change State-type Rule Table and link the *Rule1* Decision Rule that we created in the previous section.

Create a Transfer Change State type Rule Table as follows:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node.
Gateway expands the node to display the following four sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule/Rule Table category](#).

2. Right-click the **Transfer Change State** sub-node. A context menu appears. Select **New > Rule Table**.

Gateway displays a *New Rule Table* window.

3. In the **General** tab of the *New Rule Table* window, enter the following data:

Field	Enter the value...	Description
Name	ROUTING1	Unique name for the Rule Table.
Comments	(optional descriptive text)	Free text description of the Rule Table. Maximum: 80 alphanumeric characters.
Exclusive table	Click the check box	Executes only the first Decision Rule that completely matches the criteria.
Default execution type	No action	Specifies that no action is executed if no defined Decision Rule matches the current parameters.
Log level	None	Deactivate logging.

4. In the **Event** tab of the *New Rule Table* window, do not change the defaults (no processing restrictions).
5. In the **Linked Rules** tab of the *New Rule Table* window click **Rule1** in the available rules list to select it. Then click the **Link** button.

Gateway links the *Rule1* Decision Rule to the *Routing1* Rule Table.

6. Click **OK** to validate the new Rule Table.

Gateway closes the *New Rule Table* window and adds the *Routing1* entry to the display of Rule Tables and Decision Rules for Transfer Change State category.

Results

Once you have created and activated the Decision Rule and linked it to a Rule Table, the supervisor component of Gateway monitors gateway activity, analyzing incoming transfers for matches to the Decision Rule.

Incoming Transfer match

Suppose that Gateway receives an incoming transfer with the following parameter values.

- Direction: 'I' (incoming)
- Application: 'APPLI'
- Destination: GATEWAY
- Originator: TRANSFER
- Protocol: 'PeSIT'
- Record length: 1024
- File component: file.txt
- File size: 147
- State: Ended

Gateway checks the parameters of the incoming Transfer against the conditions set in the *Rule1* Decision Rule.

- Condition 1: protocol is PeSIT
- Condition 2: destination (Gateway) is not the same as the originator (Axway Transfer)
- Condition 3: record length is between 150 and 3000
- Condition 4: direction is incoming
- Condition 5: file size is not between 10 and 50
- Condition 6: the incoming file transfer is complete and passed to the *ended* state

Routing result

Since the six conditions specified in the *Rule1* Decision Rule match the parameters of the incoming Transfer, Gateway applies the Model MODEL_FTP definition. After receiving the file from Axway Transfer, Gateway immediately deposits a new Transfer Request and makes the file available to the FTP client.

Axway Gateway: Managing Events

Decision Rule usage example: Routing

[Model-based routing via the GUI](#)

[Configuring Gateway for routing](#)

[Results](#)

Model-based routing via the GUI

This topic provides an example of how to specify the routing behavior of Gateway.

The example describes how to use the GUI to route a file from an Axway Transfer product to an FTP client via Gateway, using a Decision Rule that applies a Model as the execution type.

The resulting series of processes is summarized by the following schema:



Configuring Gateway for routing

To configure your Gateway for routing you execute the following main tasks:

- [Create the Local and Remote Sites](#)
- [Create the execution Model](#)
- [Create the Decision Rule](#)
- [Create the Rule Table and link the Decision Rule](#)

The following sections provide details about how to accomplish these tasks.

Creating Local and Remote Sites

Create the following Sites:

- In Gateway:
 - Local Site = GATEWAY
 - Remote Site = TRANSFER
 - FTP Template Site = TFTP

- In Axway Transfer:
 - Local Site = TRANSFER
 - Remote Site = GATEWAY

Creating the execution Model

In this example, the Decision Rule specifies a Model as the execution type for the received Transfer. Create a General Model and, optionally, a corresponding Protocol Model with the same name. The General Model is mandatory for routing, because you can only define the Transfer destination in the General Model and not in a Protocol Model. For more information, refer to [About Models](#).

1. Create a General Model with the following properties:
 - Model name: MODEL_FTP
 - Protocol: GENERAL
 - Destination alias: TFTP
2. Create a Protocol Model with the following properties:
 - Model name: MODEL_FTP
 - Protocol: FTP
 - Destination: FTPCLIENT
 - File component: '&(dir_path)/&(file_component)'
 - Application name: '&(appli)'

The file component and application parameters are mandatory. You can define these parameters in either the General or the Protocol Model. In this example, the definition uses symbolic variables (&(appli)) to retrieve the corresponding parameter values from the incoming Transfer. Using symbolic variables not only simplifies the definition but guarantees transfer coherence.

Creating the Decision Rule

To define a Decision Rule that applies Model-based routing, create a Transfer Change State type Decision Rule as follows:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node.

Gateway expands the node to display the following four sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule/Rule Table category](#).

2. Right-click the **Transfer Change State** sub-node. A context menu appears. Select **New > Decision Rule**.

Gateway displays a *New Decision Rule* window.

3. In the *New Decision Rule* window, enter the following data.

For the parameter/condition...	Enter the value...	Description
Name	Rule1	Unique name for the Decision Rule.
Active	Click the check box	Activates the Decision Rule.
Comments	(optional descriptive text)	Free text description of the Decision Rule. Enter up to 80 alphanumeric characters. This description appears in the comment column of the Decision Rule list display pane.
Conditions		
1st condition	protocol = PeSIT	The protocol used in the incoming Transfer must be PeSIT.
2nd condition	destination#& (originator)	The destination must be different from the originator Site in the incoming Transfer.
3rd condition	rec_len[]150,3000	The record length must be greater than or equal to 150 and less than or equal to 3000.
4th condition	direction = I	The direction must be Input. Specifying the direction parameter prevents the Decision Rule from looping on input and output Transfers.
5th condition	file_size][10,50	The file size must not be between the values 10 and 50.
6th condition	state = ended	The state of the incoming Transfer is <i>ended</i>
Processing (if all conditions match)		

For the parameter/ condition...	Enter the value...	Description
Execution type	Model	Execute Model-based routing on output.
Model	MODEL_FTP	Name of the Model to use to execute routing. This Model name must correspond to the name of a General Model and can optionally correspond to a Protocol Model name. The General Model must include the <code>destination_alias</code> parameter. Either the Protocol or the General Model must include the following parameters: • <code>file_component</code> • <code>appli</code>
Link input and output transfers	Click the check box	Creates a logical link between the incoming and the outgoing Transfers.

4. When you have finished entering the values, click **OK** to accept.

Creating the Rule Table

To use an existing Decision Rule, you link the Decision Rule to a Rule Table. For this example we create a Transfer Change State-type Rule Table and link the *Rule1* Decision Rule that we created in the previous section.

Create a Transfer Change State type Rule Table as follows:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node.

Gateway expands the node to display the following four sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

Each sub-node represents a [Decision Rule/Rule Table category](#).

- Right-click the **Transfer Change State** sub-node. A context menu appears. Select **New > Rule Table**.

Gateway displays a *New Rule Table* window.

- In the **General** tab of the *New Rule Table* window, enter the following data:

Field	Enter the value...	Description
Name	ROUTING1	Unique name for the Rule Table.
Comments	(optional descriptive text)	Free text description of the Rule Table. Maximum: 80 alphanumeric characters.
Exclusive table	Click the check box	Executes only the first Decision Rule that completely matches the criteria.
Default execution type	No action	Specifies that no action is executed if no defined Decision Rule matches the current parameters.
Log level	None	Deactivate logging.

- In the **Event** tab of the *New Rule Table* window, do not change the defaults (no processing restrictions).
- In the **Linked Rules** tab of the *New Rule Table* window click **Rule1** in the available rules list to select it. Then click the **Link** button.

Gateway links the *Rule1* Decision Rule to the *Routing1* Rule Table.

- Click **OK** to validate the new Rule Table.

Gateway closes the *New Rule Table* window and adds the *Routing1* entry to the display of Rule Tables and Decision Rules for Transfer Change State category.

Results

Once you have created and activated the Decision Rule and linked it to a Rule Table, the supervisor component of Gateway monitors gateway activity, analyzing incoming transfers for matches to the Decision Rule.

Incoming Transfer match

Suppose that Gateway receives an incoming transfer with the following parameter values.

- Direction: 'I' (incoming)
- Application: 'APPLI'
- Destination: GATEWAY

- Originator: TRANSFER
- Protocol: 'PeSIT'
- Record length: 1024
- File component: file.txt
- File size: 147
- State: Ended

Gateway checks the parameters of the incoming Transfer against the conditions set in the *Rule1* Decision Rule.

- Condition 1: protocol is PeSIT
- Condition 2: destination (Gateway) is not the same as the originator (Axway Transfer)
- Condition 3: record length is between 150 and 3000
- Condition 4: direction is incoming
- Condition 5: file size is not between 10 and 50
- Condition 6: the incoming file transfer is complete and passed to the *ended* state

Routing result

Since the six conditions specified in the *Rule1* Decision Rule match the parameters of the incoming Transfer, Gateway applies the Model MODEL_FTP definition. After receiving the file from Axway Transfer, Gateway immediately deposits a new Transfer Request and makes the file available to the FTP client.

Axway Gateway: Managing Events

Decision Rule usage example: Directory Scanning

[Task overview](#)

[Example production environment](#)

[Configuring the function](#)

[Results](#)

Task overview

This topic provides an example of how to configure Gateway to scan a directory for a certain type of file.

You can create a Directory Scanning type Decision Rule so that Gateway monitors a specified folder on the host machine. You can then specify an action for Gateway to execute when it detects a file in this folder.

To configure Gateway to execute a process based on the presence of a file in a specified folder, perform the following tasks:

- [Select storage and work folders for file monitoring by Gateway](#)
- [Create execution Models for transferring the deposited files to remote machines](#)
- [Create Directory Scanning type Decision Rules](#)
- [Create a Directory Scanning type Rule Table, and link the Decision Rules to the Table](#)

Example production environment

Operation

A company handles billing records from diverse machines on its network. Several billing terminals regularly send billing records to an accounting application on a centralized UNIX machine. The accounting application analyzes each incoming billing record and creates two types of report:

- `IntlReportNxxx.txt` - concerning international accounts. These files are to be transferred and stored on SiteA.
- `NtlReportNxxx.txt` - concerning national accounts. These files are to be transferred and stored on SiteB.

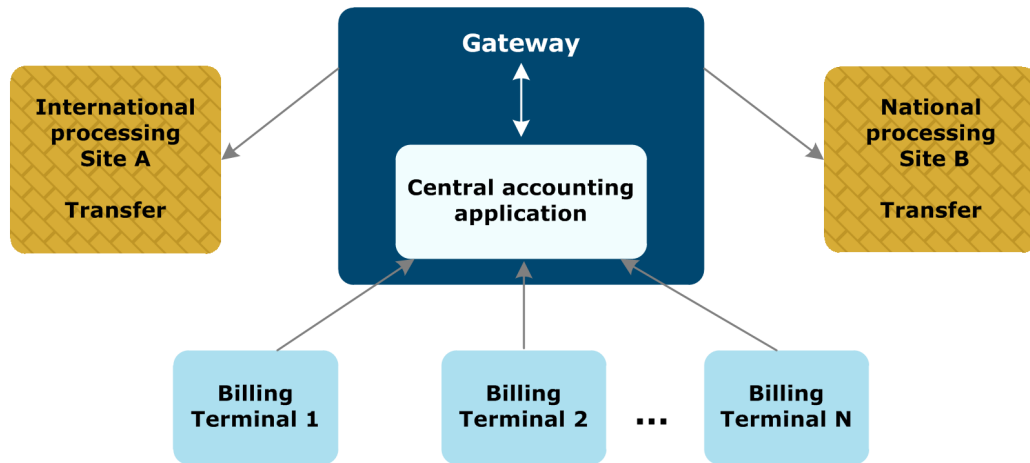
When the accounting application creates the reports it deposits them in the local folder: `/usr/local/appli/report`. When deposited to this folder, the files are in their final form and their contents are no longer modified.

Architecture

In this example:

- The central accounting application and Gateway reside on the same UNIX host
- Both International handling SiteA and National handling SiteB host Axway Transfer applications (Transfer CFT or Transfer InterPel)
- Billing terminals communicate with the central accounting application independently of Gateway

The following figure illustrates the architecture of the example.



Configuring the function

Prerequisites

- Gateway must be correctly installed and running on the same platform as the central accounting application.
- Axway Transfer applications must be correctly installed and running on SiteA and SiteB machines.
- You must have created the local and remote Sites necessary to enable transfers from the Gateway to the Axway Transfer sites. This example uses the following site aliases:
 - Gateway local site alias = GW_1
 - International processing server remote alias = SiteA
 - National processing server remote alias = SiteB
- The machines must be linked by a TCP/IP connection.
- You require a Windows environment to support the Gateway GUI.

Selecting the storage and work folder for Gateway monitoring

In this example, the folder we want to monitor is the folder that the central accounting application uses for report deposits:

`/usr/local/appli/report.`

Create the working folder in which Gateway processes files. In this example:

`/usr/local/appli/work.`

Note: The working folder and the watch folder must be on the same file system.

Creating the execution Models

In this example, we create Decision Rules that specify a Model as the execution type when a stored file is detected. We require two General Models: one for Transfers to the international processing server, and another for Transfers to the national processing server.

The [General Model](#) is mandatory for Transfers, since you can only define the Transfer destination in the General Model and not in a Protocol Model.

To create a new General Model for Transfers to the international processing server:

1. In the Directory tree structure, click  to expand the nodes:
Transfer Management > Parameters > Model
2. Right-click the **Transfer Model** sub-node, then select **New** from the context menu to display the *New General Model* window.
3. Complete the fields of the *New General Model* window as follows.
 - In the **Name** field, enter: *M_Trans_Intl_Report*.
 - Use the **Argument** and **Value** fields to assign the following values:

Argument	Value
type	TRANS
direction	0 (output)
mode	I (initiator)
file name	&(file_name)
file component	&(dir_path)/&(file_component)
application	DEFAULT_B
originator alias	GW_1
destination alias	SITEA

4. After entering the values, click **OK**.
Gateway closes the *New General Model* window and adds an entry representing the new Model to the list in the *Transfer Model* display pane.

To create a new General Model for Transfers to the national processing server:

1. In the Directory tree structure, click  to expand the nodes:
Transfer Management > Parameters > Model

- Right-click the **Transfer Model** sub-node, then select **New** from the context menu to display the *New General Model* window.
- Complete the fields of the *New General Model* window as follows:
 - In the **Name** field, enter: *M_Trans_Ntl_Report*.
 - Use the **Argument** and **Value** fields to assign the following values:

Argument	Value
type	TRANS
direction	0 (output)
mode	I (initiator)
file name	&(file_name)
file component	&(dir_path)/&(file_component)
application	DEFAULT_B
originator alias	GW_1
destination alias	SITEB

- After entering the values, click **OK**.

Gateway closes the *New General Model* window and adds an entry representing the new Model to the list in the *Transfer Model* display pane.

Creating the Directory Scanning type Decision Rules

We require two Decision Rules: one for Transfers to the international processing server, and another for Transfers to the national processing server.

To create a Decision Rule for Transfers to the International processing server:

- In the left pane of the GUI main window, click  to expand the **Event Management** node.

Gateway displays the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

- Right-click the **Directory Scanning** sub-node. A context menu appears. Select **New > Decision Rule...**

Gateway displays the *New Decision Rule* window.

- In the *New Decision Rule* window, enter the following data:

Field	Enter the value...	Description
Name	Trans_International_Report	Unique name for the Decision Rule.
Conditions		
Argument/Operator/Value	file name = IntlReportN*.txt	Name of the file to be transferred to the International processing server.
Processing (if all conditions match)		
Execution type	Model	Execute a Model in the case of a match for this Decision Rule.
Model	M_Trans_Intl_Report	Name of the Model to use to initiate a Transfer Request to the International processing server.

- After entering the values, click **OK**.

Gateway closes the *New Decision Rule* window and adds an entry representing the new Decision Rule to the list in the Decision Rule display pane.

To create a Decision Rule for Transfers to the national processing server:

- In the left pane of the GUI main window, click  to expand the **Event Management** node.

Gateway displays the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

- Right-click the **Directory Scanning** sub-node. A context menu appears. Select **New > Decision Rule...**

Gateway displays the *New Decision Rule* window.

- In the *New Decision Rule* window, enter the following data:

Field	Enter the value...	Description
Name	Trans_National_Report	Unique name for the Decision Rule.
Conditions		

Field	Enter the value...	Description
Argument/ Operator/ Value	file name = NtlReportN*.txt	Name of the file to be transferred to the International processing server.
Processing (if all conditions match)		
Execution type	Model	Execute a Model in the case of a match for this Decision Rule.
Model	M_Trans_Ntl_Report	Name of the Model to use to initiate a Transfer Request to the International processing server.

- After entering the values, click **OK**.

Gateway closes the *New Decision Rule* window and adds an entry representing the new Decision Rule to the list in the Decision Rule display pane.

Creating the Directory Scanning type Rule Table

For this example we create a Directory Scanning type Rule Table, linking the Decision Rules that we created in the previous section.

To create a Directory Scanning type Rule Table, follow these steps:

- In the left pane of the GUI main window, click  to expand the **Event Management** node.

Gateway displays the following sub-nodes:

- Scheduling
- Directory Scanning
- Transfer Change State
- XMS

- Right-click the **Directory Scanning** sub-node. A context menu appears. Select **New** then **Rule Table...**

Gateway displays a *New Rule Table* window.

3. In the **General** tab of the *New Rule Table* window, enter the following data:

Field	Enter the value...	Description
Name	Acct_file_test	Unique name for the Rule Table.
Comments	Table for accounting file transfer test	Enter a free text description of the Rule Table. Maximum: 80 alphanumeric characters.
Active	Select this option.	Activates the Rule Table.
Exclusive table	Select this option.	Executes only the first Decision Rule that completely matches the criteria.
Default execution type	No action	Specifies that no action is executed if no defined Decision Rule matches the current parameters.
Log level	None	Deactivate logging.

4. In the **File Event** tab of the *New Rule Table* window, enter the following values.

Field	Enter the value...	Description
Watch Directory	/usr/local/appli/report	Path to the directory you want Gateway to supervise (or poll) to determine the presence of a file.
Watch period	20	Time cycle (in seconds) for polling repetitions.
Work directory	/usr/local/appli/work	Path to the work directory. When Gateway polls the Watch Directory and locates a file, it transfers the file to the Work directory before beginning file processing.
Optimized cleaning	Select this option.	Enables Gateway to periodically delete a part of the contents of the Work Directory contents.

5. In the **Linked rules** tab of the *New Rule Table* window, in the **Available Rules** list:

- Click *Trans_International_Report* to select it. Then click the **Link** button.
- Click *Trans_National_Report* to select it. Then click the **Link** button again. Gateway links the two Decision Rules to the *Acct_file_test* Rule Table.
- Click **OK** to validate the new Rule Table.

Gateway closes the *New Rule Table* window and adds the *Acct_file_test* entry to the display of Rule Tables and Decision Rules for the Directory Scanning category.

Results

When you have created and activated the Decision Rules and linked them to a Rule Table as described above, Gateway scans the `/usr/local/appli/report` folder every 20 seconds for the presence of a file.

The presence of one or more files triggers the copy of the file(s) to the work folder. Gateway then extracts the information necessary to analyze the Rule Table for Decision Rule criteria matches. If Gateway finds a match, it deposits the appropriate Transfer Model and transfers the file to the specified server.

After execution of the Transfers, the original files are periodically purged from the work folder.

Axway Gateway: Managing Events

Decision Rule usage example: Scheduling

[Task overview](#)

[Example production environment](#)

[Configuring the function](#)

[Results](#)

Task overview

This topic provides an example of how to configure Gateway to trigger a specific processing task at a pre-scheduled time each week.

To configure Gateway to retrieve a file from a remote machine at pre-scheduled dates and times, perform the following tasks:

- [Create an execution Model for retrieving a specified file from a specified remote machine](#)
- [Create a Scheduling-type Decision Rule](#)
- [Create a Scheduling-type Rule Table, and link the Decision Rule to the Rule Table](#)

Example production environment

Operation

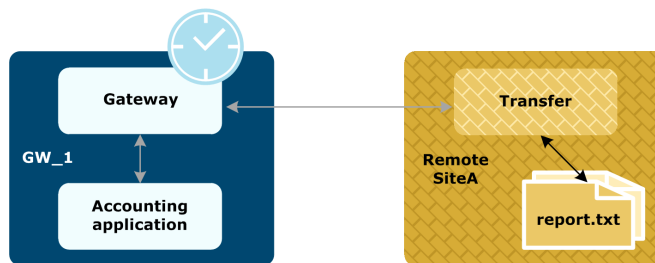
A company uses an accounting application that requires an expense report that is stored on a remote machine. The expense report is available every Monday at 3 p.m., except during the month of December.

Architecture

In this example:

- The accounting software and Gateway reside on the same UNIX host.
- The expense report directory resides on a machine with hostname *SiteA*. SiteA also hosts an Axway Transfer product.

The following figure illustrates the architecture of the example.



Configuring the function

Prerequisites

- Gateway must be correctly installed and running on the same platform as the accounting application.
- Axway Transfer must be correctly installed and running on the Remote SiteA machine.
- You must have created the local and remote Sites necessary for Transfer Request from Gateway to Axway Transfer. This example uses the following site aliases:
 - Gateway local site alias = GW_1
 - Remote machine alias = Remote_SiteA
- The machines must be linked by a TCP/IP connection.
- You require a Windows environment to support the Gateway GUI.

Creating the execution Model

In this example, we will create a Decision Rule that specifies a [Model](#) as the execution type when a stored file is detected.

To create a new General Model to retrieve a file from a remote server:

1. In the Directory tree structure, click  to expand the nodes:
Transfer Management > Parameters > Model
2. Right-click the **Transfer Model** sub-node, then select **New** from the context menu to display the *New Transfer Model* window.
3. Complete the following fields of the *New Transfer Model* window:
 - In the **Name** field, enter *M_Retrieve_Report*.
 - In the **Protocol** field, select *General*.
 - Use the **Argument** and **Value** fields to assign the following values:

Argument	Value
type	TRANS
direction	I (input)
mode	I (initiator)
file name	report.txt
file component	report.txt
application	DEFAULT_B
originator alias	REMOTE_SITEA
destination alias	GW_1

4. Click **OK** to validate the values.
Gateway closes the *New General Model* window and adds an entry representing the new Model to the General Models list in the General Model display pane.

Creating the Scheduling-type Decision Rule

To create a Decision Rule for a scheduled task:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following four sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS
2. Right-click the **Scheduling** sub-node. A context menu appears. Select **New > Decision Rule**. Gateway displays a *New Decision Rule* window.
3. In the *New Decision Rule* window, enter the following data:

In the section...	Field	Enter the value...	Description
Top	Name	Retrieve_Report	Unique name for the Decision Rule.
	Active	Click the check box.	Activates the Decision Rule.
Conditions	No conditions.		
Processing (if all conditions match)	Execution type	Model	Execute a Model in the case of a match for this Decision Rule.
	Model	M_Retrieve_Report	Name of the Model to execute.

4. After entering the values, click **OK**.

Creating the Scheduling-type Rule Table

For this example we create a Scheduling-type Rule Table, linking the Decision Rule that we created in the previous section.

To create a Scheduling-type Rule Table, follow these steps:

1. In the left pane of the GUI main window, click  to expand the **Event Management** node. Gateway expands the node to display the following four sub-nodes:
 - Scheduling
 - Directory Scanning
 - Transfer Change State
 - XMS
2. Right-click the **Scheduling** sub-node. A context menu appears. Select **New** then **Rule Table....**

Gateway displays a *New Rule Table* window.

3. In the **General** tab of the *New Rule Table* window, enter the following data:

Field	Enter the value...	Description
Name	Retrieve_file_test	Unique name for the Rule Table.
Comments	Table for scheduled file retrieval test	Free text description of the Rule Table. Enter up to 80 alphanumeric characters.
Exclusive table	Click the check box to indicate yes	Executes only the first Decision Rule that completely matches the criteria.
Default execution type	No action	Specifies that no action is executed if no defined Decision Rule matches the current parameters.
Log level	None	Deactivate logging.

4. In the **Scheduling** tab of the *New Rule Table* window click **New**.

Gateway opens the *Scheduling calendar* window.

- Use the fields in the Selected Date and Selected Hour sections to set a time and date for repeated triggering of a task. We want to trigger at 3 p.m. on all Mondays, excluding Mondays during the month of December.
- In the *Selected Date* section, click the lower radio button, then click the lower text box and select **add** from the drop-down list.
 - Gateway displays a new text box to the right of the first box. Click the newly displayed text box and select **all** from the drop-down list.
 - Gateway displays a new text box to the right of the second box. Click the newly displayed text box and select **monday** from the drop-down list.
 - Gateway displays a new text box to the right of the third box. Click the newly displayed text box and select the period (.) from the drop-down list. Reading the boxes from left to right, the boxes should now display: **add all monday**.
 - Click **Add**. Gateway adds the *add all monday* date set to the list of triggering dates in the Date frame.
- In the text box of the *Selected Hour* section, enter *15H00*. Then click **Add**. Gateway adds the *15:00 (3 p.m.)* time to the list of triggering times in the Date frame.
- In the *Selected Date* section, click the lower radio button, then click the lower text box and select **remove** from the drop-down list.
 - Gateway displays a new text box to the right of the first box. Click the newly displayed text box and select **all** from the drop-down list.
 - Gateway displays a new text box to the right of the second box. Click the newly displayed text box and select **monday** from the drop-down list.

- Gateway displays a new text box to the right of the third box. Click the newly displayed text box and select **of december** from the drop-down list.
 - Gateway displays a new text box to the right of the forth box. Click the newly displayed text box and select the period (.) from the drop-down list. Reading the boxes from left to right, the boxes should now display: ***remove all monday of december.***
 - Click **Add**. Gateway adds the *remove all monday of december* date set to the list of triggering dates in the Date frame.
 - Click **OK** at the bottom of the calendar window to accept the schedule and return to the **Scheduling** tab. This tab now displays your scheduled dates and time, labeled as *Calendar #1*.
5. In the **Linked Rules** tab of the *New Rule Table* window, in the Available Rules list click *Retrieve_Report* to select it. Then click the **Link** button.
Gateway links the *Retrieve Report* Decision Rule to the *Retrieve_file_test* Rule Table.
6. Click **OK** to validate the new Rule Table.
Gateway closes the New Rule Table window and adds the *Retrieve_file_test* entry to the display of Rule Tables and Decision Rules for the Scheduling category.

Results

When you have created and activated the *Retrieve_file_test* Decision Rule and linked it to the *Retrieve_file_test* Rule Table as described above, Gateway triggers the retrieval of the `report.txt` file from Remote SiteA at 3 p.m. on the scheduled dates.

Managing security

To preserve your data from hackers, the information previously available in the User guide, relative to Managing security has now been moved to the [Gateway Security guide > Security administration](#) (requires a login to read online).

Axway Gateway: Protocols

Protocols

About Axway Gateway protocols

Gateway enables automatic and secured file transfer via many different supporting protocols. The protocol you use depends on your network architecture, your transfer partners and your file transfer requirements.

Gateway enables you to transfer files according to normalized public protocols: PeSIT Hors SIT, OFTP (Odette), FTP, EDIINT and RosettaNet. It also implements a private protocol, PEL.

Additionally, Gateway can exchange files with platforms installed with other file transfer products if they supply the same public protocol stack.

Supported protocols

[FTP](#)

[HTTP](#)

[OFTP \(Odette\)](#)

[PEL](#)

[PeSIT Hors SIT](#)

[SFTP](#)

[SMTP/POP3](#)

[EDIINT](#)

[RosettaNet](#)

[About SWIFTNet](#)

[About Axway Gateway connectors](#)

Axway Gateway: Protocols

FTP

About FTP

[FTP](#)

[FTP and Gateway](#)

FTP introduction

FTP (File Transfer Protocol), specified in RFC 959 (October 1985), is an end-to-end protocol designed to transfer files or data over a TCP/IP network between two platforms irrespective of the hardware and software architecture of the host computers. FTP includes functions to log on to the network, list directories and copy files. Data is transferred over a data-type connection that is independent of the connection used to submit commands.

How does FTP work?

FTP uses two TCP/IP sessions for a transfer:

- The **control session** to send and receive FTP commands
- The **data session** to send and receive file data

The client always initiates the control session. The client submits its identification and then sends the commands that determine the data transfer behavior to implement, as summarized in the following table:

Data Transfer attribute	Description
Active or passive role	The data session is initiated according to the server role defined in the control session: • In active mode the server initiates the data session • In passive mode the client initiates the data session The mode is independent of the transfer direction.
Transfer direction	Send or Receive
Data format (type)	ASCII, EBCDIC or Binary
Data exchange (transmission) mode	STREAM, BLOCK, or COMPRESSED
Data structure	File or Record
Name of the file to be transferred	File name

The data connection is used to:

- Transfer data files (for example, using the **RETR** or **STOR** commands)
- Send a directory list in response to a **LIST** or **NLST** command

The following diagram gives a simple overview of the two sessions.



FTP control and data sessions

To end an FTP session, the client sends an FTP `QUIT` command to the server, which then releases the connection.

FTP and Gateway

You can use FTP in Gateway to:

- Transmit and receive files in client and server mode
- Compress data using methods including *stream*, *block* and *compressed*
- Transform data using ASCII, EBCDIC, and IMAGE (Binary) encoding
- List available files or transfers, via the DIR command
- Transfer multiple files (MPUT and MGET)

Gateway enables you to implement routing, send pre- and post-transfer commands, and launch user exits when sending or receiving transfers.

[Configuring the FTP environment](#)

[General FTP command attributes](#)

[Using FTP in Initiator mode](#)

[Using FTP in Responder mode](#)

[Using an FTP proxy](#)

[Monitoring an FTP Transfer Request](#)

[Transferring multiple files with FTP \(MPUT and MGET\)](#)

[FTP specific user exits](#)

[Executing a basic FTP transfer](#)

Axway Gateway: Protocols

FTP: Configuring the environment

[About FTP configuration](#)

[Creating an Application](#)

[Creating a Site object](#)

[Configuring FTP security](#)

About FTP configuration

Before creating a transfer, you must define specific FTP Applications and Sites if they do not already exist. At least one of the each of the following objects must be defined:

- **Application**

A default Application object is supplied for each data code: FTP_A, FTP_B, FTP_E.

- **Local Site**

Gateway creates a default Local Site during installation, that corresponds to the alias name and protocol identifier supplied at that time. It is the default Local Site for all the protocols supplied by the product, including FTP.

- **Remote Site**

Define your partner Site and communication attributes in the Remote Site object. You can use a default Template Site, TFTP, to create a dynamic site.

Create these objects using the GUI or the `pe1adm` online command.

Creating an Application

The Application object manages transfer and file attributes such as record format and padding.

Gateway provides three default Applications for FTP transfers. You must use a specific Application for each FTP type:

- FTP_A: for ASCII type files
- FTP_B: for IMAGE type files (same as BINARY)
- FTP_E: for EBCDIC type files

Application parameters

Refer to [Working with Applications](#) for detailed information about Application parameters.

Creating a Site object

A transfer begins with the physical connection of two Sites. In Gateway, the terms Initiator and Responder are used in place of the standard FTP terms Client and Server. The Initiator Site sends the connection request to which the Responder Site responds.

During the connection phase, the Initiator sends its identification. If the Responder is an Axway MFT product, it checks the Initiator identification in the Remote Site attributes.

The FTP protocol allows data transfer in both directions: the Site sending the data is the Sender, the other is the Receiver. A given Site can have the following roles:

- Initiator/Sender
- Initiator/Receiver
- Responder/Sender
- Responder/Receiver

The table below summarizes the possible roles for the Remote Site.

Action you want to perform:	The Remote Site represents:	Then it is defined as:
Send a file from your Local Site	Destination	Receiver
Receive a file	Partner	Sender

The parameters representing the mode and action of the Remote Site (`-init_sender`, `-init_receiver`, `-resp_sender`, `-resp_receiver`) are all set to Yes (Y) by default when you use command lines.

For each Axway MFT product, you can define as many Local and Remote Sites as needed, each referenced by a unique protocol identifier. The Local Site is defined by the **Partner local alias** (`-partner_loc_alias`) attribute in the Remote Site object, or by the default Local Site name (defined during the installation).

You can use the same Site for simultaneous file transfers in both directions.

Defining Remote Site parameters

Use the Remote Site to:

- Provide the network address and SAP (port) of the partner
- Select the method to use for the identification phase, and information required for this method (for example, user name and password)
- Obtain the Application selection method if it is not defined in the Transfer Request

Remote Site parameters

Refer to [Site objects: Parameters List](#) for details of the parameters to set when creating a Remote Site.

Defaults

If you do not define a parameter, Gateway uses default values from the operator interface or from the internal default attributes.

If you do not specify a group when creating a Remote Site, Gateway assigns the group **GDDEFAULT** to the Site by default. The group name of the Remote Site and Application must be the same.

Example

```
peladm create_site -alias ftppartner -protocol FTP
               -comments "FTP Remote Site" -ct TCPIP
               -dest_address hostname/21 -l_id username -l_pswd password
```

Anonymous Site

In the server database, you can define a Remote Site object without a password, named **ANONYMOUS**. This object accepts the connection from any incoming host, regardless of the password provided. The server implementation of FTP asks the Remote Site for a password and traces it in the log file before allowing the connection.

Configuring FTP security

Using FTP with SSL/TLS

You can configure Gateway to handle incoming connections in either *implicit* or *explicit* mode. You can also implement FTP security in a Gateway client in either of these modes. By default, Gateway uses TLS in implicit mode.

Implicit mode:

- Connection occurs on a secure port
- Control session opens directly in SSL or TLS

Explicit mode:

- Connection occurs on an unencrypted port
- Decision to launch the SSL/TLS layer is negotiated once the session is established

Set the following three parameters in the TLS Profile (**sprof**) to define the connection mode:

TLS Profile parameter	Description
Connection securing mode <code>-ft_sec_option</code>	Defines whether to implement security implicitly or explicitly. The possible values are: • implicit (default) • explicit
Global session policy <code>-ft_session_policy</code>	Determines whether or not the control session must be secured but are only available in explicit mode. Prohibiting security is equivalent to deactivating it. The possible values are: • mandatory • optional (default)
FTP data session policy <code>-ft_data_policy</code>	Defines whether the data sessions must be secured. The possible values are: • mandatory • optional (default) • disabled

Notes: The distinction between SSL and TLS is made during negotiation of the protocol version.

The cipher suites that you specify in the TLS Profile define the level of security.

[About FTP](#)

[Working with Local Sites](#)

[Working with Local Sites \(command line\)](#)

[Working with Remote Sites](#)

[Working with Remote Sites \(command line\)](#)

[Site objects: Parameters List](#)

[Working with Applications](#)

[Working with Applications \(command line\)](#)

[Executing a basic FTP transfer](#)

General FTP command attributes

General FTP commands

This topic introduces the attributes for general FTP commands. This command information applies to submitting a Transfer Request in either client or server mode.

Using FTP commands to define session attributes

To transfer a file, you must define the session attributes. The three standard commands for setting FTP session attributes are:

FTP Command	Options	Comments
TYPE Use this command to define the transfer mode (data conversion mode for transmission)	S (tream) (default)	Gateway establishes a new data connection for each transfer
	B (lock)	In both of these modes, Gateway keeps the data connection open, depending on the FTP response code that the Responder sends at the end of each transfer
	C (ompressed)	
MODE Use this command to define data code	A (SCII)	In virtual directories you can access ASCII Transfer Requests using the existing <i>FTP_A</i> and <i>FTP_E</i> Applications for local text files in ASCII
	E (BCDIC)	In virtual directories you can access EBCDIC Transfer Requests Gateway automatically converts EBCDIC (network) to ASCII (local)
	I (MAGE)	In virtual directories you can access binary, ASCII, and EBCDIC Transfer Requests using the existing <i>FTP_B</i> for IMAGE type files
STRU = file structure Use this command to define data representation	F (ile) (default)	File is a continuous sequence of data bytes
	R (ecord structure)	File consists of sequential records

Note: PAGE structure representation is not available in Gateway.

Summary of session attributes

The way that data is encoded depends on how you combine commands, as shown in the following table. FTP commands that define session attributes include the following:

FTP command	Gateway command parameters	Define this parameter in
MODE	-data_code (-dc)	Application (<i>Data code</i>) Transfer Request (<i>Data code</i>)
TYPE	-trans_mode (-trm)	Remote Site (<i>Transfer mode</i>)
	-rec_fmt (-rf) Values are: • F (Fixed) • FT (Fixed Text) • V (Variable) • VT (Variable Text) • S (Stream)	Transfer Request (<i>Record format</i>)
STRU	-ftp_data_struct	Remote Site (<i>Data structure</i>)

Using the SYST command

To identify a Gateway server, an FTP client can send the **SYST** command, and recognizes a Gateway server by its response to this command.

A Gateway server responds to a **SYST** command by:

```
"215 <UNIX XFB/<Operating_System_type>"
```

For example:

```
"UNIX XFB/UNIX" or "UNIX XFB/NT"
```

Many applications or Web browsers work better when the FTP server is a UNIX system. Therefore, the UNIX system has been chosen as representative of all implementations of the FTP protocol, irrespective of the actual operating system.

The first UNIX label specifies the format to use with the **LIST** command. The last one gives the real operating system type.

Note: When a Gateway FTP client receives an Axway MFT system type in response to a **SYST** command, it sends specific information using a **SITE** command before transferring a file.

Using the SITE command

To send specific commands to a Gateway FTP server, you can use the **SITE** command.

The syntax of the **SITE** command is:

```
SITE <sp> <Command_1> <sp> <Param_1>[;...;<Command_n> <sp> <Param_n>]
```

The following table lists available **SITE** commands:

SITE command	Function
P_CLIENT	Client Type
P_ORG	Originator Site name
P_DEST	Destination Site name (can be used for routing a transfer)
P_RECFTM	Format or record (F (Fixed) or V (Variable))
P_LRECL	Logical record length in bytes
P_APPLI	Name of Application to use
P_MSG	User message
FTPCS	FTPCS client version number. Prompts the Axway MFT FTP server to return the whole list of authorized commands/parameters.

If one of these special commands was already sent in the Transfer Request, only the other commands are sent to the server.

The FTP client can use a **SITE P_APPLI** command to use a specific Application on the server Site. An Application that you specify via the **SITE P_APPLI** command takes priority over one defined in the Remote Site or Transfer Request.

Note: Remote Applications must have the same `xfer_data_code` (transmission mode) value as in the current FTP session attribute (ASCII, EBCDIC, IMAGE).

Pre- and post-transfer end response timers

In **STREAM** mode, the Gateway FTP server can never guarantee that an incoming transfer is finished gracefully. The only way it can detect an error during such a transfer, and so avoid routing the file if it is required, is to receive an **ABORT** command.

However, the reception of an **ABORT** command can occur before or after the transfer is finished (for example, in **STREAM** mode, the data session is closed). To detect as many error conditions as possible, the following response timers are available in Responder/Receiver mode for **STREAM** mode / **FILE** structure sessions:

- **pre_resp_timer**: indicates the waiting period allowed between detecting the data closure and sending the transfer end response (226 Requested file action successful) to the client.
- **post_resp_timer**: indicates the waiting period allowed between sending the transfer end response and validating the transfer.

You can designate the waiting period for these two timers in the configuration file, with the `pre_resp_timer` and `post_resp_timer` fields of the `FT_FTP` entity. Any timer set to 0 is ignored, that is, it is not started, and the process goes directly to the next step.

If the Gateway Responder receives an **ABORT** command or a control session closure during one of these allowed waiting periods, it aborts the transfer. If it receives any command other than **ABORT**, it stops both timers and validates the transfer.

FTP transfer scenarios

Since Gateway *must* be able to detect that a transfer is invalid to avoid routing it, clients that do not send **ABORT** commands should not use Gateway to route transfers. Many FTP clients never send **ABORT** commands, even if the user explicitly requests a transfer abort. However, when a client sends a file to Gateway in `STREAM` mode and `FILE` structure, there is no way for the server to detect that the partner aborted the transfer.

FTP commands for directory management

To manage files on a Gateway FTP server, you can use the following standard FTP commands:

- `MKD` (make directory)
- `RMD` (remove directory) you can only delete empty directories
- `DELE` (delete file)
- `RNFR` and `RNTO` (rename file)

You must have specific rights to use these commands. In addition, these commands only apply to Real File Directories (RFD).

[Using FTP in Initiator mode](#)

[Using FTP in Responder mode](#)

Axway Gateway: Protocols

Using FTP in Initiator mode

After [configuring the Remote Site and Application for FTP](#) you can submit a Transfer Request.

Using an Application with the FTP protocol

Initiator/Sender

If a remote FTP client, the Initiator, needs to select an Application for a file transfer, you can transmit an Application name with an FTP **SITE** command using the following syntax:

```
SITE P_APPLI appli_name
```

If you are using Gateway as an FTP client, it uses the FTP **SYST** command to detect whether or not the remote FTP server is Transfer InterPel or Gateway. In this case, the **SITE P_APPLI <appli_name>** FTP command is always sent.

This command is specific to Axway MFT, and can only be used with other Axway MFT FTP servers. In addition, this command is only taken into account if the **Application method** (`get_appli_method`) field of the Gateway FTP server Remote Site is set to *User exit (AIE)*.

Using peltrans commands for an FTP file transfer

The following table is a partial list of Transfer Request parameters relevant to FTP in Initiator/ Sender and Initiator/ Receiver modes. The parameters described here are in addition to those discussed previously. For the complete list of parameters and their possible values refer to [Transfers: Parameters List](#).

Initiator/Sender mode: Transfer Request parameters

Keyword names (*)	Default	Use the parameter to:
-snd_msg (-sm) -snd_msg_file (-smf)		Enter a user message. Translated as a SITE P_MSG user_message .
-unique_filename (-uf)		Assign a unique file name on the server (STOU command is sent in FTP)
-dir_name (-dn)		Define the remote file location
-ftp_command (-ftpc)		Send FTP commands to remote servers using QUOTE or SITE command
-append (-app)	N	Append data files to a received file on the same server. For use only on the client side in sender mode.

(*) Abbreviated keyword names in parentheses.

Defining the remote location of a file

In Initiator mode, the `dir_name` parameter of a Transfer Request enables you to define the location of a remote file. This location is the destination of the file (for an outgoing transfer), or the remote path (for an incoming transfer).

In UNIX/Windows, this parameter generally indicates a path. However, since a remote FTP server does not necessarily manage files with a hierarchical file system, or file separators, the FTP server can only concatenate `dir_name` and `file_name` without further analysis of the two strings.

Example:

In this example, the following Transfer Request:

```
peltrans -mode I -direction O -dest_alias FTP_SERVER
        -appli FTP_B -file_component local_file
        -dir_name /dir1/ -file_name file
```

results in the FTP transfer command:

```
STOR /dir1/file
```

Note: If the trailing / character is omitted in the `dir_name` parameter, the transfer directory selected becomes:

```
STOR /dir1file
```

Using FTP commands for a transfer

You can define the FTP commands used to retrieve or send a file (**RETR**, **STOR**, **STOU**, **REST** and so on) via the Gateway GUI or Gateway online commands.

FTP commands that define a transfer

FTP command	Define in:	Use to:
SYST	None (automatic Axway MFT command)	Identify an Axway MFT product.
RETR (GET)	Transfer	Receive a transfer.
STOR (PUT)	Transfer	Send a transfer (a request to store the current record in an output file).
STOU	Transfer	Give the file a unique name.
REST	Remote Site	Enable restart of an interrupted transfer.
SIZE file name	None (automatic Axway MFT command)	Return the number of kilobytes transferred with the current transfer structure, mode and type to determine the restart point.

FTP command	Define in:	Use to:
REMOVEFILE	Transfer	Remove the transferred file from the remote server. The command takes no arguments.
RENAMEFILE	Transfer	Rename/move the transferred file on the remote server to the given argument. The command is used with a filename or file pathname argument.
LIST	Transfer	Retrieve directory listings in Long format. (*)
NLST	Transfer	Retrieve directory listings in Names format. (*)

(*) For a more precise listing, use the advanced queries and filters provided by Gateway.

Using the SITE or QUOTE command

Identifying commands in the transfer

Use the `-ftp_command` field in a Transfer Request to send a server-specific remote command or an unsupported command to the FTP server before or after transfer. **SITE** and **QUOTE** commands are supported as parameters of the `-ftp_command` field.

SITE command

An FTP client can use the FTP command **SITE** to send any server-specific command to a remote server.

The syntax using the `peltrans` command is:

`peltrans`

`-ftp_command "SITE <fct1> <parameter1>; SITE <fct2> <parameter2>"`

Before Gateway sends the file, it sends each specific function to the remote server.

Note: If you receive a return code other than 200 (command OK) or 202 (command not implemented) for a **SITE** command, Gateway stops the current transfer.

QUOTE command

Use the FTP command **QUOTE** to send a standard FTP command for it to be recognized by Gateway.

For example, Gateway cannot use **CWD** (Change Working Directory) as an FTP command by itself. If you need to change the current directory on the remote file system for a specified file transfer, you can use the following in the `ftp_command` field in the Transfer Request:

```
QUOTE CWD <dir>
```

Note: Receiving a return code greater than or equal to 400 for a `QUOTE` command automatically stops the current transfer.

Post-transfer SITE and QUOTE commands

`SITE` and `QUOTE` commands always indicate pre-transfer processing. To specify that a command must be processed *after* the transfer, use:

- `AE_QUOTE` or `AC_QUOTE` instead of `QUOTE`
- `AE_SITE` or `AC_SITE` instead of `SITE`

The `AE_` prefix is used when the transfer must remain *ENDED* if the command fails. With the `AC_` prefix, its state is set to *CANCELED* if an error occurs.

Note: The post-transfer command is not processed if the transfer fails (its state is not *ENDED*).

Example 1: Commands sent pre- and post-transfer

```
peltrans -da RFTP -ap UNIX -fn UNIX -fc File1.txt  
  
-ftpc QUOTE command1;AE_QUOTE command2
```

Gateway executes *command1* before the transfer begins and *command2* after the transfer ends. If *command2* fails, the transfer state remains *ENDED*.

Example 2: Command sent post-transfer

```
peltrans -da RFTP -ap UNIX -fn UNIX -fc File1.txt  
  
-ftpc AC_QUOTE command1
```

Gateway sends *command1* after the transfer ends. If *command1* fails, the transfer state is set to *CANCELED*.

Example 3: File renamed post-transfer

```
peltrans -da RFTP -ap UNIX -fn FILE1.TMP -fc File1.txt  
  
-ftpc AC_QUOTE RNFR FILE1.TMP;AC_QUOTE RNTD FILE1.OK
```

The file is renamed from `FILE1.TMP` to `FILE1.OK` after the transfer ends. If renaming fails, the transfer state is set to *CANCELED*.

Calling a script

You can place `SITE` and `QUOTE` commands in a script and call it using the following syntax:

```
peladm update_site -snd_msg "SCRIPT <script_site>"  
  
peltrans -ftp_command "SCRIPT <script_xfer>"
```

In a script file, you must write each command on a separate line or use a semi-colon (;) to separate commands.

Example script

```
QUOTE CWD /tmp

QUOTE CWD public

SITE <fct1> <param1>; SITE <fct2> <param2>

SITE <fct3> <param3>

AC_QUOTE RNFR old_name

AC_QUOTE RNTD new_name
```

Note: Gateway does not permit calling a script within a script.

Defining SITE and QUOTE commands in the Remote Site

To change directories or send additional commands each time you connect to a remote FTP server, use the Remote Site parameter `-snd_ msg`.

You can only use `QUOTE` and `SITE` commands in this field (not `AE_` and `AC_` commands).

Example

```
peladm update_site .. -snd_msg QUOTE CWD /tmp
```

This command allows you to move to the `/tmp` remote directory before starting the data transfers.

Sending a file with a unique file name command (STOU command)

Some servers (IIS FTP, for example) only support the standard RFC 959 format, which defines the `STOU` command without any parameters.

If you receive a 500 code in response to a `STOU <filename>` command, this means that the Gateway FTP client is attempting to send a `STOU` command with no arguments. Therefore the protocol file name of the transfer record is not changed.

Allocating file size (ALLO command)

If necessary, you can activate or deactivate the automatic **allocate request** option for reserving file space before you send a file in FTP. Use this feature, for example, if the rejection of your automatic allocation request causes Gateway to cancel your transfer. This parameter is active by default.

Set the `allocate_request` parameter in the configuration menu.

Configuring the ALLO command

`allocate_request = 0`: deactivates the `ALLO` command

`allocate_request = 1`: activates the `ALLO` command

Directory listing (*-type and dir_type parameters*)

Current versions of FTP (and HTTP) support the listing of directories in Initiator mode. To retrieve the contents of a given directory, Gateway requests a **DIR** type transfer. It saves the result in a file, just as for a standard download transfer. This file can then be processed in a post-transfer exit script.

This feature is useful primarily for retrieving files from a given directory when you do not know the exact file names. You can define a user script to parse the resulting directory listing file, and create a Transfer Request for each file name listed.

type and dir_type parameters

In FTP, the **DIR** transfer type indicates that you must send a **LIST** or **NLST** command instead of **RETR**. The online parameter (only used in Initiator/ Receiver mode) is `-type` and its value must be set to `dir`.

The online parameter `-dir_type` is used to define the level of list detail. You can set this parameter to one of the following values:

<code>-dir_type</code> value	Listing command sent	Disadvantage
names (default)	<code>NLST</code> Retrieves the most basic details. Each line in the file indicates the name of an item.	- No distinction between files and directories - No extra information is given. For example, the last modification date - Some servers do not send the whole list, or the complete relative path of each item
long	<code>LIST</code> Retrieves more detailed information than the <code>NLST</code> command. Its content depends on the server and is typically in a UNIX format.	More complicated to parse the resulting file to retrieve this information

dir_name and file_name

The `-dir_name` field specifies the directory to list.

The `-file_name` field is optional, and (for DIR transfers) no default value is set if it is left empty. If set, its purpose is to define a filter on file names returned by the server.

The Gateway FTP client appends `-file_name` to `-dir_name`, and sends the result in the Transfer Request. (In addition, HTTP adds the file separator `/` if it is not present at the end of `-dir_name`.) However, the server response to such a request is unpredictable as illustrated in the following example.

Example

```
peltrans -m I -dir I -oa FTP_SERVER -ty DIR  
-dt LONG -dn /users/shared/ -fn *.txt
```

In this example, the command `LIST /users/shared/*.txt` is sent to the FTP server. The server response may be any of the following:

- Send a list of `/users/shared` files whose names match the pattern `*.txt`
- Return an error
- Try to find the file whose name is exactly `*.txt`

Note: The final `/` character in the `-dir_name` field is mandatory for FTP since file separators are not specified. As a result, Gateway can only concatenate `dir_name` and `file_name` without further analysis of the two strings.

Application

An Application is not mandatory for a DIR Transfer Request. If you do not specify an Application, the protocol supplies default parameters equivalent to `FTP_A` for FTP.

If you specify an Application, the corresponding encoding parameters (file attributes and transfer attributes) must match the default parameters. Otherwise Gateway processing is unpredictable.

Identifying an Axway MFT product (SYST command)

In Initiator mode, Gateway uses the `SYST` command to identify whether or not the partner Site is an Axway MFT product.

In response to an FTP `SYST` command, an Axway MFT product sends the following string:

```
"215 <OS XFB/<Operating_System_type>"
```

Example

```
UNIX XFB/UNIX
```

[Transfers: Parameters List](#)

Using FTP in Responder mode

FTP transfers in Responder mode

After [configuring the Remote Site and Application for FTP](#) you can submit a Transfer Request.

There are certain parameters that Gateway only uses in Responder mode. The following parameters are set on the Responder side, or can be sent to Gateway by standard FTP clients communicating with a Gateway server.

Using an Application

Responder/Receiver mode

The function used to transmit the Application name is defined in the **Application method** (`get_appli_method`) field of the Remote Site object. Three methods are available for the Responder to retrieve the Application name for an incoming transfer:

Method	Element used by FTP server as Application name
File name (AIF)	File name
User message field (AIM)	<i>User message</i> field, that the client provides in the <code>SITE P_MSG</code> command argument. If this command is not provided before the transfer command (<code>STOR</code> or <code>STOU</code>), the Application name remains void and the transfer is rejected.
User exit (AIE)	The corresponding exit function. If it is not provided, the FTP client can force its value by sending the <code>SITE P_APPLI</code> command. If this command is not provided before the transfer command (<code>STOR</code> or <code>STOU</code>), the FTP server uses the default Application name for the current session type: • FTP_A for an ASCII session • FTP_E for an EBCDIC session • FTP_B for a binary (image) session

The resulting Application name must correspond to an existing Application and must not be void. Otherwise the Transfer Request is rejected.

Responder/Sender mode

When you define an Application in Responder/Sender mode, the transfer characteristics must match the FTP session attributes asked for by the Remote Site. Otherwise the Receiver cannot open the file.

Transferring files to and from virtual directories

In virtual directories, Gateway only provides a logical view of a file. Since the physical files stored on the product host machine are referenced in the corresponding Transfer Request, they cannot be accessed directly. Therefore, retrieving or sending a file depends on a successful Transfer Request to Gateway.

An FTP transfer is identified by its protocol name. For security reasons, the protocol file names managed by Gateway are logical file names which FTP clients are only able to view and browse.

Sending a file

The Transfer Request submitted to Gateway defines the logical file name (protocol file name) and physical file name (file component).

Receiving a file

The client defines the logical file name and Gateway automatically assigns the physical file name.

Notes:

- In real directories, the protocol file name is its physical name.
- You can influence the FIFO behavior for the **RETR**, **STOR**, **MDTM** and **SIZE** FTP commands when they are used in combination with file name duplication (if the VFD allows it) with the `ftp_vfd_fifo` parameter. Refer to [Advanced parameters and Trace levels](#).

FTP file transfer: peltrans command parameters

The following table is a partial list of Transfer Request parameters and their default values. The parameters described here are either specific FTP parameters, or parameters that have a particular meaning when used in an FTP transfer. For the complete list of parameters and their possible values refer to [Transfers: Parameters List](#).

Responder/Sender mode: Transfer Request parameters

Keyword names (*)	Default	Use the parameter to:
<code>-route_from_xfer</code> (<code>-rfx</code>)		Activate a link between the incoming transfer and the outgoing one.

Keyword names (*)	Default	Use the parameter to:
<code>-mode (-m)</code>		Define the transmission mode: R in Responder/Sender mode.
<code>-direction (-dir)</code>		Define the transmission direction: O in Responder/Sender mode.
<code>-file_name (-fn)</code> <code>-file_component (-fc)</code>		Define protocol file name and to select the physical file to send (mandatory fields).
<code>-permanent (-perm)</code>	N	Set Transfer Request availability to permanent.

(*) Abbreviated keyword names in parentheses

Storing and retrieving FTP transferred files

Virtual File Directory (VFD)

Gateway always uses the VFD in FTP Responder mode. In virtual directories, the real file name is never used for:

- `STOR`, `STOU` commands on the client side
- `RETR` on the server side

Consequently, you must specify the **protocol file name** to send files in both Initiator and Responder modes.

The protocol file name provided in LIST format in virtual directories, is `sXXX.filename`, where:

- `XXX` is the `xfer_ident` field of the transfer
- `filename` is the protocol file name given in either the Transfer Request or the `STOR` or `STOU` command

The Mailbox entries are available either through this name, or just the protocol file name.

Gateway file filter and list query

As a virtual directory and a file name are sometimes insufficient to describe a Mailbox entry, Gateway provides an advanced file filter feature for use in `RETR`, `STOR`, `STOU`, `MDTM`, `SIZE` and `LIST` commands. This filter replaces the file name in these commands.

For example, the command:

```
RETR filename
```

can be replaced with:

```
RETR name=filename
```

A file filter is a string containing field=value pairs, separated either by space or "&" characters. Accepted values for fields are given in the following table:

Field	Accepted values
name	A regular expression that matches paths and protocol file names, or an <i>xfer_ident</i> format (<i>s<xfer_ident>.<proto_filename></i>).
direction	IN, OUT, or BOTH (not case-sensitive)
date	The date of the transfer, in the following format: <i>YYYYMMDDhhmmss</i> • Date fields (year, month, day of month) can be separated by a "/" character. • Time fields (hours, minutes, and seconds) can be separated by ":" characters. Time can be provided without a date. The current day is then considered as the default date.
end_before or endbefore	All transfers requested or performed later than the given date, are not selected. The value is in date format.
start_after or startafter	All transfers requested or performed earlier than the given date, are not selected. The value is in date format.
origin or org	Originator Site
destination or dest	Destination Site
state	List of states in Gateway format. Each selected Transfer Request has one of the given states.
mvs_state or mvsstate	List of states in MVS format. Each selected Transfer Request has one of the given states.
ftpcs_state or ftpcsstate	List of states in Axway EndPoint FTPCS format. Each selected Transfer Request has one of the given states. The FTPCS state is the concatenation of the state in Gateway format and the state in InterPel MVS format.
local_ident or localident	A long integer
xfer_ident or xferident	Transfer Identifier

Examples

```
RETR name=dir1/file1&org=sitel&start_after=20011205
```

```
LIST name=../*.txt state=IC org=nova
```

Note that:

- For multiple responses, for every command, except `LIST` where only one entry is required, the most recent command will be kept.
- Only one list of states can be given: **state**, **mvs_state**, or **ftpcs_state**.
- For real entries and directories, the fields **origin**, **destination**, **direction**, **state** and **mvs_state**, are ignored. In addition, **date**, **end_before**, and **start_after**, are compared to the last modification date.
- If some fields are inconsistent but syntactically correct, no syntax error is returned. However, the command cannot be executed because it does not match any entry. Therefore, it can provoke negative responses such as **File not found**.

Example

```
RETR name=file&state=C
```

In this example, the user tries to download the most recent entry called "file" and in the state "canceled". This is syntactically correct, but Gateway does not allow a canceled file to be downloaded and returns the following response:

```
550 File not found
```

Gateway list query reply format

Axway MFT products use several list formats in response to the `LIST` command:

- UNIX format
- NLST format (filenames only)
- Axway EndPoint FTPCS format
- Gateway user format

UNIX list format

To obtain the UNIX format, send a `LIST` command with a standard filter field.

The UNIX format is mapped as follows:

```
drwx----- X <origin_alias> <destination_alias> <size> <date> <time>  
<filename>
```

where:

Parameter	Denotes
drwx	Type and rights of the given entry
x	Mailbox status in accordance with the Gateway MVS format (or "-" for real entries and directories)
<origin_alias>	Originator Site alias (or --NA-- for real entries and directories)
<destination_alias>	Destination Site alias (or --NA-- for real entries and directories)
<size>	Size in bytes of the file to transfer
<date>	Transfer date in YYYY month day format (for example, 2006 Jan 20)
<time>	Transfer time in HH:MM format (for example, 18:30)
<filename>	<i>s<xfer_ident>.<filename></i> or <i><filename></i> (dependent on VFD file name display option) for virtual files <i><filename></i> for real files and directories

Example

```
-r----- 9          FTPL      RFTP  2006 jan 20  18:30  file1
```

A remote client can use the "?" and "*" wildcard characters for file selection.

Standard LIST or NLST commands only display transfers to start or to restart.

Gateway list format

The format for the Gateway file filter command is the following:

```
<local_ident> <direct> XYZ <origin_alias> <dest_alias> <size> <date>
<time>
```

where:

Parameter	Denotes
<filename><local_ident>	Transfer local identifier
<direct>	Direction SEND or RECV

Parameter	Denotes
X	Gateway transfer state
Y	State translated to Gateway MVS format
Z	Route state: E(empty route), T(to route), R(routed), A (acknowledged)
<origin_alias>	Site alias of the transfer origin (or --NA-- for real entries and directories)
<dest_alias>	Site alias of the transfer destination (or --NA-- for real entries and directories)
<size>	Size in bytes of the file to transfer
<date>	Transfer date expressed as YYYY month day (ex: 2006 Jan 20)
<time>	Transfer time expressed as HH:MM (ex 18:30)
<filename>	Protocol filename provided in the Transfer Request

Example

```
49 SEND E5 R65 TFTP 4 Dec 24 15:08 s49.UNIX
```

Axway EndPoint FTPCS format

Gateway provides the field `ftpcs_list_fields` to define, in server mode, the columns that can be sent in the `LIST` command to clients connecting to Gateway with Axway EndPoint FTPCS.

The command line syntax is:

```
peladm update_site <field name>,<field name>,<field name>
```

where the available field names are taken from the following table:

Field name	Description
LOCID ⁽²⁾	Transfer local identifier
GTYPE ⁽¹⁾ ⁽²⁾	VFD type (virtual directory, real directory, virtual file, real file)
RIGHTS ⁽¹⁾	VFD rights

Field name	Description
FILENAME ⁽¹⁾	Long name • <i>s</i><xfer_ident>.<filename> or <filename> (dependent on VFD file name display option) for virtual files • <filename> for real files and directories
SFILENAME ⁽¹⁾	Short name, <filename> only
APPLI	Application name
SIZE ⁽¹⁾	Size
DATE ⁽¹⁾	Transfer creation date or Latest Modification date
TYPE	MVS transfer state
RSTATE	Route state: E(empty route), T(to route), R(routed), A (acknowledged)
UMSG	User message
ORG	Originator
DEST	Destination
P1	Undefined parameter 1
P2	Undefined parameter 2
PERM	Permanent transfer indicator
DATA_CODE	B (Binary), T (Text)

When the item described in the listing is not a virtual file, transfer Mailbox, only these fields apply and are completed by the Gateway FTP server. Also, if not positioned, they are still sent to the client (in the last columns).

By default, `ftpcs_list_fields` contains all fields.

The list is unordered which means that, irrespective of the order in which you type the fields to be displayed, they are sent to the FTP CS client in a predefined order (as laid out in the table above).

Restarting a transfer (REST command)

Gateway can automatically restart transfers interrupted by incidents or by the user. In addition, before sending or receiving a file, an FTP client can indicate a restart by issuing the restart (`REST`) command followed by a `RETR`, `STOR`, or `STOU` command.

You must set the transfer restart option (*Restart allowed* attribute) in the Remote Site that represents the client partner. This parameter is only relevant to the Gateway server. Its behavior mainly depends on the transmission mode:

Transmission mode	Restart point	Restart process
BLOCK or COMPRESSED	Named synchronization points are sent with data. Gateway keeps these synchronization points in the transfer Mailbox record. (This means that you cannot restart transfers in real directories in BLOCK or COMPRESSED mode.)	On transfer restart, the partners negotiate the synchronization point from which to restart the transfer.
STREAM	No meta-data can be added to the data flow. Therefore the client determines the restart point.	If the client is: • Downloading the file: the size already received is known • Uploading the file, the client uses the <code>SIZE</code> command to retrieve the size already received by the server The client sends this restart offset to the server. Both partners initialize their file access in relation to the restart offset and the transfer can begin.

FTP defines behavior for the `SIZE` and `REST` commands. The `SIZE` command parameter is a file name, and the `REST` command accepts the restart offset. Offsets and file sizes must be expressed as a number of 8-bit bytes transferred.

The file size varies depending on the data representation type, for example, when data is encoded as ASCII (where line endings are CRLF) or IMAGE (where they are system-dependent: CR, CRLF, NL, or other values).

Most of the time, the only way to determine an FTP file size, or a physical restart offset from the `REST` argument, is to parse the file in order to evaluate the pair (`REST` offset, physical offset).

Note: In Responder mode, the `REST` and `SIZE` commands are not supported in `RECORD` structure. Since `SIZE` is usually used before `STREAM` mode restarts, it is not supported in `BLOCK` or `COMPRESSED` mode.

Receiving a file with a unique file name (STOU command)

The store unique (`STOU`) command is supported in Responder mode in both virtual and real directories. The Gateway FTP server handles this in the same way as the `STOR` command in virtual directories (Mailbox entries are unique by definition).

In real directories, `STOU` makes the server search for a unique file name in the given directory. If a file with the same name already exists, a `.1` suffix is added. If that file name exists already, the suffix is incremented by 1 (`.2`, `.3`, and so on) until a free entry is found.

Securing a server

Defending against hammer attacks

A hammer attack can occur when a user floods the server with requests until it can no longer respond to any other users. Gateway provides the following prevention methods against this type of attack:

- Transfer quotas
- Command sensitivity
- Maximum file upload size limits

You can set each of these restrictions either using the exit routine **ExitProtocolConnection** or the Gateway Site, CGate, and CGateGroup objects.

Transfer quotas

Gateway provides four quota parameters for you to set transfer upload or download limits within a designated time period.

The following table describes transfer quota fields. By default, the fields are set to 0, which is equivalent to deactivating the quota restriction.

Transfer quota	Definition	Defense if exceeded
<code>max_upload_rate</code>	Upload quota in KB per minute.	Drops all transfers, irrespective of the direction (upload or download) in which the quota was exceeded.
<code>max_download_rate</code>	Download quota in KB per minute.	Suspends transfer rights for the relevant direction.

Transfer quota	Definition	Defense if exceeded
<code>max_upload_requests_rate</code>	Maximum number of upload-type transfers that can be requested per minute.	Refuses client the right to transfer in the appropriate direction (upload or download) until it returns within the quota limit.
<code>max_download_requests_rate</code>	Maximum number of download-type transfers that can be requested per minute.	

Command sensitivity

The principle behind this mechanism of attributing a time-limited request credit to each client user is based on an implicit cost imputed to each command emitted (transfer, directory listing and so on), and disconnection when the available credit is exceeded.

The credit is represented by the level of sensitivity of your server to client requests. The more tolerant your machine is to client requests, the more credit is available. Set the sensitivity level using the **Command sensitivity** slide in the GUI or by setting the `command_sensitivity` parameter to an integer value between 0 and 5 (where 0 is no sensitivity and 5 is maximum sensitivity).

The Gateway FTP server manages client consumption of these credits for all quotas (transfer and command sensitivity) in one sliding window. The window size is set in the configuration menu **Quota window size** field (specified in minutes). The default is 0, that is, all defenses are disabled irrespective of the quota limits already set.

Example

If, for example, the following elements are set:

- **Quota window size** = 10 minutes
- `max_upload_requests_rate` = 8 uploads per minute

Then the Gateway FTP server verifies that the client does not exceed 80 uploads in **any 10-minute interval**. If the client sends a request that exceeds this limit, the server automatically activates the defenses, cutting all connections and refusing all further client access. These defenses remain active for the number of minutes necessary for the volume of uploads over the preceding 10-minute period to return to within the quota limit.

Maximum file upload size limit

Gateway refuses a transfer from a client if the file size in question exceeds the number of megabytes that you define using the `maximum_upload_file_size` parameter.

Monitoring User quotas

In the Gateway GUI, you can track user and quota consumption via the Connected User record. The interface displays:

- Number of uploads and downloads
- Volume of data transferred
- Percentage of `max_upload_rate` and `max_download_rate` quotas used
- Percentage of `max_upload_requests_rate` and `max_download_requests_rate` quotas used

Gateway also provides the online command `pelctl reset_connected_user` to:

- Reset recorded consumption to zero
- Delete the connection if it is no longer active

This command therefore serves as a mechanism to authorize a client to reconnect after being suspended.

Defending against bounce attacks

A bounce attack can occur when a client opens a data session between an FTP server (A) and a listening port on another machine (server B). The client then uses this session to send a file with commands that server B will understand. The result of this indirect communication is that it is generally extremely difficult to identify or trace the user responsible for the commands.

The IP address provided by the remote partner is compared with the remote address in the control session. If they are not the same, an error is returned, and logged.

You can set this option in the Remote Site and in the Proxy object (in the `change_data_addr` field). To reject data sessions with addresses different to those in the control sessions, set the `change_data_addr` parameter to *No*.

[Working with Remote Sites](#)

[Working with Remote Sites \(command line\)](#)

[Configuring protocols: FTP options](#)

[Transfers: Parameters List](#)

[Monitoring Connections](#)

[Monitoring Connections \(command line\)](#)

[Monitoring Connected Users](#)

[Monitoring Connected Users \(command line\)](#)

[Working with Proxies](#)

[Working with Proxies \(command line\)](#)

Using an FTP proxy

[About the Proxy object](#)

[Completing Proxy object parameters](#)

[Generic FTP Proxy definitions](#)

[Disabling the defense against bounce attacks](#)

About the Proxy object

The Proxy object is used exclusively in client mode for contacting a proxy server. The protocol settings of a Proxy must contain the TCP address details for the connection server, including the set of connection parameters (for example, the user name and corresponding password).

Completing Proxy object parameters

Define the Proxy object using the GUI or the `peladm` online command.

The Proxy object contains the following fields:

Field	Description
name (-name)	Name of the Proxy
protocol (-protocol)	FTP or SOCKS4
address (-address)	Host and port (address = "host/port")
user (-login_user)	User name (optional)
password (-login_password)	Password (optional)
type (-type) (User script)	<i>Mandatory for FTP (otherwise ignored)</i> Name of a standard predefined proxy script or <i>USER</i> .
script (-script)	Script, if <i>USER</i> is the selected type . References the name of a text file by indicating <code>file=<script_filename>.txt</code> .

Example

```
peladm create_proxy -name x_proxy -protocol FTP -address proxy/21
```

```
-user user -password password -type user

-script file=script_filename.txt
```

Use the commands `pelbase import_proxy` and `pelbase export_proxy` to work directly on a text file.

Generic FTP Proxy definitions

The FTP Proxy **type** must describe the *connection sequence at the session level*. The description must include an ordered list of commands to send. A command consists of a line of text including the FTP command and its parameters (possibly variables).

This syntax follows the ABNF format where:

- line = command [SP *(symbol / VCHAR)]
- command = "USER" / "PASS" / "ACCT" / "SITE" / "OPEN"
- symbol = "&(proxy_user)" / "&(proxy_password)" / "&(user)" / "&(password)" / "&(host_port)"

Note: Lines can be separated by either `;` or `\n` and reserved characters must be preceded by `\'` to be escaped.

Predefined proxy scripts

The standard proxy scripts available in Gateway use these variables:

- `&(proxy_user)` : user name for the Proxy
- `&(proxy_password)` : password for the Proxy
- `&(user)` : user name for the server
- `&(password)` : password for the server
- `&(host)` : server host name (can be an IP address)
- `&(port)` : server port number
- `&(host_port)` : server host name and port number, separated by a colon `:` if the port value is given

The following table contains descriptions of the available predefined proxy scripts that you can specify in the **type** field. To customize your proxy script, enter *User script* in the **type** field.

Selecting a predefined proxy script

Type field options	Parameters
USER_AT_SITE	USER &(user)&(host_port) PASS &(password)

Type field options	Parameters
USER_PASS_ACCT	USER &(user)&(host_port) &(proxy_user) PASS &(password) ACCT &(proxy_password)
SITE	USER &(proxy_user) PASS &(proxy_password) SITE &(host_port) USER &(user) PASS &(password)
USER_OPEN	USER &(proxy_user) PASS &(proxy_password) OPEN &(host_port) USER &(user) PASS &(password)
OPEN	OPEN &(host_port) USER &(user) PASS &(password)
USER (User script)	Undefined. Enter a script.

Disabling the defense against bounce attacks

The remote IP address of a proxy control session may be different from the remote address of its data session. This is not normally allowed in Gateway for protection against bounce attacks. Use the `change_data_addr` parameter in the Proxy object to allow different IP addresses. When this parameter is set to *Yes*, FTP no longer checks if the remote IP address is the same for the control session and data session.

You can set this parameter in the Proxy and also in the Remote Site. Refer to [Defending against bounce attacks](#) for more information.

[Working with Proxies](#)

[Working with Proxies \(command line\)](#)

[Working with Remote Sites](#)

[Working with Remote Sites \(command line\)](#)

[Using FTP in Responder mode](#)

Monitoring an FTP Transfer Request

Protocol trace messages

The protocol trace is an optional feature that you define for transfer activity for a given Site.

Activate the protocol trace for more detailed diagnostics when an unexpected error occurs.

Enable or disable trace printing for a Site object at any of the following stages:

- When creating a Site
- When modifying an existing Site
- While Gateway is processing (dynamically)

Activate (or deactivate) the Remote Site **trace** (`-trace_mode`) parameter to enable (or disable) trace printing.

Alternatively, to initiate an immediate trace, use the command:

```
pelctl start_site -trace_mode
```

Gateway prints the corresponding protocol messages to the log file.

If you enable the **trace** option, also set the [log levels](#) in the configuration menu for:

- **FTP control session** - FTP commands and replies (FPDUs)
- **FTP data session** - transferred data

The three possible log levels are:

- Minimum
- Short
- Full

The protocol trace log levels apply to all protocol activity on all Sites. FTP data that is traced in **Short** mode records only the first 16 bytes of each frame (the first line of each command and response). In **Full** mode, all bytes (commands, replies, and transferred data) are logged.

Axway Gateway: Protocols

Transferring multiple files with FTP

About transferring multiple files

Gateway's FTP client can upload and download *multiple files* with a single Transfer Request.

This function is equivalent to the FTP `MPUT` and `MGET` commands:

- **MPUT**: to upload multiple files from a local directory to a remote site
- **MGET**: to download multiple files from a remote site to a local directory

When you use this function, Gateway carries out the following functions:

1. Lists the files of the specified directory.
2. Parses the list.
3. Applies a filter to specified file attributes.
4. Selects the matching files.
5. Requests and attaches the individual file transfers ("children") to a multiple transfer ("master").
6. Uploads or downloads the matching files.

This function is compatible with UNIX and Windows platforms.

Transferring multiple files using MULTI type

Prerequisite: All files must be in the same directory.

Using the GUI

Submit a Transfer Request with the following parameters:

1. In the [General tab](#):
 - In **Type**, select **MULTI**
 - In **Direction**, select **Send** (MPUT or upload) or **Receive** (MGET or download)
2. In the [Attributes tab](#):
 - Complete the **Filter** field to specify the files to send or receive. You can filter on file properties such as name, size, time, etc.
 - Select the **Hash type** to determine the file duplication behavior.
 - *(optional)* Select the **Model to use for child transfers**.

Using online commands

Use the **pe1trans** online command to submit a Transfer Request with the following parameters:

- Set **-type (-ty)** to **MULTI**.
- Set **-direction (-dir)** to **O** (MPUT or upload) or **I** (MGET or download).
- Use **-multi_filter (-muf)** to specify the files to send or receive. You can filter on file properties such as name, size, time, etc.
- Use **-multi_hash_type (-muh)** to determine the file duplication behavior.
- Use **-multi_model (-mum) <Model name>** to select a Model to apply to the *child* transfers.

Example 1: Submitting an MGET using the peltrans online command

```
peltrans -type MULTI -mode I -dir I -oa RS16_STD -da LOCAL -ap FTP_
A
      -dn public/available_things/
```

where:

`dir_name (-dn)` is the name of the directory on the remote computer.

Example 2: Submitting an MPUT using the peltrans online command

```
peltrans -type MULTI -da RS16_STD -oa LOCAL -mode I -dir O -ap FTP_
A
      -fc \Temp\doc\Rfc\HTTP -dn public/received_things/
```

where:

`file_component (-fc)` indicates the name of the local directory that contains the files to send. This parameter is normally used for the name of the file to send.

`dir_name (-dn)` is the name of the directory on the remote computer.

Transferring multiple files using NLSTMULTI type

Prerequisites

- All files must be in the same directory.
- The listing, or the filtering performed over the listing, should only match files. If a directory it is matched it will cause a new transfer, that will be refused by the server.

Notes

- Only *Receive* direction is allowed.
- Duplication check is based only on the name of the files. Using Hash type will not download a new filename with the same name of another file downloaded previously, even if the file content changed.

Using the GUI

Submit a Transfer Request with the following parameters:

1. In the **General** tab:
 - In **Type**, select **NLSTMULTI**
 - In **Direction**, select **Receive** (MGET or download)

2. In the **Attributes** tab:

- Complete the **Filter** field to specify the files to send or receive.
For **NLSTMULTI** you can filter only on file name.
- Select the Hash type to determine the file duplication behavior.
- (optional) Select the **Model to use for child transfers**.

Using online commands

Use the **peltrans** online command to submit a Transfer Request with the following parameters:

- Set **-type (-ty)** to **NLSTMULTI**.
- Set **-direction (-dir)** to **I** (MGET or download).
- Use **-multi_filter (-muf)** to specify the files to send or receive.
For **NLSTMULTI** you can filter only on file name.
- Use **-multi_hash_type (-muh)** to determine the file duplication behavior.
- Use **-multi_model (-mum) <Model name>** to select a Model to apply to the child transfers.

Displaying transfer details

Use the **peldsp display_trans** command to display the details of a Transfer. When you use this command on a master transfer id, you get the following additional lines:

```
x_multi_state='processed all ok'

x_multi_hash_type='1'

x_multi_model=''

x_multi_filter='name=*.h'

x_multi_nb_total='90'

x_multi_nb_ok='90'
```

For **x_multi_state**, there are three possible states:

State	Message displayed	Description
SUCCESS	processed all ok	All child transfers have been transferred successfully.
PARTIAL_SUCCESS	processed partial ok	Error on at least one child transfer.
ERROR	processed all ko	Errors on all child transfers.

This information is also displayed in the GUI.

[Working with File Transfer Requests](#)

[Working with Transfers \(command line\)](#)

[Transfers: Parameters List](#)

Axway Gateway: Protocols

FTP specific user exits

Caution Please note that care should be taken when **using custom user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

Get Application object for FTP: ExitFtpGetAppli

Purpose

This exit function is designed to use the Application if the Remote Site parameter **Application method** (`get_appli_method`) is set to *User exit (AIE)*.

Syntax

You can find this exit in the `exitftp.c` source file. It has the following syntax:

```
int ExitFtpGetAppli(
    char dest_alias, /* Remote Site */
    char filename, /* FTP file name */
    short data_code, /* transfer data_code */
    char appli_name, /* Application from P_APPLI site command */
    char destination, /* alias name of final destination */
    char originator, /* alias name of origin */
    char buffer, /* buffer to store Application name */
    int len); /* max buffer length */
```

If the Remote Site parameter **Application method** (`get_appli_method`) is set to *User exit (AIE)*, the Gateway server calls this function immediately after receiving an FTP **STOR** or **STOU** command.

The function gives an example using a look-up table from the transfer data code and destination alias name. The result must be stored in the `buffer` parameter whose size is **len** bytes.

The function can return three values:

- 0: Application found and stored in the buffer parameter
- 1: No application found and the default FTP internal application is used
- 2: No application found and the transfer is rejected

Customize FTP: *ExitFtpSetReturnCode*

Purpose

This feature launches customized actions on receipt of an FTP command on the server side. It is implemented with a user exit function that is called for each FTP command received (except for `USER`, `PASS` and `SYST`).

You can use the exit function to:

- Let Gateway continue with its normal process
- Supply a return code and message so that Gateway skips the command processing and responds immediately with the code and message supplied by the exit

Advantages of this feature:

- It simulates a positive completion return code for commands that are not supported
- It forces a negative completion return code for commands that are not permitted

Syntax

You can find this exit in the `exitftp.c` source file. It has the following syntax:

```
int ExitFtpSetReturnCode (  
    int ftp_ref, /* identifier of the FTP FT */  
    char ftp_cmd, /* received FTP command */  
    int ftp_code, /* response code to send */  
    char ftp_msg); /* response to send */
```

The return value of this function is always 0.

This table describes the input and output parameters:

Type	Parameters	Comments
Input	ftp_ref	Reference of the actual control connection for that command.
	ftp_cmd	Full command received, including its parameters, but without CRLF.
Output	ftp_code	Mandatory States the code that must be returned in response to the above command: • 0: If the exit function allows Gateway to process the command normally • non-0: If the command will not be processed and Gateway will immediately respond with this ftp_code
	ftp_msg	<i>Optional</i> (only used if the returned ftp_code value is not 0) In addition, if ftp_msg does not contain any message (strlen(ftp_msg) is 0), Gateway uses the standard message associated with the ftp_code. This parameter should be used with caution. The message must not exceed 200 characters.

Default exit function examples

This is the code supplied by default with Gateway. It allows Gateway to operate as if there were no exit function.

```
int ExitFtpSetReturnCode(int ftp_ref, char *ftp_cmd,
                        int *ftp_code, char *ftp_msg)
{
    *ftp_code = 0;
    ftp_msg[0] = 0;

    if (*ftp_code > 0)
        LogPrintf(LOG_INFO, "EXITFTP (%d) reply code set to %d for \
                    command \"%s\"\n", ftp_ref, *ftp_code, ftp_cmd);

    return(0);
}
```

Example 1: Simulating a positive completion response

This example shows how to simulate a positive completion for commands that are not implemented in Gateway.

```
int ExitFtpSetReturnCode(int ftp_ref, char *ftp_cmd,
                        int *ftp_code, char *ftp_msg)
{
    *ftp_code = 0;
    ftp_msg[0] = 0;

    if (memcmp(ftp_cmd, "MKD", 3) == 0)
        *ftp_code = 257;
    else if (memcmp(ftp_cmd, "RMD", 3) == 0)
        *ftp_code = 250;
    else if (memcmp(ftp_cmd, "CWD", 3) == 0)
        *ftp_code = 250;
    else if (memcmp(ftp_cmd, "CDUP", 4) == 0)
        *ftp_code = 200;
    if (*ftp_code > 0)
        LogPrintf(LOG_INFO, "EXITFTP (%d) reply code set to %d for \
            command \"%s\"\\n", ftp_ref, *ftp_code, ftp_cmd);

    return(0);
}
```

Example 2: Forcing a negative completion response

This example shows how to force a negative completion for some commands. If you decide that files should never be received on the Gateway server, a simple way to do this is to intercept **STOR** and **STOU** commands and respond with a 502 (command not implemented) or 532 (need account for storing file) return code. In addition, the `ftp_msg` parameter is used to reply with a non-standard message.

```
int ExitFtpSetReturnCode(int ftp_ref, char *ftp_cmd,
                        int *ftp_code, char *ftp_msg)
{
    *ftp_code = 0;
    ftp_msg[0] = 0;
    if ((memcmp(ftp_cmd, "STOR", 4) == 0) ||
        (memcmp(ftp_cmd, "STOU", 4) == 0))
    {
        *ftp_code = 502;
        strcpy(ftp_msg, "Sorry this server does not accept \
            store commands");
    }
    if (*ftp_code > 0)
```



```
LogPrintf(LOG_INFO, "EXITFTP (%d) reply code set to %d for \
        command \"%s\"\n", ftp_ref, *ftp_code, ftp_cmd);
return(0);
}
```

[Internal user exits](#)

The topic contains information about the following FTP specific user exits:

- [ExitFtpGetAppli](#)
- [ExitFtpSetReturnCode](#)
- [ExitFtpRespPasv](#)
- [ExitFtpCmdPort](#)

Axway Gateway: Protocols

HTTP

About HTTP

[HTTP](#)

[HTTP and Gateway](#)

HTTP Introduction

The **H**yper**t**ext **T**ransfer **P**rotocol (HTTP) is an application-level protocol, used as a generic protocol for communication between users, proxies, or gateways and other Internet systems.

Gateway implements both HTTP/1.0 client and server specifications. Configuration of the relevant objects for HTTP depends on whether you are using Gateway in server mode or client mode.

How does HTTP work?

HTTP is a client/server protocol based on a simple request-response dialog. As a stateless protocol, HTTP requires that every request sent by a client contain all the information (for example, identification parameters) necessary for the server to execute it. A complete dialog, which is always initiated by the client, consists of one request and one response.

Sometimes one or more intermediaries, such as a proxy, firewall, or gateway, are present in the request/response chain.

To end an HTTP session the client takes no explicit action as the connection is closed after each request/response dialog pair.

HTTP and Gateway

You can use HTTP in Gateway to:

- Send and receive files in client and server mode
- List available files or transfers
- Customize error pages and PUT command results
- Access virtual file directories via the Gateway GUI

Gateway enables you to implement routing and execute user exits when sending or receiving transfers.

[Configuring the HTTP environment](#)

[Using HTTP in server mode](#)

[Using HTTP in client mode](#)

[Using HTTP/HTTPS proxies](#)

[Adding non-standard HTTP headers](#)

Axway Gateway: Protocols

HTTP: Configuring the environment

[About HTTP configuration](#)

[HTTP configuration](#)

[Configuring HTTP security](#)

About HTTP configuration

Before creating a transfer, you must configure a specific HTTP Application and Sites if they do not exist. In both client and server mode, you must define at least one each of the following Gateway objects:

- **Local Site**
During installation, Gateway creates a default Local Site corresponding to the alias name and protocol identifier you supplied at that time. It is the default Local Site for all the protocols supplied by the product, including HTTP.
- **Remote Site**
Define your partner Site and communication attributes in the Remote Site object.

- **Application**

A default Application object is delivered for each data code.

You can create these objects either via the GUI or the `pe1adm` online command.

HTTP configuration

Creating a Site object

A transfer begins with the physical connection of two Sites. In Gateway the terms *Initiator* and *Responder* are used in place of the standard HTTP terms *Client* and *Server*.

The Responder Site responds to the connection request from the Initiator Site. During the connection phase:

- The Initiator sends its identification
- If the Responder is an Axway MFT product, it checks the Initiator identification in the Remote Site attributes

The HTTP protocol allows data transfer in both directions: the Site sending the data is the Sender, the other is the Receiver. A given Site can have the following roles:

- Initiator/Sender
- Initiator/Receiver
- Responder/Sender
- Responder/Receiver

Gateway references a Site by its protocol identifier. For each Axway MFT product, you can define as many Local and Remote Sites as needed. The Local Site is defined by the **Partner local alias** (`-partner_loc_alias`) attribute in the Remote Site object, or by the default Local Site name (defined during the installation).

It is possible to use the same Site for simultaneous file transfers in both directions.

Defining Remote Site parameters

Use the Remote Site to:

- Provide the network address and SAP of the partner
- Select the method to use for the identification phase, and information relevant to this method (user name and password for the login method)
- Define the Application selection method if it is not defined in the Transfer Request

If you do not define a parameter, Gateway uses default values from the operator interface or from the internal default attributes.

The following table summarizes the possible roles the Remote Site should have to perform specific actions.

Action	If RS represents:	Then it must be defined as:
Send a file from your Local Site	Destination	Receiver
Receive a file	Partner	Sender

The parameters representing the mode and action of the Remote Site (**init_sender**, **init_receiver**, **resp_sender**, **resp_receiver**) are all set to Yes (Y) by default.

If you do not specify a group when creating a Remote Site, **GDDEFAULT** is the group name associated with the Site by default. The group name of the Remote Site and Application must be the same.

Creating an Application

The Application object manages transfer and file attributes such as record format and padding.

HTTP uses the default FTP Application, *FTP_B*, which should not be modified.

Configuring HTTP security

TLS and SSL protocols are fully supported by both HTTP client and server and their use is completely transparent for the HTTP protocol. To implement security, you must:

- Select the option TLS/SSL in the configuration of the protocol.
- Create security objects:
 - Network Profile (netprof) for the client
 - TLS Security Profile (sprof)
 - Certificate

The use of TLS or SSL with Web browsers consists in replacing *http://server.com* with *https://server.com* in the browser navigation bar.

[About Site objects](#)

[Site objects: Parameters List](#)

[About Applications](#)

[Using HTTP in client mode](#)

[Using HTTP in server mode](#)

Using HTTP in server mode

About the HTTP server

Use HTTP in server mode, to perform one of the following tasks:

- **Upload files**
Accept a file that a client (Initiator/Receiver) is sending. When you download files, the URL points to a file on the server.
- **Download files**
Make files available to a client (Initiator/Sender). The client request contains the file to be stored in the directory to which the URL points.
- **List directories**
Display a list of available files (Responder/Sender). The URL points to a directory.

For each request received, the Gateway HTTP server must identify the action requested by the client. The server first extracts relevant elements from the client request (for example, URL, and CGI parameters). It then verifies user identification, checks the URL and executes the requested action.

Before you transfer a file

Before you send or receive an HTTP transfer, you must configure CGates and VFDs in Gateway. These objects assign directories and directory rights to connecting parties. It is also recommended that you configure the identification method.

CGates

CGates associate certain product elements (for example, Sites or Templates) with a protocol connection (identification). They are required to initialize and manage the virtual file system.

The HTTP server uses CGates to set up certain variables (for example, the user home page, or the directory listing template).

VFD

The Virtual File Directory (VFD) is a virtual file system that HTTP clients see when they connect to Gateway. This file system can monitor transfers, display virtual directories, and provide links to real file system directories.

The HTTP server first initializes a new VFD session with the identified CGate parameters (including root directory, and access rights details) and then uses the VFD to associate the requested URL to a file, a transfer, or a directory.

Configuration

Throughout this topic, references are made to HTTP configurable elements, such as the authentication type, that you can modify using the [configuration menu](#).

Processing HTTP client requests

The HTTP server processes a client request in the following sequence:

-
- 1** Analyzes the first line of the request and extracts the relevant elements:
 - HTTP transfer method (GET, PUT, POST)
 - URL accessed
 - CGI parameters
 - Protocol used (HTTP/1.0, HTTP/1.1)
 - 2** Stores received HTTP headers
 - 3** Extracts CGI parameters sent in the body of the request (POST method only)
 - 4** Suspends the incoming TCP flow once the entire request is received
 - 5** Identifies the connected party via the user, password or session ID received in CGI parameters or HTTP headers (depending on the server identification mode set in the configuration menu)
 - 6** Identifies the requested URL and checks access rights
 - 7** Releases the incoming TCP flow
 - 8** Processes the request (upload, download, or directory listing)
-

If an error occurs during the execution of one of these steps, an error page is returned to the connected client. For more details, refer to [Error pages](#) further on in this topic.

Identification methods

How the Gateway server identifies the client

Before the Gateway HTTP server can execute an HTTP request and before it checks the URL provided, it must identify the connected party. The Gateway HTTP server retrieves the *user/password* pair and assigns CGate and Site parameters to it.

The Gateway HTTP server can collect identification parameters in various ways, but it can only be configured to use one identification method at a time. The Gateway HTTP server uses **Web** identification by default.

You can change identification parameters (*user*, *password*, and *sessionid*) in the [configuration menu](#). This menu also enables you to configure session management (including, for example, the maximum number of created sessions).

Supported identification methods:

- Web
- CGI user and password
- CGI session ID
- Cookie user and password
- Cookie session ID

Web

Web identification, which can be called WWW authorization, is the only standard identification method defined by the HTTP protocol. The other methods presented in this topic are application-level mechanisms.

Summary of the Gateway HTTP server search mechanism and specific actions using the *Web* identification method:

Search method (in sequence)	If element found	If login fails
Searches in the request header for an Authorization field and identifies the connected party from the parameters found	Server logs in the user	Server adds a negative response to the header and returns it with an error page informing the client that the credentials were not valid and prompting the user to try again
Uses anonymous to log in		

Note: Browsers typically try to access the requested URL anonymously the first time they connect to a server. If **anonymous** is not allowed to log in, the connected user is prompted for a login and password.

CGI user and password

Clients can send these parameters in either or both of these elements:

- URL, separated from the path by a question mark ("?")
- Body of the request (POST requests only)

As the CGI parameters are user-level parameters, standard HTTP clients, such as browsers, do not send them on each request. It is the role of the Gateway server to send parameters to the HTML pages that are requested and to elements such as links and forms.

Summary of the Gateway HTTP server search mechanism and specific actions using the *CGI user and password* identification method:

Search method (in sequence)	If element found
User-level parameters received	
Uses anonymous to log in	Server logs in the user

CGI session ID

Using CGI session identification, it is the role of the Gateway HTTP server to send the *sessionid* parameter to the returned HTML pages.

Summary of the Gateway HTTP server search mechanism and specific actions using the *CGI session ID* identification method:

Search method (in sequence)	If found and login correct
Searches for <i>user</i> and <i>password</i> parameters in the CGI parameters received	Server uses them as credentials and if login succeeds, creates a session ID
Searches for the <i>sessionid</i> parameter used to retrieve the <i>user</i> and <i>password</i> provided during session creation	Server logs in the user
Uses anonymous to log in	

Cookie user and password

Cookies are set for the whole directory tree (everything under path "/"). No expiration dates are attached to them: Web browsers store cookies for the current session and automatically destroy them when the current session is closed.

Summary of the Gateway HTTP server search mechanism and specific actions using the *Cookie user and password* identification method:

Search method (in sequence)	If element found	If login fails
CGI parameter list		
Cookies sent by the client	Sever logs in the user	Server returns two Set-Cookie fields with empty values in order to clear cookies set on the browser side
Uses anonymous to log in	Server returns two Set-Cookie header fields to the client. These fields request the client to store user and password within its cookies.	

Cookie session ID

Summary of the Gateway HTTP server search mechanism and specific actions using the *Cookie session ID* identification method:

Search method (in sequence)	If element found and login correct	If login fails
Searches for user and password in the CGI parameters list	Server logs in the user and creates a new session	
Searches for a session ID in the cookies sent by the client	Server logs in the user with the <i>user</i> and <i>password</i> values saved for this session ID	Server returns a Set-Cookie field with an empty value in order to clear the cookie set on the browser side

When the Gateway HTTP server creates a new session, it returns a **Set-Cookie** field to the client. This field requests the client to store *sessionid* in its cookies list. When stored as a cookie, *sessionid* is automatically re-sent on each new request to the server.

Partner-Dependant identification

The server determines the authentication method to use from the request headers sent by the client software. HTTP clients can log in using any of the authentication methods supported by Gateway. Note that the HTTP server accepts all the connections having a supported identification method.

HTTP session ID limitation

You can limit the number of simultaneous opened sessions for a user.

A global configuration parameter, `sid_limit`, limits the number of sessions a user can have on the server.

Each time a user logs in, the value of this parameter is compared with the number of sessions registered for that user.

If the number of registered sessions is lower than the value of the configuration parameter, the user is allowed to login. Otherwise the login is refused.

You can find the `sid_limit` parameter in the Gateway [advanced parameters](#).

This parameter is only taken into account if you have set the HTTP server identification **Method** to *Cookie session id* or *CGI session id* in the configuration menu.

Directory listings

An HTTP directory listing returns to the client an HTML page containing a list of links to elements (files and subdirectories) contained in the directory. A directory listing is performed when the accessed URL refers to a directory.

The Gateway virtual file system can contain the following elements:

- **Directories**
- **Files:** Only available if the directory refers to a real system directory
- **Transfers:** References to Gateway transfers created in that directory. Only available if the directory is virtual (VFD). When displaying Gateway transfers, the server differentiates the available files in the directory listings as described in the following table:

If the Transfer type is:	Then the connected party can:
available	Retrieve the transfer (that is, the file) by clicking the transfer name, which is a hypertext link
unavailable	Display transfers in the current directory (similar to FTP). Transfers are displayed for informational purposes only and no hypertext links are required.

To fulfill these requirements, Gateway uses an HTML page to display directory contents. This page is called a listing template and includes elements that adapt the listing to the contents of the listed directory (for example, files, sub-directories, and transfers).

HTTP listing template files

The listing template files contain the HTML formats used to list directories. First, the server must determine which file to use as a pattern. The file path to this pattern is specified in the:

- CGate selected during login phase.
The value can be taken from one of the selected parent CGateGroups.
- Default template file defined in the HTTP configuration part of Gateway.
You can modify this value using the configuration menu.

Once found and loaded, the server stores the patterns in memory and reloads them only when their files were modified. If no file is defined or if the file cannot be loaded, an error is returned to the client.

When the Responder (server) retrieves the listing pattern, it processes the listing in the following sequence:

- 1** Substitutes the global symbols (for example "%") in the page with their respective values (refer to *Symbol substitution* later in this topic for more information)
- 2** Searches for special tags (defined in *Using special tags* below) and replaces them with each corresponding element of the listed directory
- 3** Returns the dynamically-generated pages to the Initiator (client)

Using special tags

Gateway recognizes the following special tags that you can use in the template files.

Tag	Comment
<PARENT> </PARENT>	Text between these tags is displayed only if the listed directory is not "/".

Tag	Comment
<DIR> </DIR>	Text between these tags is duplicated for each directory in the current directory. The following special names are also replaced: • @NAME@: directory name encoded for URL • @name@: directory name • @DATE@: Directory last modification date, per component: @DATE_Y@ (year), @DATE_M@ (month), @DATE_m@ (month as a decimal number), @DATE_D@ (day), @DATE_d@ (day as a decimal number), @DATE_T@: directory last modification time • @CONTENT@: number of files and number of available files from current directory. For example: 43 files (30 available)
<REAL_FILE> </REAL_FILE>	Text between these tags is duplicated for each file in the directory. It is only available under mounted directories. The following special tags are also replaced: • @FNAME@: file name encoded for URL • @fname@: file name • @DATE@: Directory last modification date, per component: @DATE_Y@ (year), @DATE_M@ (month), @DATE_m@ (month as a decimal number), @DATE_D@ (day), @DATE_d@ (day as a decimal number), @DATE_T@: directory last modification time • @SIZE@: file size • @ICON@: icon name associated with the directory path and filename, encoded for URL (for more information, refer to Configuration file).

Tag	Comment
<code><XFER_AVAIL></code> <code></XFER_AVAIL></code>	Text between these tags is duplicated for each available transfer in the current virtual directory. These tags are ignored inside real directories. The following special names are also replaced: • @FNAME@: transfer file name encoded for URL • @fname@: transfer file name • @DATE@: Directory last modification date, per component: @DATE_Y@ (year), @DATE_M@ (month), @DATE_m@ (month as a decimal number), @DATE_D@ (day), @DATE_d@ (day as a decimal number), @DATE_T@: directory last modification time • @SIZE@: file size • @LOCAL_ID@: transfer local identifier • @STATE@: transfer status • @ICON@: icon name associated with the directory path and filename, encoded for URL (for more information, refer to Configuration file).
<code><XFER_UNAVAIL></code> <code></XFER_UNAVAIL></code>	Text between these tags is duplicated for each unavailable transfer in the current directory. These tags are ignored inside real directories. Substitutions described above apply here

Symbol encoding

As a mechanism for the prevention of XSS (Cross-Site Scripting), Gateway offers the possibility to encode HTML entities according to the context in which they are used. Special tags like @NAME@ and @FNAME@ are already encoded for an URI context; however a higher level of control is desirable. For example, the directory or file name can be used in a different context than that of an URI; such as a JavaScript/HTML context. The proper encoding can be done by using a special tag with no default encoding, and suffixing it with an encoding context specifier; e.g. @name@forJsHtml(), @fname@forJsHtml().

The following types of encoding can be used:

- `forUri()`
URI context
- `forUriComponent()`
URI component context
- `forCssUri()`
CSS URI context
- `forCssString()`
CSS string context

- `forJsSource()`
JavaScript source context
- `forJsAttribute()`
JavaScript attribute context
- `forJsBlock()`
JavaScript block context
- `forJsHtml()`
JavaScript/HTML context
- `forHtml()`
HTML context

Caution Make sure every special tag that could be controlled by an attacker is using HTML entity encoding, either explicit or implicit. See the example below, and the sample templates delivered with the product.

Note For special tags with implicit encoding (@NAME@, @FNAME@), overwriting by specifying an explicit encoding is not supported.

For more detail regarding XSS (Cross-Site Scripting) and HTML entity encoding, please refer to the OWASP website:

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

(URLs are outside the Axway domain, and liable to change without notice).

Example

The following is an example of a directory listing template file.

```
<HTML>
<HEAD>
<TITLE> %u </TITLE>
</HEAD>
<BODY>
<H1> Index of %u </H1>
<PARENT>
<A href="%N%i">Parent directory (%n)</A> <BR>
</PARENT>
<HR> <PRE>
<DIR>
@DATE@ <A href="%U%/@NAME@/%i">@name@ .forHtml() </A>
</DIR>
Date Size State Name Local Ident
<REAL_FILE>
@DATE@ @SIZE@ <A href="%U%/@FNAME@/%i">@fname@ .forHtml() </A>
</REAL_FILE>
<XFER_AVAIL>
@DATE@ @SIZE@ @STATE@ <A href="%U%/@FNAME@?local_ident=@LOCAL_
```

```

IDENT@%&%I">@fname@ .forHtml() </A> @LOCAL_IDENT@
</XFER_AVAIL>
<XFER_UNAVAIL>
@DATE@ SIZE@ @STATE@ @fname@ .forHtml() @LOCAL_IDENT@
</XFER_AVAIL>
</PRE> <HR>
</BODY>
</HTML>

```

Downloading files

To download a file, the client must send a GET (or POST) request with a URL that matches either a [real file](#) or an available transfer.

As different transfers in a directory can have the same file name, you can use a [selection filter](#) to identify a specific transfer.

Uploading files

To upload a file onto the server, you must send the following elements:

- Path name (directory in which to receive the transfer)
- File name
- File contents
- Optional transfer parameters

The URL always contains the directory where the transfer is performed. The file name, however, is deduced as follows:

Attributing file names when uploading

HTTP request method	If URL format is	Then the filename is
PUT (not supported by browsers)	file path (directory + filename)	The last component of the file path
	directory name	The value of the CGI parameter <i>filename</i>

HTTP request method	If URL format is	Then the filename is
	file path (directory + filename)	The last component of the file path
POST (content type must be "multipart/form-data")	directory name	The value of the CGI parameter <i>filename</i> , if present, or the name of the file sent (extracted from the form field used to send the file)

Defining Axway MFT specific parameters

If the client is an Axway MFT product, optional parameters are available for more detailed communication. The parameters that you can set in the request include:

- Application (**P_APPLI**)
- Remote destination of the transfer (**P_DEST**)
- Origin of the transfer (**P_ORG**)
- Message to attach to the transfer (**P_MSG**)

These optional transfer parameters are sent through CGI parameters, either in the URL or in the request body (POST requests).

The syntax is:

parameter=value

Parameters are separated by an ampersand (&).

For examples of requests using the PUT and POST methods, refer to [Using HTTP in client mode - Sending files](#).

Note: The server checks and stores these parameters before the file transfer begins. This implies that they must be sent before the file itself. In other words, these parameters must be set either in the URL CGI parameters or in the CGI parameters preceding the file. If the reception results from a form submission, then the **file parameter must be positioned as the last parameter of the form**.

Restricting selection: File filters

As a virtual directory and a file name are insufficient to describe a Mailbox entry, Gateway provides an advanced file filter feature when downloading and listing requests.

A file filter comprises strings that contain **field=value** pairs. When you use HTTP, these strings are extracted from CGI parameters. Accepted values for the *field* portion of this pair are given in the table below:

Field	Accepted value
name	Pattern matching paths and protocol filenames
direction¹	IN, OUT, or BOTH. Not case-sensitive
date²	Exact date and time of the transfer, expressed in the format: <i>YYYYMMDDhhmmss</i> • Date fields (year, month, day of month) can be separated by "/" characters. • Time fields (hours, minutes, and seconds) can be separated by ":" characters. You can specify time without a date. Current day is considered the default date.
end_before or endbefore²	Date format (Transfers requested or performed earlier than the given date are not selected)
start_after or startafter²	Date format (Transfers requested or performed later than the given date are not selected)
origin or org¹	Origin of the transfer
destination or dest¹	Destination of the transfer
state¹	List of states in Gateway format. Each selected transfer must have one of the given states.
mvs_state or mvsstate¹	List of states in MVS format. Each selected transfer must have one of the given states.
ftpcs_state or ftpcsstate	List of states in FTPCS format. Each selected transfer must have one of the given states. The FTPCS state is the concatenation of the state in Gateway format and the state in InterPel MVS format.
local_ident or localident	Long integer corresponding to the unique transfer local identifier
xfer_ident or xferident	Long integer corresponding to a transfer identifier

(1) For real entries and directories, these fields are ignored.

(2) These fields are compared to the last modification date of the file.

Example 1: Filtering by date

Downloading:

```
GET /transfer/out/file1.txt?org=sitel&start_after=20011205
```

Example 2: Filtering by transfer state

Directory listing:

```
GET /?name=*.txt&state=IC&org=sitel
```

Notes: When you retrieve a transfer, if several records match the request, the most recent one is returned.

Example 3: Filtering by file type

Requesting an incorrect file type:

```
GET /transfer/file?state=C
```

In this example, the user tries to download the most recent entry called **file** and in the state **canceled**. Although this is syntactically correct, Gateway does not allow downloading of canceled files and returns the following response:

404 Not found

Sort/Order parameters

There are two parameters for sorting and setting the order of fields:

- `order (order=<order_field>)`

If negative, the sorting order descends.

If positive, sorting order ascends.

- `sort (sort=<sort_order_type>)`

Only one field can be chosen from the following:

"xfer_type", "size", "date", "local_id", "protocol", "rights", "direction", "state", "org", "dest",
"remote", "fname", "xfer_id", "route_state", "perm", "appli", "msg", "code", "route_from",
"route_to", "dup_from", "param1", "param2", "user_state", "local_ident", "name", "xfer_ident"

Gateway specific template files

This section describes the HTML pages returned by the server when:

- Errors occur
- Uploads complete successfully
- Uploads fail

Error pages

Error pages are HTML files returned to the client when an error occurs. The file returned depends on both the type of error and the languages accepted by the connected client.

When Web browsers connect to a server, they typically send an HTTP header field named *Accept Language* that contains an ordered list of accepted languages. The language used by the HTTP server is the first that matches one of its accepted languages. Use the parameter `err_languages` to configure the server-accepted languages, whose default values are **en** [English], and **fr** [French]. If none of the client languages match, the first default error language is used (en).

Error files are located in the sub-directories of the error directory (parameter `err_dir` whose default value is `<Gateway installation directory>/run_time/http/pages`). The sub-directories must have the same name as the accepted languages. Default directories are:

- `<Gateway installation directory>/run_time/http/pages/en` for English
- `<Gateway installation directory>/run_time/http/pages/fr` for French

The following table lists the files used:

Error pages displayed by return codes		
Filename	Return code	Error description
ERR_BAD_REQUEST	400	The request contains invalid or missing parameters
ERR_BAD_VERSION	505	Protocol version is not compatible
ERR_FORBIDDEN	403	Requested action is forbidden (read, write, and so on)
ERR_INTERNAL	500	An internal error occurred, for example a memory error, or a real or virtual file system access error
ERR_NOT_FOUND	404	Could not find requested URL
ERR_SESSIONID	403	Could not find received session ID (unknown or expired session)
ERR_UNAUTHORIZED	401 (Web auth)	Invalid credentials

Error pages are loaded in memory the first time you access them. The HTTP server reloads them only when their files are modified. [Symbol substitutions](#) apply to these files.

Upload success and failure pages

The following table summarizes the features of the upload success and failure pages:

Page	Returned when	Associated HTTP return code	Additional symbol substitutions
Upload success	File correctly received	200	"%x" replaced by the received and validated request URL, or "/" if the request URL cannot be validated.
Upload failure	File incorrectly received	500	"%x" replaced by the received and validated request URL, or "/" if the request URL cannot be validated.

Note For security reasons, using the CGI parameters `success_url` and `failure_url` for symbol substitution is no longer supported.

Configuration file

The configuration file is used to set up all Web server behavior that is based on URLs. The default file is `<Gateway installation directory>/run_time/http/server.ini`. The HTTP server automatically reloads the configuration when the file is modified.

Each line of the file must take the form:

```
command "argument" pattern [&]
```

Empty lines and lines beginning with "#" are ignored. Patterns can match file extensions ("`.txt`", "`.htm`"), file names ("`*-test-*.*`", "`*.tar.gz`") or URLs ("`/transfer/*`").

Configurable elements are:

- Content types returned to the client for the accessed URL
- Icons to use with directory listing entries
- Files for which symbol substitutions must be performed dynamically

Content types

The content type (also known as media type or MIME type) is a description of the data that the browser uses to determine how to process the received data.

Lines beginning with the `AddType` command define the **content-type** header field returned to a browser.

AddType must be followed by a content type and patterns. If one of the patterns matches the URL, the content type is returned to the client.

Example

The following example forces browsers to display all files ending with `.htm` or `.html` as Web pages. All other pages must be prompted for downloading:

```
AddType    "text/html"    *.htm *.html
AddType    "application/octet-stream" *
```

Icons

When modifying templates for directory listings, you can display an icon for each kind of file present in the directory by using the `AddIcon` command, followed by the icon URL and patterns. When a pattern matches the URL of a listed element (file or transfer), the symbol `@ICON@` of the listing template is replaced by the URL icon.

Example

In the following example, the listing template `@ICON@` symbol is replaced by:

- `/icons/text.gif` for file names that end with `.txt`
- `/icons/binary.gif` for file names that end with anything else

```
AddIcon    "/icons/text.gif" *.txt
AddIcon    "/icons/binary.gif" *
```

Substituting symbols in files

The server can apply the same symbol substitution to downloaded files as the one applied to other elements such as error pages and listing templates.

To substitute symbols in files, add a line beginning with `AddHandler`, followed by the kind of action to perform: *internal*, and then the patterns.

Example

In the following example, a symbol substitution is applied to every real file ending with `.mhtml`.

```
AddHandler internal *.mhtml
```

Symbol substitution

Symbol substitution consists in replacing symbols (for example, %) read in a file with a value, which is then returned to the client. The substitutions apply to:

- Directory listing template files
- Error pages
- Upload success and failure pages
- Real files that match an **AddHandler** condition of the server configuration file. [Real files](#) are files located under a [mounted directory](#).

Symbol substitution in HTML files

Symbol	Replaced by
%g	Fragment that follows the marker at the end of a URL
%G	URL CGI parameters
%h	Returned HTTP code
%H	First line of the request
%i	CGI identification string (preceded by a "?"): • user and password (?user=&&password=) • session ID (?sessionid=) empty for other cases
%I	CGI identification string (%i without "?")
%j	User login
%J	User login encoded for URL
%k	User password
%K	User password encoded for URL
%l	Session ID (if any)
%L	Session ID encoded for URL
%n	Upload directory path
%N	Upload directory path encoded for URL

Symbol	Replaced by
%p	Client HTTP method
%P	Client HTTP protocol
%r	Error number
%R	Error string
%s	CSRF token (see Cross-Site Request Forgery prevention on page 1137)
%S	CSRF token encoded for URL (see Cross-Site Request Forgery prevention on page 1137)
%t	Home directory for user
%T	Home directory for user encoded for URL
%u	URL decoded
%U	URL
%v	Home page for user
%V	Home page for user encoded for URL
%%	Character %
??	Character ? - if identification string is not empty or null
&&	Character & - if identification string is not empty or null

HTTP transcoding

In server mode, Gateway stores payloads with `Content-Type=text/*` with native line ending. In cases where the header is missing or has another value, the payload is treated as binary and is stored unchanged. The behavior is the same for both PUT and POST HTTP methods (request types). Only text entities using ASCII CR / LF characters as line endings are supported. If transferred text entities are encoded with a multi-byte character set that uses other characters for new lines, the payload will be corrupted.

Transfer monitoring and log levels

A monitoring option exists for the virtual file system directories. This option allows you to send and receive files in silent mode in these directories, without creating monitor Transfer Requests.

Disabling the monitoring option in directories where files can be downloaded several times without transfer controls (for example, size and progression indicators) can increase performance.

HTTP uses a log level (`http_log_level`) to determine whether to add a request trace to the log file. This parameter is set in the `conf` file during configuration or dynamically prior to starting the transfer.

You can set four levels:

- **0**: no requests are logged
- **1**: only monitor Transfer Requests are logged
- **2**: all Transfer Requests are logged (monitored or unmonitored transfers)
- **3**: (default) all requests are logged (including listings, and invalid requests)

HTTP Strict Transfer Security

The HTTP Strict Transfer Security (HSTS) allows Gateway to declare that compliant user agents should only interact with it using secure HTTP connections. Please refer to the RFC 6797 (<https://tools.ietf.org/html/rfc6797>) for more details.

This is communicated by Gateway (acting as a server) via the "Strict-Transport-Security" HTTP response header field. Gateway supports the following directives: `max-age` and `includeSubDomains`

Configuring the HTTP Strict Transfer Security header

The HSTS header can be configured using the following configuration parameters, using the `peluconf` command:

Parameter	Description
<code>hsts_enabled</code>	Enables the inclusion of the HSTS header field in HTTPS responses.
<code>hsts_max_age</code>	Sets the <code>max-age</code> directive in the HSTS response header field to the specified value (default 0).
<code>hsts_include_subdomains</code>	Enables the inclusion of the <code>includeSubDomains</code> directive in the HSTS response header field.

All the configuration parameters are in the `ft_http` configuration section. For example, use:

```
peluconf set -s ft_http hsts_enabled 1
```

to enable the HSTS header field, and

```
peluconf set -s ft_http hsts_max_age 86400
```

to set the `max-age` directive to 24 hours.

Note Setting the `hsts_max_age` parameter to `0` will not disable the inclusion of the HSTS response header field. Instead, the HSTS header field will be sent with a `max-age=0` directive. This will effectively remove the Gateway domain from the User Agent's Known HSTS Hosts cache.

Note The HSTS response header field is only sent through HTTPS connections.

Identification methods

How the Gateway server identifies the client

Before the Gateway HTTP server can execute an HTTP request and before it checks the URL provided, it must identify the connected party. The Gateway HTTP server retrieves the *user/password* pair and assigns CGate and Site parameters to it.

The Gateway HTTP server can collect identification parameters in various ways, but it can only be configured to use one identification method at a time. The Gateway HTTP server uses **Web** identification by default.

You can change identification parameters (*user*, *password*, and *sessionid*) in the [configuration menu](#). This menu also enables you to configure session management (including, for example, the maximum number of created sessions).

Supported identification methods:

- Web
- CGI user and password
- CGI session ID
- Cookie user and password
- Cookie session ID

Web

Web identification, which can be called WWW authorization, is the only standard identification method defined by the HTTP protocol. The other methods presented in this topic are application-level mechanisms.

Summary of the Gateway HTTP server search mechanism and specific actions using the *Web* identification method:

Search method (in sequence)	If element found	If login fails
Searches in the request header for an Authorization field and identifies the connected party from the parameters found	Server logs in the user	Server adds a negative response to the header and returns it with an error page informing the client that the credentials were not valid and prompting the user to try again
Uses anonymous to log in		

Note: Browsers typically try to access the requested URL anonymously the first time they connect to a server. If **anonymous** is not allowed to log in, the connected user is prompted for a login and password.

CGI user and password

Clients can send these parameters in either or both of these elements:

- URL, separated from the path by a question mark ("?")
- Body of the request (POST requests only)

As the CGI parameters are user-level parameters, standard HTTP clients, such as browsers, do not send them on each request. It is the role of the Gateway server to send parameters to the HTML pages that are requested and to elements such as links and forms.

Summary of the Gateway HTTP server search mechanism and specific actions using the *CGI user and password* identification method:

Search method (in sequence)	If element found
User-level parameters received	Server logs in the user
Uses anonymous to log in	

CGI session ID

Using CGI session identification, it is the role of the Gateway HTTP server to send the *sessionid* parameter to the returned HTML pages.

Summary of the Gateway HTTP server search mechanism and specific actions using the *CGI session ID* identification method:

Search method (in sequence)	If found and login correct
Searches for <i>user</i> and <i>password</i> parameters in the CGI parameters received	Server uses them as credentials and if login succeeds, creates a session ID
Searches for the <i>sessionid</i> parameter used to retrieve the <i>user</i> and <i>password</i> provided during session creation	Server logs in the user
Uses anonymous to log in	

Cookie user and password

Cookies are set for the whole directory tree (everything under path "/"). No expiration dates are attached to them: Web browsers store cookies for the current session and automatically destroy them when the current session is closed.

Summary of the Gateway HTTP server search mechanism and specific actions using the *Cookie user and password* identification method:

Search method (in sequence)	If element found	If login fails
CGI parameter list		
Cookies sent by the client	Sever logs in the user	Server returns two Set-Cookie fields with empty values in order to clear cookies set on the browser side
Uses anonymous to log in	Server returns two Set-Cookie header fields to the client. These fields request the client to store user and password within its cookies.	

Cookie session ID

Summary of the Gateway HTTP server search mechanism and specific actions using the *Cookie session ID* identification method:

Search method (in sequence)	If element found and login correct	If login fails
Searches for user and password in the CGI parameters list	Server logs in the user and creates a new session	
Searches for a session ID in the cookies sent by the client	Server logs in the user with the <i>user</i> and <i>password</i> values saved for this session ID	Server returns a Set-Cookie field with an empty value in order to clear the cookie set on the browser side

When the Gateway HTTP server creates a new session, it returns a **Set-Cookie** field to the client. This field requests the client to store *sessionid* in its cookies list. When stored as a cookie, *sessionid* is automatically re-sent on each new request to the server.

Partner-Dependant identification

The server determines the authentication method to use from the request headers sent by the client software. HTTP clients can log in using any of the authentication methods supported by Gateway. Note that the HTTP server accepts all the connections having a supported identification method.

HTTP session ID limitation

You can limit the number of simultaneous opened sessions for a user.

A global configuration parameter, *sid_limit*, limits the number of sessions a user can have on the server.

Each time a user logs in, the value of this parameter is compared with the number of sessions registered for that user.

If the number of registered sessions is lower than the value of the configuration parameter, the user is allowed to login. Otherwise the login is refused.

You can find the *sid_limit* parameter in the Gateway [advanced parameters](#).

This parameter is only taken into account if you have set the HTTP server identification **Method** to *Cookie session id* or *CGI session id* in the configuration menu.

Directory listings

An HTTP directory listing returns to the client an HTML page containing a list of links to elements (files and subdirectories) contained in the directory. A directory listing is performed when the accessed URL refers to a directory.

The Gateway virtual file system can contain the following elements:

- **Directories**
- **Files:** Only available if the directory refers to a real system directory
- **Transfers:** References to Gateway transfers created in that directory. Only available if the directory is virtual (VFD). When displaying Gateway transfers, the server differentiates the available files in the directory listings as described in the following table:

If the Transfer type is:	Then the connected party can:
available	Retrieve the transfer (that is, the file) by clicking the transfer name, which is a hypertext link
unavailable	Display transfers in the current directory (similar to FTP). Transfers are displayed for informational purposes only and no hypertext links are required.

To fulfill these requirements, Gateway uses an HTML page to display directory contents. This page is called a listing template and includes elements that adapt the listing to the contents of the listed directory (for example, files, sub-directories, and transfers).

HTTP listing template files

The listing template files contain the HTML formats used to list directories. First, the server must determine which file to use as a pattern. The file path to this pattern is specified in the:

- CGate selected during login phase.
The value can be taken from one of the selected parent CGateGroups.
- Default template file defined in the HTTP configuration part of Gateway.
You can modify this value using the configuration menu.

Once found and loaded, the server stores the patterns in memory and reloads them only when their files were modified. If no file is defined or if the file cannot be loaded, an error is returned to the client.

When the Responder (server) retrieves the listing pattern, it processes the listing in the following sequence:

- 1 Substitutes the global symbols (for example "%") in the page with their respective values (refer to *Symbol substitution* later in this topic for more information)
- 2 Searches for special tags (defined in *Using special tags* below) and replaces them with each corresponding element of the listed directory
- 3 Returns the dynamically-generated pages to the Initiator (client)

Using special tags

Gateway recognizes the following special tags that you can use in the template files.

Tag	Comment
<PARENT> </PARENT>	Text between these tags is displayed only if the listed directory is not "/".
<DIR> </DIR>	Text between these tags is duplicated for each directory in the current directory. The following special names are also replaced: • @NAME@: directory name encoded for URL • @name@: directory name • @DATE@: Directory last modification date, per component: @DATE_Y@ (year), @DATE_M@ (month), @DATE_m@ (month as a decimal number), @DATE_D@ (day), @DATE_d@ (day as a decimal number), @DATE_T@: directory last modification time • @CONTENT@: number of files and number of available files from current directory. For example: 43 files (30 available)
<REAL_FILE> </REAL_FILE>	Text between these tags is duplicated for each file in the directory. It is only available under mounted directories. The following special tags are also replaced: • @FNAME@: file name encoded for URL • @fname@: file name • @DATE@: Directory last modification date, per component: @DATE_Y@ (year), @DATE_M@ (month), @DATE_m@ (month as a decimal number), @DATE_D@ (day), @DATE_d@ (day as a decimal number), @DATE_T@: directory last modification time • @SIZE@: file size • @ICON@: icon name associated with the directory path and filename, encoded for URL (for more information, refer to Configuration file).

Tag	Comment
<XFER_AVAIL> </XFER_AVAIL>	Text between these tags is duplicated for each available transfer in the current virtual directory. These tags are ignored inside real directories. The following special names are also replaced: • @FNAME@: transfer file name encoded for URL • @fname@: transfer file name • @DATE@: Directory last modification date, per component: @DATE_Y@ (year), @DATE_M@ (month), @DATE_m@ (month as a decimal number), @DATE_D@ (day), @DATE_d@ (day as a decimal number), @DATE_T@: directory last modification time • @SIZE@: file size • @LOCAL_ID@: transfer local identifier • @STATE@: transfer status • @ICON@: icon name associated with the directory path and filename, encoded for URL (for more information, refer to Configuration file).
<XFER_UNAVAIL> </XFER_UNAVAIL>	Text between these tags is duplicated for each unavailable transfer in the current directory. These tags are ignored inside real directories. Substitutions described above apply here

Symbol encoding

As a mechanism for the prevention of XSS (Cross-Site Scripting), Gateway offers the possibility to encode HTML entities according to the context in which they are used. Special tags like @NAME@ and @FNAME@ are already encoded for an URI context; however a higher level of control is desirable. For example, the directory or file name can be used in a different context than that of an URI; such as a JavaScript/HTML context. The proper encoding can be done by using a special tag with no default encoding, and suffixing it with an encoding context specifier; e.g. @name@forJsHtml(), @fname@forJsHtml().

The following types of encoding can be used:

- `forUri()`
URI context
- `forUriComponent()`
URI component context
- `forCssUri()`
CSS URI context
- `forCssString()`
CSS string context

- `forJsSource()`
JavaScript source context
- `forJsAttribute()`
JavaScript attribute context
- `forJsBlock()`
JavaScript block context
- `forJsHtml()`
JavaScript/HTML context
- `forHtml()`
HTML context

Caution Make sure every special tag that could be controlled by an attacker is using HTML entity encoding, either explicit or implicit. See the example below, and the sample templates delivered with the product.

Note For special tags with implicit encoding (@NAME@, @FNAME@), overwriting by specifying an explicit encoding is not supported.

For more detail regarding XSS (Cross-Site Scripting) and HTML entity encoding, please refer to the OWASP website:

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

(URLs are outside the Axway domain, and liable to change without notice).

Example

The following is an example of a directory listing template file.

```
<HTML>
<HEAD>
<TITLE> %u </TITLE>
</HEAD>
<BODY>
<H1> Index of %u </H1>
<PARENT>
<A href="%N%i">Parent directory (%n)</A> <BR>
</PARENT>
<HR> <PRE>
<DIR>
@DATE@ <A href="%U%/@NAME@/%i">@name@ .forHtml() </A>
</DIR>
Date Size State Name Local Ident
<REAL_FILE>
@DATE@ @SIZE@ <A href="%U%/@FNAME@/%i">@fname@ .forHtml() </A>
</REAL_FILE>
<XFER_AVAIL>
@DATE@ @SIZE@ @STATE@ <A href="%U%/@FNAME@?local_ident=@LOCAL_
```



```

IDENT@%&%I">@fname@ .forHtml() </A> @LOCAL_IDENT@
</XFER_AVAIL>
<XFER_UNAVAIL>
@DATE@ SIZE@ @STATE@ @fname@ .forHtml() @LOCAL_IDENT@
</XFER_AVAIL>
</PRE> <HR>
</BODY>
</HTML>

```

Downloading files

To download a file, the client must send a GET (or POST) request with a URL that matches either a [real file](#) or an available transfer.

As different transfers in a directory can have the same file name, you can use a [selection filter](#) to identify a specific transfer.

Uploading files

To upload a file onto the server, you must send the following elements:

- Path name (directory in which to receive the transfer)
- File name
- File contents
- Optional transfer parameters

The URL always contains the directory where the transfer is performed. The file name, however, is deduced as follows:

Attributing file names when uploading

HTTP request method	If URL format is	Then the filename is
PUT (not supported by browsers)	file path (directory + filename)	The last component of the file path
	directory name	The value of the CGI parameter <i>filename</i>

HTTP request method	If URL format is	Then the filename is
	file path (directory + filename)	The last component of the file path
POST (content type must be "multipart/form-data")	directory name	The value of the CGI parameter <i>filename</i> , if present, or the name of the file sent (extracted from the form field used to send the file)

Defining Axway MFT specific parameters

If the client is an Axway MFT product, optional parameters are available for more detailed communication. The parameters that you can set in the request include:

- Application (**P_APPLI**)
- Remote destination of the transfer (**P_DEST**)
- Origin of the transfer (**P_ORG**)
- Message to attach to the transfer (**P_MSG**)
- URL to use in file substitution if file was correctly received (**success_url**)
- URL to use in symbol substitution if the file could not be retrieved (**failure_url**)

These optional transfer parameters are sent through CGI parameters, either in the URL or in the request body (POST requests).

The syntax is:

parameter=value

Parameters are separated by an ampersand (&).

For examples of requests using the PUT and POST methods, refer to [Using HTTP in client mode - Sending files](#).

Note: The server checks and stores these parameters before the file transfer begins. This implies that they must be sent before the file itself. In other words, these parameters must be set either in the URL CGI parameters or in the CGI parameters preceding the file. If the reception results from a form submission, then the **file parameter must be positioned as the last parameter of the form**.

Restricting selection: File filters

As a virtual directory and a file name are insufficient to describe a Mailbox entry, Gateway provides an advanced file filter feature when downloading and listing requests.

A file filter comprises strings that contain **field=value** pairs. When you use HTTP, these strings are extracted from CGI parameters. Accepted values for the *field* portion of this pair are given in the table below:

Field	Accepted value
name	Pattern matching paths and protocol filenames
direction¹	IN, OUT, or BOTH. Not case-sensitive
date²	Exact date and time of the transfer, expressed in the format: <i>YYYYMMDDhhmmss</i> • Date fields (year, month, day of month) can be separated by "/" characters. • Time fields (hours, minutes, and seconds) can be separated by ":" characters. You can specify time without a date. Current day is considered the default date.
end_before or endbefore²	Date format (Transfers requested or performed earlier than the given date are not selected)
start_after or startafter²	Date format (Transfers requested or performed later than the given date are not selected)
origin or org¹	Origin of the transfer
destination or dest¹	Destination of the transfer
state¹	List of states in Gateway format. Each selected transfer must have one of the given states.
mvs_state or mvsstate¹	List of states in MVS format. Each selected transfer must have one of the given states.
ftpcs_state or ftpcsstate	List of states in FTPCS format. Each selected transfer must have one of the given states. The FTPCS state is the concatenation of the state in Gateway format and the state in InterPel MVS format.
local_ident or localident	Long integer corresponding to the unique transfer local identifier
xfer_ident or xferident	Long integer corresponding to a transfer identifier

(1) For real entries and directories, these fields are ignored.

(2) These fields are compared to the last modification date of the file.

Example 1: Filtering by date

Downloading:

```
GET /transfer/out/file1.txt?org=site1&start_after=20011205
```

Example 2: Filtering by transfer state

Directory listing:

```
GET /?name=*.txt&state=IC&org=site1
```

Notes: When you retrieve a transfer, if several records match the request, the most recent one is returned.

Example 3: Filtering by file type

Requesting an incorrect file type:

```
GET /transfer/file?state=C
```

In this example, the user tries to download the most recent entry called **file** and in the state **canceled**. Although this is syntactically correct, Gateway does not allow downloading of canceled files and returns the following response:

404 Not found

Sort/Order parameters

There are two parameters for sorting and setting the order of fields:

- `order (order=<order_field>)`

If negative, the sorting order descends.

If positive, sorting order ascends.

- `sort (sort=<sort_order_type>)`

Only one field can be chosen from the following:

"xfer_type", "size", "date", "local_id", "protocol", "rights", "direction", "state", "org", "dest",
"remote", "fname", "xfer_id", "route_state", "perm", "appli", "msg", "code", "route_from",
"route_to", "dup_from", "param1", "param2", "user_state", "local_ident", "name", "xfer_ident"

Gateway specific template files

This section describes the HTML pages returned by the server when:

- Errors occur
- Uploads complete successfully
- Uploads fail

Error pages

Error pages are HTML files returned to the client when an error occurs. The file returned depends on both the type of error and the languages accepted by the connected client.

When Web browsers connect to a server, they typically send an HTTP header field named *Accept Language* that contains an ordered list of accepted languages. The language used by the HTTP server is the first that matches one of its accepted languages. Use the parameter `err_languages` to configure the server-accepted languages, whose default values are `en` [English], and `fr` [French]. If none of the client languages match, the first default error language is used (en).

Error files are located in the sub-directories of the error directory (parameter `err_dir` whose default value is `<Gateway installation directory>/run_time/http/pages`). The sub-directories must have the same name as the accepted languages. Default directories are:

- `<Gateway installation directory>/run_time/http/pages/en` for English
- `<Gateway installation directory>/run_time/http/pages/fr` for French

The following table lists the files used:

Error pages displayed by return codes		
Filename	Return code	Error description
ERR_BAD_REQUEST	400	The request contains invalid or missing parameters
ERR_BAD_VERSION	505	Protocol version is not compatible
ERR_FORBIDDEN	403	Requested action is forbidden (read, write, and so on)
ERR_INTERNAL	500	An internal error occurred, for example a memory error, or a real or virtual file system access error
ERR_NOT_FOUND	404	Could not find requested URL
ERR_SESSIONID	403	Could not find received session ID (unknown or expired session)
ERR_UNAUTHORIZED	401 (Web auth)	Invalid credentials

Error pages are loaded in memory the first time you access them. The HTTP server reloads them only when their files are modified. [Symbol substitutions](#) apply to these files.

Upload success and failure pages

The following table summarizes the features of the upload success and failure pages:

Page	Returned when	Associated HTTP return code	Additional symbol substitutions
Upload success	File correctly received	200	"%x" replaced by the received CGI parameter <i>success_url</i> .
Upload failure	File incorrectly received	500	"%x" replaced by the received CGI parameter <i>failure_url</i> .

Configuration file

The configuration file is used to set up all Web server behavior that is based on URLs. The default file is `<Gateway installation directory>/run_time/http/server.ini`. The HTTP server automatically reloads the configuration when the file is modified.

Each line of the file must take the form:

```
command "argument" pattern [&]
```

Empty lines and lines beginning with "#" are ignored. Patterns can match file extensions ("`.txt`", "`.htm`"), file names ("`*-test-*.*`", "`*.tar.gz`") or URLs ("`/transfer/*`").

Configurable elements are:

- Content types returned to the client for the accessed URL
- Icons to use with directory listing entries
- Files for which symbol substitutions must be performed dynamically

Content types

The content type (also known as media type or MIME type) is a description of the data that the browser uses to determine how to process the received data.

Lines beginning with the `AddType` command define the **content-type** header field returned to a browser.

AddType must be followed by a content type and patterns. If one of the patterns matches the URL, the content type is returned to the client.

Example

The following example forces browsers to display all files ending with `.htm` or `.html` as Web pages. All other pages must be prompted for downloading:

```
AddType    "text/html"    *.htm *.html
```

```
AddType    "application/octet-stream"  *
```

Icons

When modifying templates for directory listings, you can display an icon for each kind of file present in the directory by using the `AddIcon` command, followed by the icon URL and patterns. When a pattern matches the URL of a listed element (file or transfer), the symbol `@ICON@` of the listing template is replaced by the URL icon.

Example

In the following example, the listing template `@ICON@` symbol is replaced by:

- `/icons/text.gif` for file names that end with `.txt`
- `/icons/binary.gif` for file names that end with anything else

```
AddIcon    "/icons/text.gif"    *.txt
```

```
AddIcon    "/icons/binary.gif"  *
```

Substituting symbols in files

The server can apply the same symbol substitution to downloaded files as the one applied to other elements such as error pages and listing templates.

To substitute symbols in files, add a line beginning with `AddHandler`, followed by the kind of action to perform: *internal*, and then the patterns.

Example

In the following example, a symbol substitution is applied to every real file ending with `.mhtml`.

```
AddHandler internal *.mhtml
```

Cross-Site Request Forgery prevention

As a prevention mechanism against Cross-Site Request Forgery (CSRF) attacks, Gateway is using a token-based mitigation approach. This mechanism is activated using the `csrf_token_active` configuration parameter:

```
peluconf set -s ft_http csrf_token_active 1
```

When activating the CSRF prevention mechanism, the HTTP templates must be updated to include the CSRF token for all state changing requests. The CSRF token itself is generated by Gateway and will be substituted in the HTTP template using a special symbol (`%s` for the non-encoded, and `%S` for the URI-encoded form - see the Symbol Substitution in HTML files section above). For example, an upload form could include the CSRF token as a hidden field:

```
<INPUT type=hidden name="token" value="%s" />
```

For more detail regarding CSRF (Cross-Site Request Forgery), please refer to the OWASP website:

- <https://owasp.org/www-community/attacks/csrf>
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

(URLs are outside the Axway domain, and liable to change without notice).

[Configuring protocols: HTTP options](#)

[About HTTP](#)

[Using HTTP in client mode](#)

[About CGates and CGateGroups](#)

[About Virtual File Directories](#)

Axway Gateway: Protocols

Using HTTP in client mode

[About the HTTP client](#)

[Client login: Identification methods](#)

[Sending files](#)

[Retrieving files](#)

[Retrieving a directory listing](#)

About the HTTP client

The HTTP client is used to perform transfers where Gateway is the connection Initiator. This includes the following tasks:

- Sending a file to a server
- Retrieving a file from a server
- Retrieving a directory listing from a server

Client login: Identification methods

You must set the login parameters for the [Remote Site](#) to be used to connect to the HTTP server.

For information on configuration, refer to [Configuring protocols: HTTP options](#).

Supported identification methods are described in the following table:

Identification method	Elements sent in Transfer Request	Comment
Anonymous	ID and password (set in the Remote Site) are not sent.	–
CGI	ID and password values are sent using the CGI parameters <i>user</i> and <i>password</i> .	Parameter names (<i>user</i> and <i>password</i>) are the same as those used by the HTTP server. You can modify these parameter names in the Gateway configuration menu.
Web	ID and password values are sent using the standard HTTP identification header field (Authorization).	–
Cookies	If a login request is sent, the cookies returned with the login response are sent back to the server with each Transfer Request, on condition that the HTTP session is not closed.	Received cookies are stored with their directory settings. They are only sent back if the Transfer Request is performed under the defined directory. Expiration dates are not taken into account.
NTLM	NTLM authentication is used to connect to the HTTP server.	see section <i>NTLM and NTLMv2 authentication</i> below.

NTLM and NTLMv2 authentication

NTLM and NTLMv2 authentication is supported in HTTP client mode.

Caution Only the authentication feature of NTLM is supported. Other NTLM features like session security are not supported; also, LM and LM2 authentication packages are not used. Depending on the Security Policies of the HTTP server, it is possible to require NTLM features not supported by Gateway; in this case, the NTLM authentication could **fail** or become **unstable**.

The NTLM version (NTLM or NTLMv2) cannot be negotiated between the client (Gateway) and the remote HTTP server; instead, it must be configured at both ends. With Gateway, this is done using the command line, for example:

```
peladm update_site -a REMOTE_SITE -ntlm_version NTLM2
```

By default, NTLMv2 is used.

Login request

An optional login request is performed the first time Gateway initiates an HTTP connection with a Remote Site. Login applies to both incoming and outgoing transfers.

The login request is performed under two conditions:

- **Cookies** client identification method must be in use
- Login path must not be empty (otherwise login is not performed, which is equivalent to performing the transfer anonymously)

Login requests are used with server CGI scripts that return a cookie to the Gateway client. For each transfer performed after the login phase, this cookie is sent back to the server. As a result, the login path must be the URL path to the server CGI script that identifies Gateway and returns the cookie, for example, `/cgi-bin/login.php`. You can also configure the HTTP method to use, either GET or POST.

On login, the *ID* and *password* parameters that you set in the **ID/Password to send** field of the Remote Site are sent using the CGI parameters *user* and *password*. You can add optional CGI parameters to the login request.

If login succeeds, the cookies returned by the server are stored in memory and sent with each Transfer Request. Once Gateway no longer needs to perform any more transfers with the server, the connection is closed and the cookies are deleted.

Example

Configuration for SERVER1 as a Remote Site

Site parameter	Value
IP address or machine name	webserver1
ID to send	login_1
password to send	password_1
identification method	Cookies
login path	/cgi-bin/login.php
HTTP method	POST
login CGI parameters	client=Gateway&site=SERVER1

A file transfer with SERVER1 will trigger a login request formatted as follows:

```
POST /cgi-bin/login.php HTTP/1.0
Content-Length: 64
Host: webserver1
Content-Type: application/x-www-form-urlencoded
User-Agent: Gateway/680
Connection: Keep-alive

user=login_1&password=password_1&client=Gateway&site=SERVER1
```

Sending files

When you use the HTTP protocol to send a file as Initiator, Gateway sends the following parameters from the Remote Site representing the Responder:

- Connection parameters (including `dest_address`, and `comm_type`)
- Identification method to use and all its related parameters (including `login_type`, `login_ident` and `login_password`).
- Axway MFT specific transfer attributes, if the remote server is an Axway MFT product (`is_monitor`)

Gateway can also send additional CGI parameters specified in the Remote Site `snd_msg` parameter.

Axway MFT specific transfer attributes

The table below describes the Transfer Request parameters required by Gateway:

Parameter	Description
(file_component)	Local file attributes including file name and path
(dir_name/file_name)	URL path to send: • directory name (POST) • directory name plus the file name (PUT)
(request_method)	HTTP method to use: POST or PUT The request method to use depends on the remote server configuration.

Parameter	Description
P_ORG (org)	
P_DEST (dest)	Transfer parameters
P_APPLI (appli)	When available, these parameters are sent if the Remote Site is configured as an Axway MFT product (is_monitor).
P_MSG (snd_msg)	
(ftp_command)	Additional CGI parameters to send

Examples of sending a file using the POST method

Example 1: CGI login type

The following example creates a Remote Site representing the server and:

- Defines the remote partner as an Axway MFT product
- Defines the login type as CGI
- Sends a file with the POST command

```
peladm create_site -a LOOP_HTTP1 -pr HTTP -check_login_
ident LOOP_HTTP1
-partner_loc_alias LOC_SITE -dest_address localhost/8080 -is_
monitor Y
-login_ident user -login_password passwd -login_type CGI
-snd_msg "site=loop_http1"

peltrans -da LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir O -
mode I
-file_name "machine.txt" -fc ./machine -snd_msg "message 1"
-ftp_command "fc=machine" -request_method POST
```

This command sends the following request:

```
POST / HTTP/1.0

Content-Length: 1397

Host: localhost:8080

Content-Type: multipart/form-data;

boundary=-----LOC_SITE-AxwayGW-20020304160043-
883517092697405393766851741526

User-Agent: Gateway/680

Connection: Keep-alive
```

Example 2: Cookies login type

The following example creates a Remote Site representing the server and:

- Defines the login path for the login request (*login_url*)
- Selects the login type *Cookies*
- Selects the GET login method for the login request
- Selects the POST request method for the transfer

```
peladm create_site -a LOOP_HTTP1 -pr HTTP -check_login_
ident LOOP_HTTP1
-partner_loc_alias LOC_SITE -dest_address localhost/8080 -is_
monitor N
-login_ident user -login_password passwd -login_type Cookies
-login_url "/cgi-bin/login.php" -login_method GET
-snd_msg "site=loop_http1"

peltrans -da LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir O -
mode I
-request_method POST -file_name "machine.txt" -fc ./machine
-snd_msg "message 1" -ftp_command "fc=machine"
```

This command sends the following requests:

GET /cgi-bin/login.php?user=user&password=passwd HTTP/1.0

Host: localhost:8080

User-Agent: Gateway/680

Connection: Keep-alive

and

POST / HTTP/1.0

Content-Length: 641

Cookie: sessionid=Synchrony3C839199A725636418

Host: localhost:8080

Content-Type: multipart/form-data;

boundary=-----LOC_SITE-AxwayGW-20020304161231-
134212818461445708661152709889

User-Agent: Gateway/680

Connection: Keep-alive

Examples of sending a file using the PUT method

Example 1: CGI login type

The following example creates a Remote Site representing the server and:

- Defines the remote partner as an Axway MFT product
- Defines the login type as CGI
- Sends a file with the PUT command

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -check_login_ident LOOP_HTTP1
-partner_loc_alias LOC_SITE -dest_address localhost/8080
-is_monitor Y -login_ident user -login_password passwd
-login_type CGI -snd_msg "site=loop_http1"

peltrans
-da LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir O -mode I
-file_name "machine.txt" -fc ./machine -snd_msg "message 1"
-ftp_command "fc=machine" -request_method PUT
```

This command sends the following request:

```
PUT /machine.txt?user=user&password=passwd&P_DEST=AxwayGW&P_ORG=LOC_SITE&
P_APPLI=FTP_B&P_MSG=message%201&fc=machine&site=loop_http1 HTTP/1.0

Content-Length: 4

Host: localhost:8080

User-Agent: Gateway/680

Connection: Keep-alive
```

Example 2: Cookies login type

The following example creates a Remote Site representing the server and:

- Defines the login path for the login request (*login_url*)
- Selects the login type *Cookies*
- Selects the POST login method for the login request
- Selects the PUT request method for the transfer

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -check_login_ident LOOP_HTTP1
-partner_loc_alias LOC_SITE -dest_address localhost/8080
-is_monitor N -login_ident user -login_password passwd
-login_type Cookies -login_url "/cgi-bin/login.php" -login_
```

```
method POST
-snd_msg "site=loop_http1"

peltrans
-da LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir O -mode I
-file_name "machine.txt" -fc ./machine -snd_msg "message 1"
-ftp_command "fc=machine" -request_method PUT
```

This command sends the following request:

```
POST /cgi-bin/login.php HTTP/1.0

Content-Length: 25

Host: localhost:8080

Content-Type: application/x-www-form-urlencoded

User-Agent: Gateway/680

Connection: Keep-alive

user=user&password=passwd
```

and

```
PUT /machine.txt?fc=machine&site=loop_http1 HTTP/1.0

Content-Length: 4

Cookie: sessionid=Synchrony3C8399622152645421

Host: localhost:8080

User-Agent: Gateway/680

Connection: Keep-alive
```

Retrieving files

When you use the HTTP protocol to retrieve a file from a server, Gateway extracts the following parameters from the Remote Site:

- Connection parameters (`dest_address`, `comm_type`)
- Identification method to use and all its related parameters (`login_type`, `login_ident`, `login_password`)
- Additional CGI parameters in Remote Site (`snd_msg`)

Gateway extracts the parameters from each Transfer Request as described in the Axway MFT specific transfer attributes table.

Examples of retrieving a file using the POST method

Example 1: CGI login type

The following example creates a Remote Site representing the server and:

- Defines the login type as CGI
- Retrieves a file with the POST command

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -dest_address localhost/8080
-check_login_ident LOOP_HTTP1 -login_ident user -login_
password passwd
-login_type CGI -snd_msg "site=loop_http1"

peltrans
-da LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir I -mode I
-file_name "dispo.txt" -fc retrieve.txt dir_name /dispo -request_
method POST
```

This command sends the following request:

```
POST /dispo/dispo.txt HTTP/1.0

Content-Length: 41

Host: localhost:8080

Content-Type: application/x-www-form-urlencoded

User-Agent: Gateway/680

Connection: Keep-alive

user=user&password=passwd&site=loop_http1
```

Example 2: Cookies login type

The following example creates a Remote Site representing the server and:

- Defines the login path for the login request (*login_url*)
- Selects the login type *Cookies*
- Selects the GET login method for the login request
- Selects the POST request method for the Transfer Request

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -dest_address localhost/8080
-check_login_ident LOOP_HTTP1 -login_ident user -login_
password passwd
-login_type Cookies -login_url "/cgi-bin/login.php" -login_
```



```
method GET
-snd_msg "site=loop_http1"

peltrans
-oa LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir I -mode I
-file_name "dispo.txt" -fc retrieve.txt dir_name /dispo
-request_method POST
```

This command sends the following requests:

```
GET /cgi-bin/login.php?user=user&password=passwd HTTP/1.0
```

```
Host: localhost:8080
```

```
User-Agent: Gateway/680
```

```
Connection: Keep-alive
```

and

```
POST /dispo/dispo.txt HTTP/1.0
```

```
Content-Length: 15
```

```
Cookie: sessionid=Synchrony3C84FA4244304328
```

```
Host: localhost:8080
```

```
Content-Type: application/x-www-form-urlencoded
```

```
User-Agent: Gateway/680
```

```
Connection: Keep-alive
```

Examples of retrieving a file using the GET method

Example 1: CGI login type

The following example creates a Remote Site representing the server and:

- Defines the login type as CGI
- Retrieves a file with the GET command

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -dest_address localhost/8080
-check_login_ident LOOP_HTTP1 -login_ident user -login_
password passwd
-login_type CGI -snd_msg "site=loop_http1"

peltrans
-oa LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir I -mode I
-file_name "dispo.txt" -fc retrieve.txt dir_name /dispo
-request_method GET
```

This command sends the following request:

```
GET /dispo/dispo.txt?user=user&password=passwd&site=loop_http1 HTTP/1.0

Host: localhost:8080

User-Agent: Gateway/680

Connection: Keep-alive
```

Example 2: Cookies login type

The following example:

- Creates a Remote Site representing the server
- Defines the login path for the login request (*login_url*)
- Selects the login type *Cookies*
- Selects the POST login method for the login request
- Selects the GET request method for the Transfer Request

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -dest_address localhost/8080
-check_login_ident LOOP_HTTP1 -login_ident user -login_
password passwd
-login_type Cookies -login_url "/cgi-bin/login.php" -login_
method POST
-snd_msg "site=loop_http1"

peltrans
-oa LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir I -mode I
-file_name "dispo.txt" -fc retrieve.txt dir_name /dispo
-request_method GET
```

This command sends the following requests:

```
POST /cgi-bin/login.php HTTP/1.0

Content-Length: 25

Host: localhost:8080

Content-Type: application/x-www-form-urlencoded

User-Agent: Gateway/680

Connection: Keep-alive

user=user&password=passwd

and

GET /dispo/dispo.txt?site=loop_http1 HTTP/1.0

Cookie: sessionId=Synchrony3C85E636C6730196
```

Host: localhost:8080

User-Agent: Gateway/680

Connection: Keep-alive

Retrieving a directory listing

When you use the HTTP protocol to retrieve a directory listing from a server, the **peltrans** parameter **type** must be set to **dir**.

Example of retrieving a directory listing

The following example:

- Creates a Remote Site representing the server
- Defines the login type as CGI
- Retrieves a directory listing with the GET command

```
peladm create_site
-a LOOP_HTTP1 -pr HTTP -dest_address localhost/8080
-check_login_ident LOOP_HTTP1 -login_ident user -login_
password passwd
-login_type CGI -snd_msg "site=loop_http1"

peltrans
-oa LOOP_HTTP1 -dest AxwayGW -appli FTP_B -dir I -mode I
type dir dir_name /dispo -request_method GET
```

This command sends the following request:

GET /dispo/dispo.txt?user=user&password=passwd&site=loop_http1 HTTP/1.0

Host: localhost:8080

User-Agent: Gateway/680

Connection: Keep-alive

[About HTTP](#)

[Configuring the HTTP environment](#)

[Working with Remote Sites](#)

[Configuring protocols: HTTP options](#)

[Using HTTP in server mode](#)

Using HTTP/HTTPS proxies

[Configuring proxies](#)

[HTTP/HTTPS proxy details](#)

Configuring proxies

The configuration of a Proxy object is defined in the file `proxy.ini`.

To use a Proxy, add `@proxy_name` to the TCP address of the Remote Site.

The TCP address of Sites (`IP@proxy/port`) must not exceed 63 characters.

Configuring HTTP proxies

Configure HTTP proxies as follows:

```
[proxy_name]
protocol = HTTP
address = proxy.domain/8080
user = login
password = pass
```

Configuring HTTPS proxies

Configure HTTPS proxies as follows:

```
[proxy_name]
protocol = HTTPS
address = proxy.domain/8080
user = login
password = pass
```

HTTP/HTTPS proxy details

HTTP proxies

To proxy HTTP through Web proxies, the basic procedure is as follows:

1. Connect to the Web proxy.
2. Send the HTTP requests formed with the full URL (http://) and the proxy identification parameters.
3. Wait for the response from either the Web proxy or the remote Web server.

HTTPS proxies

To proxy HTTPS through Web proxies, the basic procedure is as follows:

1. Connect to the Web proxy.
2. Send an HTTP connect request for the distant HTTPS server.
3. Wait for the proxy response.
4. Initiate the SSL connection with the remote server.
5. Send the HTTP requests.

To connect to an HTTPS server through a web proxy, the HTTP protocol and the Remote Site representing the HTTPS server must be set to use TLS security.

Axway Gateway: Protocols

Adding non-standard HTTP headers

Introduction

Some HTTP servers require specific, non-standard headers when they receive files. For example, they may require an HTTP header such as *X-Transmit-ID*.

The HTTP header extension in Gateway enables you to add such headers in HTTP requests.

The header extension is available for the HTTP protocol only (not for AS2_HTTP or RN_HTTP).

Supported mode, direction and request methods

- Gateway acting as client
- Mode: Initiator
- Direction: Send
- Supported request methods: POST and PUT (for sending files only)

Specifying non-standard HTTP headers

Specify the required HTTP headers, along with any CGI parameters, in the Transfer Request `snd_msg` parameter. If you are using the GUI, enter this information in the **Message to send** parameter on the [Details](#) tab of the Transfer Request.

When you submit the Transfer Request, the non-standard headers found in `snd_msg` will be sent together with the standard HTTP headers.

How to complete the `snd_msg` parameter

When completing the Transfer Request `snd_msg` parameter, observe the following rules:

- The HTTP header string must start with `[xs4hh]` and end with `[/xs4hh]`. Anything that follows the ending tag `[/xs4hh]` is treated as CGI parameters.
- `[xs4hh]` must be the very first string in the `snd_msg` parameter.
- The HTTP header pairs within the string must use:
 - "=" to separate the HTTP header id from its value
 - "&" to separate HTTP header id/value pairs

Example: `X-Transmit-ID=doc1&X-Bla=doc2`

- The `[xs4hh]` and `[/xs4hh]` separators must not be used for any other purpose (for example, for CGI parameter values).

All of these rules must be respected. If not, Gateway treats all the `snd_msg` strings as CGI parameters.

Note that the convention for non-standard headers is to prefix the header name with X-.

Formatting a string containing HTTP headers and CGI parameters

Use the following format:

```
[xs4hh]X-Id1=v1&X-Id2=v2[/xs4hh]xfer_par1=val1&xfer_par2=val2
```

Escape character

If you need to use one of the HTTP header pair separators ("=" or "&") inside a header field value, you need to use the escape char: "%".

Example: To include "=234" at the end of a header field value, format the string as follows:

```
[xs4hh]X-Id1=v1%=234[/xs4hh]
```

The "=" is treated as the assignment separator and the "%=" is translated into the ordinary equals part of the field value (`v1=234`).

Note: Also use the "%" character if you need to escape the percentage character itself.

[Using HTTP in client mode](#)

[About HTTP](#)

[Transfers: Parameters List](#)

Axway Gateway: Protocols

OFTP (Odette)

About OFTP

[OFTP](#)

[Gateway objects used for OFTP](#)

[OFTP services](#)

[How OFTP processing works](#)

OFTP Introduction

OFTP (Odette File Transfer Protocol) is an end-to-end protocol designed to transfer files between two partners, irrespective of the network support, hardware, and software architecture of the host computers.

Gateway implements RFC5024, the Internet standard specification for OFTP. It supports OFTP versions R2.0 and all preceding versions (R1.2, R1.3 and R1.4).

Gateway objects used for OFTP

Site

In Gateway, the Site represents a communication entity.

A Local Site represents the communication entity on the local platform, while a Remote Site represents the communication entity on the remote platform. Each Site is characterized by a set of attributes such as:

- Protocol Identifier (for example; OFTP SSID and password)
- Physical network address (IP address)
- Protocol session parameters (Data Exchange buffer size, send/receive capability, restart capability, flow control windows size, secured authentication, and so on)
- Authentication certificate

OFTP allows data transfers in both directions: the Site initiating the connection establishment is termed *Initiator*, the other is *Responder*. The Site sending the data is termed the *Sender*, the other is the *Receiver*. Each Site has one of the following roles:

- Initiator/Sender
- Initiator/Receiver
- Responder/Sender
- Responder/Receiver

You can define as many Local Sites and Remote Sites as required.

Trading Partner

A Trading Partner is the original file sender or the final file receiver (destination).

A **local** Trading Partner is located on the local platform. A **remote** Trading Partner is located on the remote platform.

In OFTP, the Trading Partner is essentially associated with file security services (file signature, file encryption, file compression, and signed end-to-end response). Each Trading Partner has a set of attributes concerning these services, such as:

- Trading Partner ID (for example OFTP SSID)
- Signing certificate and algorithm to use
- Encryption certificate and algorithm to use
- Compress algorithm to apply
- Constraints on the required file security

Application

An Application object is used to indicate the file attributes of the transmitted file and any possible desired transformation (transcoding, padding, and so on) to apply to the data before its transmission to the underlying network. Its main purpose is to allow Gateway to transform a local platform-specific physical file into a platform-independent OFTP virtual file for transmission or vice versa.

OFTP services

Supported OFTP functions

OFTP provides the following functions:

- File transmission and reception
- Transfer restart
- Data compression
- Change Direction (CD)

- Flow control
- End-to-End-Response (EERP) – signed and non-signed
- Mutual authentication
- File security services (encryption, signature, compression)

The following table describes each of these functions and indicates the parameters to set for the required Gateway objects.

Function	Description	Gateway object & parameter
File transmission	To transfer data, you must first establish a logical link with the remote partner.	Remote Site: Role: Select <i>Initiator/Receiver</i> or <i>Responder/Receiver</i>
File reception		Remote Site: Role: Select <i>Initiator/Sender</i> or <i>Responder/Sender</i> .
Transfer restart	You can restart a transfer that is interrupted before completion. Interruption closes the file and disables the protocol and network connections. The Initiator always initiates the restart, and the Responder decides on the synchronization point from which to resume the transfer.	Remote Site: Select the Restart allowed option.
Data compression	OFTP provides horizontal data compression (suppression of repeated characters in a record).	Remote Site: Select the Compression allowed option.

Function	Description	Gateway object & parameter
Change Direction (CD)	Enables you to change the transfer direction dynamically. CD can be seen as a <i>token</i> exchanged between partners. The protocol uses the token to change the direction of file transmission within a session, reversing the roles of the partners involved (Sender becomes Receiver, and Receiver becomes Sender). Only the partner with the token (the Initiator) has the right to send a file (to the Responder). The OFTP file transfer stack in Gateway is half duplex, which means that the same Site can be used for consecutive file transfers in both directions. Gateway connects as Initiator/Sender (as defined by the protocol) and follows with a Change Direction command that allows the Responder to send a file.	Transfer Request: To receive a transfer in Initiator mode, you must specify Mode = <i>Initiator</i> and Direction = <i>Receive</i>.
Flow control	OFTP uses a Credit feature as an acknowledgment mechanism for regulating data flow. The credit limit specifies the number of Data Exchange Buffers that can be sent before the Initiator has to wait for a Credit command from the Responder.	Remote Site: Credit limit is negotiated in the start session phase, and defined via the Network credit parameter.
End-to-End Response (EERP)	An OFTP session has two possible EERP styles: French and International. These describe the position of the destination and originator fields in the EERP: • French mode: the originator field precedes the destination field • International mode: the destination field precedes the originator field There are two methods of sending the EERP: • Automatic EERP sending by its final destination can be activated via the acknowledgment option parameter in the Remote Site. • Manual EERP sending can be initiated via submitting an EERP request through API, command line or GUI.	Remote Site: Set the EERP Style to <i>French</i> or <i>International</i>. Set the acknowledgment option to <i>By monitor</i> or <i>By user</i> to activate automatic EERP sending by final destination.

Function	Description	Gateway object & parameter
Mutual authentication	<i>Only for OFTP R2.0 or higher.</i> Enables mutual authentication of the Initiator and Responder.	Local Site and Remote Site: Specify the authentication certificate to use in Authentication certificate: Type and Name .
File security	<i>Only for OFTP R2.0 or higher.</i> Enables user payload to be signed, encrypted, and/or compressed by the sender and the payload signature to be verified, decrypted and/or decompressed by the receiver.	Trading Partner: Define a Trading Partner for the sender and the receiver with appropriate security parameters.

Session establishment

The transfer begins with the physical connection of two Sites, using the network address information defined in the Remote Site.

Once the network connection is established, the protocol session follows. During this phase, the Initiator and Responder send their identification and password in turn to mutually identify themselves.

Session secured authentication

If the Remote Site is OFTP R2.0 or higher, and an authentication certificate has been specified, OFTP continues with the mutual authentication phase.

Using the Initiator's public authentication certificate key, the Responder sends an encrypted challenge to the Initiator. The Initiator decrypts this challenge with its own authentication certificate, and sends back the decrypted challenge to authenticate itself.

The process is then repeated but this time to authenticate the Responder. Using the Responder's authentication certificate, the Initiator sends an encrypted challenge to the Responder. The Responder decrypts this challenge with its own authentication certificate, and sends back the decrypted challenge to authenticate itself.

File security services

The actual file transfer starts with a file identification phase, where the sender specifies the files characteristics, its originator and its destination, and whether any file security services are in use.

The user payload (file to be transferred) can be secured via file security services. The sender can ask for the file to be signed, encrypted, and/or compressed before its transmission. Upon reception, the receiver can automatically verify the payloads signature, decrypt and/or decompress its content.

You can only use file security if:

- The Remote Site supports OFTP R2.0 or higher
- The originator and destination are predefined Trading Partners

Transfer identification

OFTP uses the following information to uniquely identify a transfer:

Parameter	Description
File name	26-character string associated with each transfer
Date and Time	Date and time of the Transfer Request
Destination	Protocol name of the receiver Site
Originator	Protocol name of the sender Site

File and transfer attributes

The file contents are transmitted over the network in data blocks that you define in the Application object. Automatic code conversion of the file contents can be carried out if required in the Application object.

Routing

OFTP provides automatic Store-and-Forward, allowing a transfer from one end Site to another via intermediate Sites. In other words, if the current file receiver is not the final destination, the file can be automatically routed to the next node. With Store-and-Forward, an end-to-end transfer acknowledgment is supplied (via an EERP command) by the final destination to notify the initial sender that the file has reached its final destination.

File transfer phases

OFTP processes file transfers in the following sequence:

1	Protocol connection After the establishment of the network connection, the Responder sends an OFTP Ready Message. The Initiator sends its identification code and password to identify itself. The Responder then identifies itself in the same way.
2	Authentication <i>This phase only applies to OFTP R2.0.</i> During the authentication phase, Initiator and Responder mutually authenticate each other using their respective authentication certificates.
3	Selection of the information set to be transferred Requested by the Initiator, and validated and negotiated by the server. The selection phase incorporates file-opening services.

4	Data transfer After selection, the data is sent with a header and flow control regulation, if requested.
5	End of transfer When the data is sent, a command containing checking information (including number of records, and bytes) informs the receiver of the end of data transfer. The receiver checks the validity of the information received and then accepts or rejects the file. The end of transfer phase incorporates file closing and de-selection services.
6	Protocol connection release Initiated by the sender, after an end of transfer, Change Direction, or EERP protocol message.

In a typical file transfer sequence, phases 3 to 5 are repeated until all the files are transferred.

How to use a Gateway Application object with OFTP

At the beginning of the transfer, OFTP sends a start file identification command (SFID) comprising two text fields, FILE DATASET NAME and USER MESSAGE, available for use by the client Application. As other user Applications may already reserve these two fields, Gateway supplies the user exit function [ExitOdtGetAppli\(\)](#) as an alternative for transmitting the Application name to the receiver.

Summary of methods to transmit the Application name

Application method	Application name is retrieved from	Comment
User message field	USER MESSAGE field of the SFID command	If the Application transmitted with the SFID command does not exist, the Transfer Request is rejected.
File name	FILE DATASET NAME field of the SFID command	
User exit	Exit function	Use this method when User message field and File name are not available.

Specify the method to use via the Remote Site **Application method** (`-get_appli_method`) parameter.

OFTP partner identification

OFTP partner identification uses a structure based on the ISO - IS6523 specifications. The SIO (Structure for the Identification of Organization) is given in the table below. The format is not mandatory, but you must complete the **Protocol identifier** field (`-proto_ident` parameter) in the Remote Site.

Field	Description
OFTP Identifier	Value 0 (1 byte)

Field	Description
Organization Identifier: • ICD (International Code Designator) • Organization Code (*) • CSA (Computer Sub Address) (*) Code provided by the OFTP codification Group (Group 7)	Generally, the company registration number: • 4 bytes (numeric) • 14 bytes (A to Z, 0 to 9, space and hyphen) • 6 bytes (at the discretion of the company)

You can use this structure in prefixed or non-prefixed notation. Non-prefixed notation comprises only the Organization Code and the CSA.

Examples

prefixed notation: O1207SOPRA-PARIS UNIX12

non-prefixed notation: SOPRA-PARIS UNIX12

How OFTP processing works

OFTP protocol messages

OFTP protocol messages comprise commands and responses called File Protocol Data Units (FPDUs). Each FPDU starts with a single ASCII character representing its identification, followed by the fields defined by the OFTP protocol. The following table lists the OFTP FPDUs:

OFTP FPDU	Meaning	Header (ASCII)	Sent by	Usage
SSRM	Ready Message	I	Responder	Notify that server is ready
SSID	Identification Password & Profile	X	Initiator & Responder	Identification and protocol negotiations
CD	Change Direction	R	Sender	Change session direction
ESID	End of Session	F	Sender	Terminate session
SFID	Start File Information	H	Sender	Start file transfer
SFPA	Send File Positive Answer	2	Receiver	Acknowledge file transfer

OFTP FPDU	Meaning	Header (ASCII)	Sent by	Usage
SFNA	Send File Negative Answer	3	Receiver	Refuse file transfer
DATA	File Data	D	Sender	Send file data
EFID	End of File Information	T	Sender	End file transfer
EFPA	End of File Positive Answer	4	Receiver	Acknowledge file transfer
EFNA	End of File Negative Answer	5	Receiver	Refuse end of file transfer
CDT	Set Credit	C	Receiver	Release data flow
EERP	End-to-End Response	E	Receiver	Acknowledge file transfer from end to end
RTR	Ready To Receive	P	Receiver	Acknowledge previous EERP
SECD	Security Change Direction	J	Initiator & Responder	CD during authentication phase
AUCH	Authentication challenge	A	Initiator & Responder	Sending an authentication challenge
AURP	Authentication Response	S	Initiator & Responder	Sending an authentication response
NERP	Negative End-to-End Response	N	Receiver	Negatively acknowledge file transfer from end to end

Typical OFTP protocol FPDU exchange

The following is a summary of a typical OFTP protocol FPDU exchange for sending a file with an end-to-end acknowledgment:

Site A	Network	Site B	Comments
NET_CON_REQ	---->	NET_CON_IND	Network connection.
NET_CON_CONF	<----	NET_CON_RESP	

Site A	Network	Site B	Comments
SSID (Initiator)	<---- ----> <----	SSRM SSID (Responder)	A is sender, B is receiver. Receiver is ready. Sender sends ID. Receiver replies with ID.
SECD AURP	----> <---- ---->	AUCH	Start Initiator authentication. Send encrypted challenge. Send back decrypted challenge.
AUCH	<---- ----> <----	SECD AURP	Start Responder authentication. Send encrypted challenge. Send back decrypted challenge.
SFID	----> <----	SFPA	File transmission request. Receiver accepts it.
DATA DATA DATA DATA DATA	----> ----> <---- ----> ----> <---- ---->	CDT CDT	Sender sends file data and waits for credit responses from Receiver.
EFID CD	----> <---- ---->	EFPA (change_dir= Yes)	Sender informs the partner of the end of data transmission. Receiver accepts end of transfer and requests a Change Direction to send end-to-end acknowledgment. Then sender can only change direction or close session. A becomes receiver, B becomes sender.
RTR	<---- ----> <----	EERP CD	B sends acknowledgment (EERP). A sends RTR (Ready To Receive). Then B has nothing further to do, so it changes direction. A becomes sender, B becomes receiver.
ESID NET_DISCON_IND	----> <----	NET_DISCON_REQ	A has nothing further to do either, so it closes session and B closes network connection.

OFTP transfer phases

There are six phases involved:

- Start session
- Authentication
- Start file transfer
- File transfer
- End of file transfer
- Change Direction (optional)

1. Start session phase

The start session phase begins after the network connection. The Site that initiates the network connection is referred to as the *Initiator* while its corresponding partner Site is the *Responder*.

- The Responder sends an SSRM command, notifying the Initiator that it is ready.
- The Initiator sends a SSID command to identify itself.
- The Responder checks the incoming SSID from the Initiator, negotiates the parameters for the protocol, then sends its SSID command to identify itself.

2. Authentication phase

This phase only applies to OFTP R2.0.

- The Initiator sends an SECD to the Responder.
- The Responder sends an AUCH (encrypted challenge) to the Initiator.
- The Initiator decrypts the AUCH and sends an AURP to authenticate itself.
- The Responder sends an SECD to the Initiator.
- The Initiator sends an AUCH (encrypted challenge) to the Responder.
- The Responder decrypts the AUCH and sends an AURP to authenticate itself.

The session is now ready for the exchange: data transfer, end-to-end acknowledgment, or Change Direction.

3. Start file transfer phase

To start the file transfer, the:

- Initiator sends an SFID command.
- Responder accepts the Transfer Request by sending an SFPA response.

After the start file transfer phase, data flows from the Initiator (sender) to the Responder (receiver).

4. File transfer phase

During the file transfer phase, the Initiator can send a specific number of Data Exchange Buffers before it receives acknowledgment from the Responder. This number is negotiated in the start session phase, as the *credit*. For example, if the credit value is set to 4, the Initiator can send four DATA commands.

- The Initiator waits for the Responder to release the data flow by sending the CDT response, before it can send additional DATA commands.
- On receiving the CDT response, the Initiator sends DATA commands until it reaches either the end of file, or the credit value.

5. End of file transfer phase

When all data is transmitted, the Initiator must send a command to indicate the end of the file transfer.

- The Initiator sends an end-of-file command: EFID.
- The Responder acknowledges the end of transfer by sending one of the following responses:
 - EFPA (positive): the file transfer is accepted
 - EFNA (negative): the file transfer is rejected
 - EFPA with CD (Change Direction) request.
- If the Responder returns an EFPA with CD request:
 - The Initiator must issue a CD command to allow the partner to send an end-to-end response command (EERP)
 - It does not issue any more commands until it receives the Ready to Receive command (RTR).

6. Change Direction phase

Both the Sender and Receiver can initiate a change in the file transfer direction. If accepted, the Sender becomes the Receiver and vice versa.

- The sender issues a CD command to change the file transfer direction. The sender can issue a CD command after each file transfer.

You can activate a different connection hold timer by setting `odt_different_conn_hold_timers` to 1. In this case, the connection will be closed by a timer set by the `odt_conn_hold_send_delay` setting.

- The receiver sends an EFPA response with the Change Direction field set to Yes to request a change in the file transfer direction. On receiving this command, the sender must issue a CD command or close the session.

Configuring Gateway for OFTP

[Activating OFTP protocol](#)

[Gateway objects required for OFTP](#)

[Creating a Local Site](#)

[Creating a Remote Site](#)

[Creating an Application](#)

[Creating Trading Partners](#)

Activating OFTP protocol

Before you can use Gateway to transfer files using OFTP, you must activate the protocol in the configuration menu.

Select **Connectivity > Legacy protocols > OFTP (Odette)**. Then select either **TCP/IP**.

Follow the procedure given in [Configuring protocols](#).

Gateway objects required for OFTP

Before you submit a Transfer Request, you must create and configure the following Gateway objects:

- **Local Site**
When you install Gateway, it creates a default Local Site for all protocols, including OFTP. However, you can create as many Local Sites as necessary.
- **Remote Site**
Define your partner Site and communication attributes in the Remote Site object.
- **Application**
Define at least one Application.

If you use any of the OFTP R2.0 file security services (file signature, encryption, compression, signed EERP or NERP), you also require Trading Partner objects.

- **Trading Partners**
Define at least one local Trading Partner and one or more remote Trading Partners.

Use either the GUI or the `pe1adm` online command to create these objects. The following examples use the `pe1adm` online command.

Creating a Local Site

Example:

The following command creates a Local Site with protocol ident O1207SOPRA-PARIS UNIX1:

```
peladm create_loc_site -alias loc_odette
                        -pi 01207SOPRA-PARIS UNIX1
                        -comments "Local Site Odette"
                        -auth_obj_type SECS_ALIAS
                        -auth_obj_name mycert
                        -auth_obj_password mypass
```

An authentication certificate is required if you use any of the OFTP R2.0 file security services (file signature, file encryption, file compression or signed EERP/NERP).

Creating a Remote Site

You must create Remote Sites for your partners before you can make a file transfer.

If you do not explicitly supply attributes, Gateway uses default values. If you do not specify a group when creating a Remote Site, Gateway assigns the default group name GDEFAULT.

An authentication certificate is required if you use any of the OFTP R2.0 file security services (file signature, file encryption, file compression or signed EERP/NERP).

RFC5024 compliancy with OFTP 1.3

When contacting a partner using OFTP 1.3, by default Gateway assumes that the partner is RFC2204 compliant. Gateway sends the protocol release level SSIDLEV=1 when contacting the partner.

If you want to define a partner as being RFC5024 compliant, you must add the string "R1.4" in the Remote Site `comments` field. In this case, Gateway sends SSIDLEV=2 when contacting the partner.

Example 1: Remote Site for TCP/IP

The following command creates a Remote Site that uses TCP/IP and has the following characteristics:

- IP address: 193.56.234.20
- Port: 22222
- Block size: 3500
- Number of data blocks between two acknowledgments: 5
- Maximum number of simultaneous server connections: 10
- Protocol ident (non-prefixed notation): *SOPRA-PARIS UNIX12*
- Password to send: *UNIX12*
- Password to check: *UNIX1*
- Local Site used for identification: *LOC_Odette*
- Protocol field used to transmit Gateway Application name: *FILE DATASET NAME*
- EERP style: International
- Acknowledgment option: Automatically send an acknowledgement request

```
peladm create_site -alias TCP_odette
                  -comments "Remote Site: Odette on TCP/IP"
                  -protocol Odette
                  -pi SOPRA-PARIS UNIX12
                  -passwd UNIX12
                  -check_password UNIX1
                  -partner_loc_alias LOC_Odette
                  -get_appli_method AIF
                  -comm_type TCP
                  -dest_address 193.56.234.20/22222
                  -data_size_max 3500
                  -check_window 5
                  -resp_conn_max 10
                  -version R2.0
                  -auth_obj_type SECS_ALIAS
                  -auth_obj_name yourcert
                  -eerp_style IN
                  -acknowledgment_option ack_by_monitor
```

Creating an Application

You must define an Application to send or receive OFTP files.

The only Application parameter specific to the OFTP protocol is the ability to set the Transfer Attribute **Record format** (`-x_rec_fmt`) to *Text*.

Example 1: Application to transfer ASCII files

The following command creates an Application to transfer ASCII files of 255-byte fixed-length records, using the default group name:

```
peladm create_appli -name fix255
                  -comments "fixed record length: 255"
                  -rec_fmt F
                  -rec_len 255
                  -data_code A
                  -x_rec_fmt F
                  -x_rec_len 255
                  -x_data_code A
                  -dir_path <your received files directory>
```

Example 2: Application to transfer EBCDIC files

The following command creates an Application to transfer EBCDIC files of variable record length, with maximum record length of 255 bytes, and the group name GRVAR:

```
peladm create_appli -name var255
                    -group GRVAR
                    -comments "variable record length: 255"
                    -rec_fmt V
                    -rec_len 255
                    -data_code E
                    -x_rec_fmt V
                    -x_rec_len 255
                    -x_data_code E
                    -dir_path <your received files directory>
```

Creating Trading Partners

If you use any of the file security services (file signature, encryption, compression, signed EERP or NERP), you must create both local and remote Trading Partners. Define at least one local Trading Partner. You must also define one or more remote Trading Partners.

Example 1: Local Trading Partner

```
peladm create_tradeport -name "my OFTP SSID"
                        -alias myoftp
                        -format CMS
                        -is_local Y
                        -encryption_obj_type { SECS_ALIAS | XPP_
ENTITY | XPP_CERTIFICATE }
                        -encryption_obj_name "Sopra Sample User
1"
                        -sign_obj_type { SECS_ALIAS | XPP_ENTITY
| XPP_CERTIFICATE }
                        -sign_obj_name "Sopra Sample User 1"
                        -sign_obj_password user1
```

Example 2: Remote Trading Partner

```
peladm create_tradeport -name "your OFTP SSID"
                        -alias youroftp
                        -format CMS
                        -is_local N
                        -remote_site r_odette
                        -sign N
                        -signing_algo SHA1withRSA
                        -encrypt N
                        -encryption_algo { AES256 | 3DES }
                        -compress N
                        -compress_algo zlib
                        -encryption_obj_type { SECS_ALIAS | XPP_
```

```
ENTITY | XPP_CERTIFICATE }  
  
2"  
  
| XPP_CERTIFICATE }  
  
-encryption_obj_name "Sopra Sample User  
-sign_obj_type { SECS_ALIAS | XPP_ENTITY  
-sign_obj_name "Sopra Sample User 2"  
-force_signed { Y | N }  
-force_encrypted { Y | N }
```

OFTP identification code

RFC - Identification code description:

The Initiator (sender) and Responder (receiver) participating in an ODETTE-FTP session are uniquely identified by an Identification Code based on [ISO-6523]. This ISO describes the structure for the identification of organizations and organization parts (SIO).

The locations are considered to be adjacent for the duration of the transmission.

The SIO has the following format:

```
o-----  
--o  
| Pos | Field | Description | Format |  
|-----+-----+-----+-----+-----+  
--|  
| 0 | SIOOID | ODETTE Identifier | F X(1) |  
| 1 | SIOICD | International Code Designator | V 9(4) |  
| 5 | SIOORG | Organisation Code | V X(14) |  
| 19 | SIOCSA | Computer Subaddress | V X(6) |  
o-----  
--o
```

Table of fields

Field	Description
SIOOID ODETTE Identifier Character	Value: 'O' Indicates ODETTE is assigned an Organisation Identifier. Other values may be used for non-ODETTE codes.

Field	Description
SIOICD International Code Designator String (4)	A code forming part of the Organization Identifier.
SIOORG Organisation Code String(14)	A code forming part of the Organization Identifier. This field may contain the letters A to Z, the digits 0 to 9 and space and hyphen characters.
SIOCSA Computer Subaddress String(6)	A locally assigned address that uniquely identifies a system within an organization (defined by an Organization Identifier).

Where:

X(n) - An alphanumeric field of length n octets.

9(n) - A numeric field of length n octets.

Axway Gateway: Protocols

Submitting an OFTP Transfer Request

[Overview](#)

[Transfer Request parameters for OFTP](#)

[Submitting a Transfer Request without an Application](#)

[Submitting a Transfer Request with an Application](#)

Overview

You can submit Transfer Requests using:

- The GUI
- The **peltrans** online command
- User programs using APIs

The examples provided in this topic use the [peltrans](#) command.

Transfer Request parameters for OFTP

List of Transfer Request parameters

The following table describes parameters relevant to OFTP. For full details of Transfer Request parameters and their default values, refer to [Transfers: Parameters List](#).

Parameter name (and abbreviation)	Default	Comments
-org_alias (-oa)		Initiator site ID (a Local Site)
-dest_alias (-da)		Responder site ID (a Remote Site)
-file_component (-fc)		File to send
-file_name (-fn)		Symbolic name for the sent file
-appli (-ap)		Defines the file attributes for the transfer
-originator (-org)		File originator OFTP ID
-destination (-dest)		File destination OFTP ID
-mode (-m)	I	Transfer mode: I nitiator or R esponder
-direction (-dir)	O	Transfer direction: O = out (Sender), I = in (Receiver)
-compress (-comp)	U	Horizontal compression: values H, V and B have the same action
-priority (-prio)	0	Transfer priority
-rec_fmt (-rf)	F	Record format: F ixed, V ariable or S tream
-rec_len (-rl)	0	Record length: 0 – 32767
-block_size (-bs)	0	File block size (used by some operating systems)
-data_code (-dc)	U	Data code of local file
-x_rec_fmt (-xrf)	S	Transfer record format
-x_rec_len (-xrl)	512	Transfer record length
-x_data_code (-xdc)	B	Transfer data code
-padding (-pad)	0	Used for padding transfer records
-type	TRANS	If specified: TRANS or EERP
-purge_option (-po)	N	Purge file option for purge during transfer deletion
-user_processed (-usr_proc)	N	User processed flag

The following parameters concern file security (OFTP R2.0 only):

Parameter names (and abbreviation)	Default	Comments
-sfidsign	N	Is signed EERP required? { Y N }
-sfiddesc		Virtual file description (free text)
-trade_format (-tredef)		Must be CMS (Cryptographic Message Syntax)
-trade_org (-tradeo)		Original Trading Partner (sender)
-trade_dest (-traded)		Destination Trading Partner (receiver)
-trade_sign (-trades)		Is file signature required? { Y N }
-trade_sign_algo (-tradesa)		Signing algorithm: SHA1withRSA
-trade_encrypt (-tradee)		Is file encryption required? { Y N }
-trade_encryption_algo (-tradeea)		Encryption algorithm: { AES256 3DES }
-trade_compress (-tradecomp)		Is file compression required? { Y N }

Minimal set of Transfer Request parameters

The minimal set of parameters required for an OFTP Transfer Request is:

```
peltrans
```

```

-org_alias      <request originator (sender)>
-dest_alias     <request destination (receiver)>
-file_component <file to be sent>
-appli         <application name>
```

In this case, the file originator and destination coincide with the Site in communication.

If the file originator or final destination is different from the Site in communication, then you must specify them in the `originator` and `destination` parameters.

```
peltrans
```

```

-org_alias      <request originator>
-dest_alias     <request destination>
```

```
-file_component <file to be sent>

-appli          <application name>

-originator     <file originator (sender)>

-destination    <file destination (receiver)>
```

Transfer Request parameters when using file security services

If you use any of the file security services, the originator and destination must be predefined Trading Partners, with the appropriate security parameters.

peltrans

```
-org_alias      <request originator (sender)>

-dest_alias     <request destination (receiver)>

-file_component <file to be sent>

-appli          <application name>

-trade_org      <file originator>

-trade_dest     <file destination>

-trade_format   <CMS>

-trade_sign     < Y | N >

-trade_encrypt  < Y | N >

-trade_compression < Y | N >

-trade_encryption_algo < AES256 | 3DES >

-signed_eerp    < Y | N >

-vfile_desc     <optional virtual file description>
```

Notes:

trade_org, trade_dest and trade_format are mandatory.

Cryptographic parameters (signature, encryption, compression) are optional. If not specified, their values are those defined in the corresponding Trading Partner.

Submitting a Transfer Request without an Application

Gateway can process Transfer Requests submitted **without** an Application. All file system attributes that are normally retrieved from Application objects must be supplied in the Transfer Request.

For this type of transfer, Gateway uses a default Application named PEL1.

Note: You can also supply file system attributes for Transfer Requests submitted *with* an Application. In this case, Transfer Request parameters override the corresponding Application attributes.

Example

The following command submits a Transfer Request to send a file that contains 80-byte fixed-length records:

```
peltrans [-org_alias loc_odette]
          -dest_alias X25_odette
          -file_component <name of file to send>
          -f_org S
          -rec_fmt F
          -rec_len 80
```

Submitting a Transfer Request with an Application

Sites and Applications always belong to a group. For a Transfer to be valid, the Remote Site and Application must belong to the same group.

Sending a file: Initiator mode

A Site represents the Initiator/Sender, while the partner is the Responder/Receiver. In Initiator mode, the partner automatically receives the file, without submitting any other request.

Example 1: Sending a file on a TCP/IP network

The following **peltrans** command transmits a file with variable-length records (maximum length: **255**).

The command requests ASCII to EBCDIC character set conversion.

In this example, the Application used is var255, and the Remote Site is TCP_odette (as defined in [Configuring Gateway for OFTP](#)). The Application var255 is specified in the `file_name` attribute.

```
peltrans [-org_alias loc_odette]
          -dest_alias TCP_odette
          -appli var255
          -file_name var255
          -file_component <name of file to send>
```

Sending a file : Responder mode

The Responder can submit a transfer in Responder/Sender mode. This operation registers the request in the Gateway Mailbox.

The Remote Site partner initiates the file transmission by submitting a Transfer Request in Initiator/Receiver mode.

Example

In the following example, the Site `loc_odette` submits a Transfer Request in Responder/Sender mode, using an Application named `fix255`.

```
peltrans [-org_alias loc_odette]
         -dest_alias X25_odette
         -appli fix255
         -mode R
         [-direction O]
         -file_component <name of file to send>
```

Receiving a file: Initiator mode

A Site can submit a Transfer Request to receive a file from the partner Site. The Site receiving the file is the Initiator/Receiver, while the partner Site is the Responder.

As described in the *Responder mode* section below, the Initiator receives the file if the Remote Site previously submitted a Transfer Request in Responder/Sender mode.

OFTP cannot select which file to receive. Therefore, you must connect to the server so that the protocol can process corresponding existing transfers as illustrated in the following example:

Example

```
peltrans [-org_alias loc_odette]
         -dest_alias X25_odette
         [-mode I]
         -dir I
         [-rc_max 2]
```

Receiving a file: Responder mode

The Responder (Server) Site is the file receiver. This mode is used when the remote Initiator Site submits a Transfer Request to send a file. The Receiver automatically receives the file, without submitting any further request.

Sending an end-to-end response (EERP/NERP)

To send an OFTP end-to-end-response (EERP/NERP) acknowledgment, you must specify the following parameters:

Sending an EERP: Transfer Request parameters		
Parameter names	Default	Comments
-type	EERP	Identify this as an acknowledgment request
-reply_to_xfer		ID of the transfer to acknowledge
-nack	N	Is it a negative ACK? { Y N }

Axway Gateway: Protocols

Monitoring an OFTP transfer

[Protocol error reporting](#)

[Protocol trace messages](#)

Protocol error reporting

Gateway returns an error when a problem is encountered during Transfer Request processing. The error format depends on the Gateway user interface you use:

- For online commands and the GUI, the error message is displayed directly on output.
- For API calls, the return value is **-1** and an error code is automatically loaded into a global variable named **ErrCod**. You can use this code to retrieve the error message with the [GikErrMsg](#) API function.

Protocol errors are reported in end-of-transfer exits and in the log file.

Errors associated with ESID response messages

Code	Meaning	Action *
00	Normal session termination	None
01	Command not recognized	Retry, backup, stop
02	Protocol violation	Retry, backup, stop
03	User code not known	Stop
04	Invalid password	Stop
05	Local Site emergency shutdown	Retry, backup, stop

Code	Meaning	Action *
06	Command contained invalid data	Retry, backup, stop
07	NSDU size error	Retry, backup, stop
08	Resource not available	Retry, backup, stop
09	Timeout	Retry, backup, stop
10	Incompatible mode or capabilities	Stop
11	Invalid challenge response	Retry, backup, stop
12	Secure requirement incompatible	Retry, backup, stop
99	Unspecified abort code	Retry, backup, stop

* Actions

The possible Gateway actions in response to the ESID codes in the above table are as follows:

- **None** - No action necessary as the session ended normally
- **Stop** - Stop Site by disabling Site Responder mode
- **Retry, backup, stop** - Try to re-establish session, possibly with backup communication path, and finally stop Site by disabling Site Responder mode

Errors associated with SFNA and EFNA response messages

Code	Meaning	Action *
00	Normal transfer termination	None
01	Invalid file name	Cancel
02	Invalid destination	Cancel
03	Invalid origin	Cancel
04	Storage record format not supported	Cancel
05	Maximum record length not supported	Cancel
06	File size too big	Cancel
10	Invalid record count	Restart from the beginning
11	Invalid byte count	Restart from the beginning
12	Access method failure	Restart
13	Duplicate file	Cancel
14	File direction refused	Cancel
15	Cipher suite not supported	Cancel
16	Encrypted file not allowed	Cancel
17	Unencrypted file not allowed	Cancel

Code	Meaning	Action *
18	Compression not allowed	Cancel
19	Signed file not allowed	Cancel
20	Unsigned file not allowed	Cancel
21	Invalid file signature	Cancel
22	File decryption failure	Cancel
23	File decompression failure	Cancel
99	Unspecified reason	Restart

* Actions

The possible Gateway actions in response to the SFNA and EFNA codes in the above table are as follows:

- **None** - No action necessary as the transfer ended normally
- **Cancel** - Set transfer state to Canceled: no transfer restart
- **Restart** - Try to resume from the last transfer position
- **Restart from the beginning** - Try to resume transfer from beginning of file

Protocol trace messages

Each protocol command or response corresponds to a protocol trace message that you can record in the log file. The protocol trace is an optional feature that you define for transfer activity for a given Site.

You can enable or disable the protocol trace for a Site, either dynamically during Gateway processing, or when you define or modify the attributes of a Site.

When a protocol trace is enabled or disabled dynamically, it is validated only in memory (the modification is lost when Gateway is stopped).

Activating protocol trace

1. In the Gateway GUI, select the Remote Site for which you want to activate the trace.
2. Right-click the Remote Site to display a context menu.
3. Select **Trace mode**.

Alternatively, enter an online command such as:

```
pelctl start site -alias remote_oftp -trace_mode
```

Note: The FPDUs that the protocol uses to send a diagnostic command (ESID, SFNA and EFNA) are always written to the log file with the previous FPDU, which may be the origin of the error.

[OFTP trace example](#)

[OFTP protocol log messages \(ODT\)](#)

[About OFTP](#)

[Configuring Gateway for OFTP](#)

[Submitting an OFTP Transfer Request](#)

Axway Gateway: Protocols

OFTP trace example

This topic shows an example of a trace for a typical OFTP R2.0 protocol exchange. The trace is displayed in the Gateway Log file.

Activate the protocol trace using the GUI (right-click the Remote Site and select **Trace mode**) or by entering a command such as:

```
pelctl start site -alias remote_oftp -trace_mode
```

```
TRADE123I 11:21:04 [89] CMS File encoded from 'local odette
ident' to 'remote odette ident'.
NET101I 11:21:04 (98) outgoing connection request [src_
add="/0",dest_add="mycomputer/61260"].
NET104I 11:21:04 (98) outgoing connection confirmation [resp_
add="127.0.0.1/2213"].
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 )
req:Read len:19 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:04 Type: CONX - FPDU: SSRM
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSRMSG Ready Message ..... :
ODETTE FTP READY <- R_R_OFTP_A 97.
ODT101I 11:21:04 SSRMCR ..... :
. <- R_R_OFTP_A 97.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 )
req:Write len:61 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:04 Type: CONX - FPDU: SSID
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDLEV Protocol Release Level ..... : 5
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCODE Initiator's Identification Code :
OFTP_SITE_B <- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDPSWD Initiator's Password ..... :
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDSDEB Data Exchange Buffer Size ..... :
04000 <- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDSR Send / Receive Capabilities .... : B
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCMPR Buffer Compression Indicator ... : N
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDREST Restart Indicator ..... : Y
```

```

<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDSPEC Special Logic Indicator ..... : N
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCRED Credit ..... :
004 <- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDAUTH Secure Authentication ..... : Y
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDRSV1 Reserved ..... :
<- S_R_OFTP_A 97.
ODT101I 11:21:04 SSIDUSER User Data ..... :
<- S_R_OFTP_A 97.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 ) req:Read
len:61 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:04 Type: CONX - FPDU: SSID
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDLEV Protocol Release Level ..... : 5
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCODE Initiator's Identification Code : <-
R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDPSWD Initiator's Password ..... :
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDSDEB Data Exchange Buffer Size ..... :
04000 <- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDSR Send / Receive Capabilities .... : B
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCMPR Buffer Compression Indicator ... : N
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDREST Restart Indicator ..... : Y
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDSPEC Special Logic Indicator ..... : N
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCRED Credit ..... :
004 <- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDAUTH Secure Authentication ..... : Y
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDRSV1 Reserved ..... :
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDUSER User Data ..... :
<- R_R_OFTP_A 97.
ODT101I 11:21:04 SSIDCR ..... :
. <- R_R_OFTP_A 97.
ODT200I 11:21:04 R_OFTP_A(97) connection established with L_
OFTP_B.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 )
req:Write len:1 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:04 Type: AUTH - FPDU: SECD
<- S_R_OFTP_A 97.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 ) req:Read
len:192 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:04 Type: AUTH - FPDU: AUCH
<- R_R_OFTP_A 97.
ODT101I 11:21:04 AUCHCHIL Challenge Length ..... :
0x00BD <- R_R_OFTP_A 97.

```

```

ODT101I 11:21:04 AUCHCHAL Challenge ..... :
0...*.H.....0... <- R_R_OFTP_A 97.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 )
req:Write len:21 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:04 Type: AUTH - FPDU: AURP
<- S_R_OFTP_A 97.
ODT101I 11:21:04 AURPRSP Challenge Response ..... :
D...z....!|qJ.T....Ad <- S_R_OFTP_A 97.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 ) req:Read
len:1 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:04 Type: AUTH - FPDU: SECD
<- R_R_OFTP_A 97.
ODT101I 11:21:04 Prot trace for: R_OFTP_A (97 )
req:Write len:330 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:04 Type: AUTH - FPDU: AUCH
<- S_R_OFTP_A 97.
ODT101I 11:21:04 AUCHCHIL Challenge Length ..... :
0x0147 <- S_R_OFTP_A 97.
ODT101I 11:21:04 AUCHCHAL Challenge ..... :
0...*.H.....0... <- S_R_OFTP_A 97.
ODT101I 11:21:05 Prot trace for: R_OFTP_A (97 ) req:Read
len:21 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:05 Type: AUTH - FPDU: AURP
<- R_R_OFTP_A 97.
ODT101I 11:21:05 AURPRSP Challenge Response ..... :
_...CO~."...q.-mr.Z.. <- R_R_OFTP_A 97.
ODT101I 11:21:05 Prot trace for: R_OFTP_A (97 )
req:Write len:199 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:05 Type: PROT - FPDU: SFID
<- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDDSN Virtual File Dataset Name ..... :
STREAM <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDRSV1 Reserved ..... :
<- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDDATE Virtual File Date stamp ..... :
20080424 <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDTIME Virtual File Time stamp ..... :
1121040000 <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDUSER User Data ..... :
<- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDDEST Destination ..... :
remote odette ident <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDORIG Originator ..... :
local odette ident <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDFMT File Format ..... : U
<- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDLRECL Maximum Record Size ..... :
00000 <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDFSIZ File Size, (1K blocks) ..... :
0000000000001 <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDOSIZ Original File Size, (1K blocks) :
0000000000001 <- S_R_OFTP_A 97.
ODT101I 11:21:05 SFIDREST Restart Position ..... :

```

```

00000000000000000000    <- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDSEC Security Level ..... : 03
<- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDCIPH Cipher suite selection ..... : 02
<- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDCOMP File compression algorithm ..... : 1
<- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDENV File enveloping format ..... : 1
<- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDSIGN Signed EERP request ..... : Y
<- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDDESCL Virtual File Description length :
034                                <- S_R_OFTP_A 97.
ODT101I      11:21:05 SFIDDESC Virtual File Description ..... :
Odette v2 virtual file description <- S_R_OFTP_A 97.
ODT101I      11:21:05 Prot trace for: R_OFTP_A              (97 ) req:Read
len:18 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I      11:21:05 Type: PROT - FPDU: SFPA
<- R_R_OFTP_A 97.
ODT101I      11:21:05 SFPAACNT Answer Count ..... :
000000000000000000            <- R_R_OFTP_A 97.
ODT202I      11:21:05 R_OFTP_A(97) [89] beginning of transmission,
file name:STREAM, date:20080424, time:1121040000.
ODT203I      11:21:05 R_OFTP_A(97) [89] beginning of transmission
from L_OFTP_B, file: e:\working\v613\run_time\tmp\F0000089.ec.
ODT101I      11:21:05 Prot trace for: R_OFTP_A              (97 )
req:Write len:4000   Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I      11:21:05 Type: DATA - FPDU: DATA
<- S_R_OFTP_A 97.
ODT101I      11:21:05 Prot trace for: R_OFTP_A              (97 )
req:Write len:4000   Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I      11:21:05 Type: DATA - FPDU: DATA
<- S_R_OFTP_A 97.
ODT101I      11:21:05 Prot trace for: R_OFTP_A              (97 )
req:Write len:686     Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I      11:21:05 Type: DATA - FPDU: DATA
<- S_R_OFTP_A 97.
ODT101I      11:21:05 Prot trace for: R_OFTP_A              (97 )
req:Write len:35       Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I      11:21:05 Type: PROT - FPDU: EFID
<- S_R_OFTP_A 97.
ODT101I      11:21:05 EFIDRCNT Record Count ..... :
000000000000000000            <- S_R_OFTP_A 97.
ODT101I      11:21:05 EFIDUCNT Unit Count ..... :
00000000000008546             <- S_R_OFTP_A 97.
ODT101I      11:21:05 Prot trace for: R_OFTP_A              (97 ) req:Read
len:2 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I      11:21:05 Type: PROT - FPDU: EFPA
<- R_R_OFTP_A 97.
ODT101I      11:21:05 EFFACD Change Direction Indicator ..... : N
<- R_R_OFTP_A 97.
ODT204I      11:21:05 R_OFTP_A(97) [89] end of transmission, file
name:STREAM, date:20080424, time:1121040000.

```

```

ODT205I 11:21:05 R_OFTP_A(97) [89] end of transmission from L_
OFTP_B, file: e:\working\v613\run_time\tmp\F0000089.ec.
AMT524I 11:21:05 remote odette ident[89] XFBCConnInitParams().
AMT524I 11:21:05 remote odette ident[89] XFBCConnXferNotif(h_
Conn=00A52FC8, event=1[XFBCONN_NOTIF_SENT], h_Params=00A53FA8)..
ODT101I 11:21:05 Prot trace for: R_OFTP_A (97 )
req:Write len:1 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:05 Type: PROT - FPDU: CD
<- S_R_OFTP_A 97.
ODT101I 11:21:06 Prot trace for: R_OFTP_A (97 ) req:Read
len:1213 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:06 Type: PROT - FPDU: EERP
<- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPDSN Virtual File Dataset Name ..... :
STREAM <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPDRV1 Reserved ..... :
<- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPDATE Virtual File Date stamp ..... :
20080424 <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPTIME Virtual File Time stamp ..... :
1121040000 <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPUSER User Data ..... :
<- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPDEST Destination ..... :
local odette ident <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPORIG Originator ..... :
remote odette ident <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPHSHL Virtual File hash length ..... :
0x0014 <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPHSH Virtual File hash ..... :
...#+f...'Z..tJ..(o <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPISGL EERP signature length ..... :
0x043B <- R_R_OFTP_A 97.
ODT101I 11:21:06 EERPISGL EERP signature ..... :
0...*.H.....0....1.0...+..... <- R_R_OFTP_A 97.
AMT524I 11:21:06 remote odette ident[89] XFBCConnXferNotif()
rc=1 ..
AMT524I 11:21:06 remote odette ident[89] XFBCConnFreeParams(h_
Params=00A53FA8)..
ODT220I 11:21:06 R_OFTP_A(97) [90] receive file acknowledge,
file name:STREAM, date:20080424, time:1121040000.
ODT221I 11:21:06 R_OFTP_A(97) [90] receive file acknowledge for
L_OFTP_B.
ODT101I 11:21:06 Prot trace for: R_OFTP_A (97 )
req:Write len:1 Prot:ODETTE <- S_R_OFTP_A 97.
ODT101I 11:21:06 Type: PROT - FPDU: RTR
<- S_R_OFTP_A 97.
AMT524I 11:21:06 local odette ident[90] XFBCConnInitParams().
AMT524I 11:21:06 local odette ident[90] XFBCConnXferNotif(h_
Conn=00A52FC8, event=1[XFBCONN_NOTIF_SENT], h_Params=00A546C0)..
ODT101I 11:21:06 Prot trace for: R_OFTP_A (97 ) req:Read
len:1 Prot:ODETTE <- R_R_OFTP_A 97.
ODT101I 11:21:06 Type: PROT - FPDU: CD

```

```

<- R R_OFTP_A 97.
AMT524I    11:21:06 local odette ident[90] XFBCConnXferNotif() rc=1
..
AMT524I    11:21:06 local odette ident[90] XFBCConnFreeParams(h_
Params=00A546C0)..
ODT216I    11:21:26 R_OFTP_A(97) disconnect request send by L_
OFTP_B [reason="Normal session termination"].
ODT101I    11:21:26 Prot trace for: R_OFTP_A          (97 )
req:Write len:33      Prot:ODETTE <- S R_OFTP_A 97.
ODT101I    11:21:26 Type: CONX - FPDU: ESID
<- S R_OFTP_A 97.
ODT101I    11:21:26 ESIDREAS Reason Code ..... : 00
<- S R_OFTP_A 97.
ODT101I    11:21:26 ESIDREASL Reason Text Length ..... :
026                                     <- S R_OFTP_A 97.
ODT101I    11:21:26      ESIDREAST Reason Text .....
: Normal session termination          <- S R_OFTP_A 97.
NET106I    11:21:26 (98) disconnection indication
[reason="No error (0x0)"] [origin="0"].

```

[OFTP protocol log messages \(ODT\)](#)

[About OFTP](#)

[Submitting an OFTP Transfer Request](#)

[Monitoring an OFTP transfer](#)

Axway Gateway: Protocols

Using PassPort PM with OFTP

[Prerequisites](#)

[Configuring PassPort PM for OFTP](#)

[Configuring Gateway for OFTP](#)

[Defining Sites for OFTP](#)

[Submitting OFTP Transfer Requests](#)

This topic describes how to use Axway PassPort PM in conjunction with OFTP in Gateway.

Prerequisites

- Make sure that the `JAVA_HOME` environment variable points to the correct JVM home directory and the following directories are in your `PATH` environment variable:
 - `<java_home>/bin`
 - `<java_home>/jre/bin/server/`
 - `<passport_pm_home>/Passport_PM/bin`
- Gateway must be installed.

Configuring PassPort PM for OFTP

Refer to the PassPort PM documentation for information about how to use the PassPort PM portal to define a Trading Partner and its associated communication protocol attributes.

The following sections only describe the protocol attributes used in an OFTP Trading Partner.

Defining a PassPort PM OFTP Trading Partner

An OFTP Trading Partner contains the following attributes:

Identity Attributes

Attribute	Description	Value
ProtoIdent (*)	Key field uniquely defines an OFTP protocol site	Corresponds to SSIDCODE in SSIDCMD.
Version (*)	Identifies the protocol version to use	Accepted values: R1.2, R1.3, R1.4, R2.0
PartnerId (*)	Key field uniquely identifies a Trading Partner	Corresponds to the SFIDORIG and SFIDDEST in SFIDCMD.

(*) indicates a mandatory attribute.

Transport Level Protocol Attributes

Attribute	Description	Value
NetworkAddress (*)	The network address where the site is located.	Format is TCP:host:port

Attribute	Description	Value
ClientPassword	Password	Corresponds to the SSIDPSWD in SSIDCMD when this Site is in initiator (client) mode. The OFTP initiator uses it to identify itself to the responder.
ServerPassword	Password	Corresponds to the SSIDPSWD in SSIDCMD when this Site is in responder (server) mode. The OFTP responder uses it to identify itself to the initiator.
ConnectionMessage	OFTP Connection Message	Corresponds to the SSIDUSER in SSIDCMD.
EERPStyle	EERP style	Accepted values: • International (default) • French
AckOption	acknowledgment option	Accepted values: • NONE: No EERP action is defined • BY_MONITOR: Automatic EERP generated by Gateway • BY_USER: EERP generated by user application • NOT_TO_ACK: EERP not generated

Attribute	Description	Value
ApplicationMethod	Defines the method used to deduce the Application object used by Gateway.	Accepted values: • IN_FILE_NAME: Application object name is in SFIDDSN • IN_USER_MSG: Application object name is in SFIDUSER • IN_USER_EXIT: Application object name is given by user exit
SSLTLS	Transport level encryption	Accepted values: • NONE: No transport level encryption is used • SSL: Transport level encryption is SSL • TLS: Transport level encryption is TLS
CertificateType	Ignored if SSLTLS option is NONE, otherwise it is mandatory.	Accepted value: certificate
CertificateId	Ignored if SSLTLS option is NONE, otherwise it is mandatory. The identifier used to retrieve the X.509 certificate from the DB. This certificate is used to establish the SSL/TLS session.	
CertificatePassword	Ignored if SSLTLS option is NONE. Only required if the certificate has a private key.	

Attribute	Description	Value
AuthenticationMethod	Defines the secured authentication method to use.	Accepted values: • NONE • ENVELOPPEDDATA • ENCRYPTEDDATA
AuthCertificateType	Ignored if AuthenticationMethod = NONE, otherwise it is mandatory.	Accepted value: certificate
AuthCertificateId	Ignored if AuthenticationMethod = NONE, otherwise it is mandatory. The identifier used to retrieve the X.509 certificate from the DB. This certificate is used to establish the OFTP R2.0 secured authentication session.	
AuthCertificatePassword	Ignored if AuthenticationMethod = NONE. Required only if the certificate has a private key	

(*) indicates a mandatory attribute. All other attributes are optional.

EDI Level Protocol Attributes

Attribute	Description	Value
SignCertificateType	X.509 certificate	Accepted value: certificate
SignCertificateId	The identifier used to retrieve the X.509 certificate from the DB. This certificate is used to sign or verify a CMS SignedData.	

Attribute	Description	Value
SignCertificatePassword	Only required if the certificate has a private key. In other words, used to sign a CMS SignedData.	
CiphCertificateType	X.509 certificate	Accepted value: certificate
CiphCertificateId	The identifier used to retrieve the X.509 certificate from the DB. This certificate is used to generate or verify a CMS EnvelopedData.	
CiphCertificatePassword	Only required if the certificate has a private key. In other words, used to verify a CMS EnvelopedData.	
KeyEncryptionAlgorithm	Signing algorithm	Accepted value: RSA
DataEncryptionAlgorithm	Encryption algorithm	Accepted values: • AES256 • 3DES
CompressionAlgorithm	Compression algorithm	Accepted value: ZLIB
SignatureRequired	Is signature required?	Accepted values: YES, NO If YES, the received file must be signed, otherwise it is rejected.
EncryptionRequired	Is encryption required?	Accepted values: YES, NO If YES, the received file must be encrypted, otherwise it is rejected.

(*) indicates a mandatory attribute. All other attributes are optional.

Usage Notes

As described above, ProtoIdent corresponds to OFTP SSIDCODE while PartnerId corresponds to OFTP SFIDORIG / SFIDDEST. The values of these two fields can be identical or different, depending on your configuration.

There are two possible usage cases:

Case 1: There is a one-to-one mapping between ProtoIdent and PartnerId

There is only one SFIDORIG / SFIDDEST per SSIDCODE. In this case, you need only define one PassPort PM communication profile per Trading Partner.

Case 2: There is a one-to-many mapping between ProtoIdent and PartnerId

There is more than one SFIDORIG / SFIDDEST per SSIDCODE. In this case, you need to define:

- One PassPort PM communication profile for ProtoIdent with only transport level attributes. The ProtoIdent value must be unique among all communication profiles defined for OFTP. You must also set the PartnerId field to the same value as the ProtoIdent field.
- For each PartnerId associated with this ProtoIdent, one PassPort PM communication profile with only EDI attributes. The PartnerId value must be unique among all communication profiles defined for OFTP. You must also set the ProtoIdent field to the same value as the PartnerId field.

Configuring Gateway for OFTP

In the Gateway configuration menu:

1. [Activate the OFTP protocol.](#)
Specify the Gateway listening port and activate the protocol.
2. [Activate the PassPort PS connector.](#)
Specify the PassPort PS server IP address and TCP port.
3. [Activate the PassPort PM connector.](#)
Specify the PassPort PM home directory and parameters file.

Defining Sites for OFTP

Local Site

Define at least one OFTP Local Site, with an authentication certificate.

Whether or not secured authentication will be carried out depends on the communication profile defined for the remote Trading Partner. If the remote Trading Partner is an OFTP R2.0 partner with an authentication certificate, then the OFTP secured session authentication will be activated.

OFTP template Site

In the OFTP template Site TODT, the local partner alias must point to the Local Site defined above.

Submitting OFTP Transfer Requests

To submit an OFTP Transfer Request using a PassPort PM definition, you must specify the corresponding PassPort PM communication protocol profiles (DEA) used.

Depending on the PassPort PM definition, there are two possible cases:

Case 1: There is a one-to-one mapping between Protoident and PartnerId (most common case)

```
peltrans
    -org_alias      local_site_alias
    -file_component  file_to_send
    -sfidsign        yes | no
    -sfiddesc        "virtual file description"
    -tredef          CMS
    -tpmlocdea        tpm_local_partner
    -tpmdea          tpm_remote_partner
    -trades          yes | no
    -tradee          yes | no
    -tradecomp       yes | no
    -tradea          AES256 | 3DES
```

tpmlocdea and tpmdea must both contain transport level attributes and EDI level attributes.

Case 2: There is a one-to-many mapping between Protoident and PartnerId

```
peltrans
    -org_alias      local_site_alias
    -file_component  file_to_send
    -sfidsign        yes | no
    -sfiddesc        "virtual file description"
    -tredef          CMS
    -tpmlocdea        tpm_local_partner
    -tpmdea          tpm_remote_partner
    -tpmdea2         tpm_remote_partner2
    -trades          yes | no
    -tradee          yes | no
    -tradecomp       yes | no
    -tradea          AES256 | 3DES
```

tpmlocdea must contain transport level attributes and EDI level attributes.

tpmdea must contain EDI attributes for the remote partner.

tpmdea2 must contain transport level attributes for the remote partner.

Refer to [Usage Notes](#) for a detailed explanation.

For full details of Transfer Request parameters and their default values, refer to [Transfers: Parameters List](#).

[About OFTP](#)

[Configuring Gateway for OFTP](#)

[Monitoring an OFTP transfer](#)

[Submitting an OFTP Transfer Request](#)

Transfer to an existing file name

The following properties define the actions Gateway performs on a pre-existing file before receiving the transfer of a file with the same name, in the case of OFTP and PeSIT protocols.

- **File availability**
Specifies whether the Gateway server accepts incoming files with the same name as existing ones.
- **File action**
Specifies the action Gateway takes if an incoming file has the same name as an existing one.

File availability	File action	File with same name as the incoming file	Transfer accepted
New	Erase	exists	No
	Delete	exists	No
	Verify	exists	No
New	Erase	does not exist	Yes
	Delete	does not exist	Yes
	Verify	does not exist	Yes
Old	Erase	exists	Yes
	Delete	exists	Yes

File availability	File action	File with same name as the incoming file	Transfer accepted
	Verify	exists	Yes, if the existing file is empty No, if the existing file is not empty
	Erase	does not exist	No
	Delete	does not exist	No
Both	Verify	does not exist	No
	Erase	exists	Yes
	Delete	exists	Yes
	Verify	exists	Yes, if the existing file is empty No, if the existing file is not empty
	Erase	does not exist	Yes
	Delete	does not exist	Yes
	Verify	does not exist	Yes

Receiving a pre-existing file

The values in this section define the action of the monitor if the file being transferred already exists.

File availability

Select one of the following values:

- **Undefined**
- **New**
- **Old**
- **Both**

Gateway responds to the various possible pre-existing file conditions depending on the values of the *File availability* and *File action* parameters.

Refer to the description of the *File action* parameter below.

File action

Select one of the following values:

- **Undefined**
- **Erase**
- **Delete**
- **Verify**

To determine how to handle a pre-existing file, Gateway interprets the File availability and File action parameter values as follows:

If *File availability* is set to:

- **NEW** and a valid file with the same file name already exists, then:
 - **ERASE** = refuse
 - **DELETE** = refuse
 - **VERIFY** = refuse
- **NEW** and a file with the same file name does not already exist, then:
 - **ERASE** = accept
 - **DELETE** = accept
 - **VERIFY** = accept
- **OLD** and a valid file with the same file name already exists, then:
 - **ERASE** = accept
 - **DELETE** = accept
 - **VERIFY** = accept if empty, refuse if not empty
- **OLD** and a file with the same file name does not already exist, then:
 - **ERASE** = refuse
 - **DELETE** = refuse
 - **VERIFY** = refuse
- **BOTH** and a valid file with the same file name already exists, then:
containing
 - **ERASE** = accept
 - **DELETE** = accept
 - **VERIFY** = accept if empty, refuse if not empty
- **BOTH** and a file with the same file name does not already exist, then:
containing
 - **ERASE** = accept
 - **DELETE** = accept
 - **VERIFY** = accept

where:

accept = accept the transferred file
refuse = refuse the transferred file

Axway Gateway: Protocols

About PEL

PEL

Introduction to PEL

PEL is an end-to-end protocol designed to transfer files between two platforms, irrespective of the network support, hardware, and software architecture of the host computers.

Gateway uses PEL with different communication types: SNA in LU 6.2 mode, TCP/IP, and some proprietary communication stacks such as DSA and SPX/IPX. You define these communication attributes in Remote Site objects.

PEL services

The general PEL characteristics are:

- File transmission
- File reception
- Restart of an interrupted transfer
- Data compression
- Two-way transfers over the same logical connection
- Message transfer

File transmission

This service enables you to transfer data to a remote partner. A logical link must have been previously established between the two partner Sites.

File reception

This service enables you to transfer file data belonging to the Remote Site.

Restart of an interrupted transfer

PEL can automatically restart transfers interrupted by technical incidents or by a user.

The interruption closes the file and disables the protocol and network connections. It is always the Initiator who launches the restart, and the data receiver who decides from which synchronization point the transfer is to be restarted.

Data compression

Data can be compressed in order to reduce the size of data actually transferred.

Three types of compression are available:

- Horizontal compression: deletes repeated characters in a logical record
- Vertical compression: transmits only data which differs from the previous record
- Horizontal and vertical compression: combines both of the above methods

Two-way transfers over the same logical connection

PEL allows only one file transfer in a given session at a time. Nevertheless, Gateway can perform several concurrent transfers either as:

- *consecutive* transfers within the same session
- several *simultaneous* file transfers on *different* sessions, and in both directions.

Gateway PEL file transfer (F.T.) enables you to use the same Site for simultaneous file transfers in both directions.

Message transfer

PEL protocol can carry user messages of up to 80 bytes. They are transmitted, unmodified, by the sender to the partner Site, and can be used in end-of-transfer exit functions to automate specific processing.

You supply these user messages using the **Message to send** (`snd_msg`) parameter in the Transfer Request.

Site and transfer identification

PEL uses the following information to uniquely identify a transfer:

Parameter	Description
Site name	Gateway assigns each Site in a network a unique: • Protocol name • Local Site defined in the configuration file by the <code>local_site_proto_ident</code> parameter

Parameter	Description
Application	PEL associates an Application with each transfer. This Application is defined in Gateway, and references parameters such as the compression mode and file formatting attributes.
PEL-specific Application, PEL1	Manages Transfer Requests submitted without an explicit Application. Gateway retrieves the file and transfer attributes from the Transfer Request.
Rank number	The Sender Site transmitting the file assigns this to each transfer.

Transfer modes

The Initiator Site initiates the physical connection and the Responder Site responds to it. PEL allows data transfers in both directions: the Site sending data is the Sender, and the one receiving data is the Receiver. A given Site object can have one of the following roles:

- Initiator/Sender
- Initiator/Receiver
- Responder/Sender
- Responder/Receiver

Protocol commands and response messages are exchanged between the two partner Sites at some specific steps during file transfer processing, as described in [How PEL processing works](#).

File and transfer attributes

The file content is transmitted over the network in data blocks whose size you define.

An acknowledgment mechanism is used by PEL to regulate data flow. It is configured via the `-check_window` attribute of the Remote Site object.

Protocol data is coded in EBCDIC. Code conversion allows automatic conversion of file data transmitted to the partner Site. It is configured via `-data_code` attributes defined in Application objects.

Routing

PEL supplies a Store-and-Forward mechanism, allowing a transfer from one end Site to another via intermediate Sites. Gateway can use Logical Sites to submit file transfer towards a non-adjacent partner.

File transfer phases

The transfer actually begins with the physical connection of two Sites. PEL processes a file transfer in the following sequence:

Phase	Description
Protocol connection	Initiated by server Site after the physical connection and a possible logon phase. User identification and password can be checked at this stage for security purposes.
Selecting the set of information to be transferred	Initiated by Initiator Site. If files are to be transferred from the Responder Site to the Initiator Site, this selection is made from the set of information previously received from the Responder.
Data transfer	After the selection, file data is sent without a header, and with flow control regulation, if requested.
Deselecting the set	When all file data is sent, a command containing checking information indicates the end of data transmission to the Receiver. The Receiver Site verifies the validity of the received information, and then accepts or rejects the set.
Protocol connection release	Initiated by Initiator Site (after de-selecting the set) and acknowledged by the Responder.

How PEL processing works

Protocol commands and responses

Protocol messages comprise commands and responses. Each message is prefixed with its identification string followed by items defined in the PEL protocol.

Commands are prefixed with "?" and responses with "*" as illustrated in the following table:

Type	Name	Description
Command	?DEBUT	Beginning of protocol connection
	?FIN	End of protocol connection
	?LOTS	Request for the list of selectable sets
	?TRANS	Request for a file transfer

Type	Name	Description
Response	*ACCEPTE	Acceptance of the connection
	*ACQ	Data acknowledgment
	*ADL	Acceptance of the set
	*DDL	Beginning of the set
	*FDL	End of the set
	*FIN	End of connection
	*LL	List of sets
	*NDL	Refusal of a set
	*NON	Negative response
	*OK	Positive response
	*RDL	Restart of a set
	*REFUSE	Rejection of a connection

Interaction diagrams

The following interaction diagrams illustrate the communication flows between Initiator and Responder.

File transmission

Initiator	Responder
	<----- ?DEBUT
*ACCEPTE Initiator	----->
	<----- *OK
?TRANS set	----->
	<----- *OK
*DDL	----->

Initiator		Responder	
		<-----	*OK or *RDL
data	----->		
&..			
data	----->		
		<-----	(*ACQ)
data	----->		
&..			
data	----->		
		<-----	(*ACQ)
*FDL	----->		
		<-----	*ADL
?FIN	----->		
		<-----	*END OF REQUEST

File reception

Initiator		Responder	
		<-----	?DEBUT
*ACCEPTE Initiator	----->		
		<-----	*OK
?LOTS	----->		
		<-----	*LL list
?TRANS set	----->		
		<-----	*DDL
*OK or *RDL	----->		

Initiator		Responder	
		<-----	data &..
		<-----	data
(*ACQ)	----->		
		<-----	data &..
		<-----	data
(*ACQ)	----->		
		<-----	data &
		<-----	*FDL
*ADL	----->		
		<-----	*OK
?FIN	----->		
		<-----	*END OF REQUEST

Refer to [About user exits](#) for details of exit routines supplied for the PEL protocol.

Configuring Gateway for PEL

- [About PEL configuration](#)
- [Creating Site objects](#)
- [Creating an Application](#)

About PEL configuration

Before submitting a Transfer Request, you must create the following Gateway objects if they do not already exist:

- **Local Site**

On installation, you must create a Local Site corresponding to the `local_site_alias` and `local_site_proto_ident` parameters in the configuration file `<Gateway installation directory>/run_time/etc/conffile`. This is the only Local Site that you can use with the PEL protocol.

- **Remote Site**

Define as many Remote Sites as needed.

- **Application**

Define at least one Application.

You can use either the GUI or the `peladm` online command to create these objects.

Creating Site objects

The following examples describe how to create Gateway Site objects using the `peladm` online command. If you do not supply attributes explicitly, Gateway uses either default values from the operator interface or from Gateway internal default attributes.

Certain platforms may require non-default values for some optional parameters. Refer to platform-specific documentation for more information.

Example 1: Local Site locpel

```
peladm create_loc_site -alias locpel  
  
-comments "Local Site"
```

If you do not specify a group for the Remote Site that you create, the Site uses the default group `GDEFAULT`.

Example 3: Remote Site using TCP/IP

The following command creates a Remote Site with the following attributes:

- IP address: 193.56.234.20
- port number: 22222
- block size: 4000
- number of data blocks between two acknowledgments: 5
- maximum number of simultaneous server connections: 10

```
peladm create_site -alias tpelpc  
  
-protocol PEL  
  
-comments "Remote Site: PEL on TCP/IP"  
  
-comm_type TCP  
  
-dest_address 193.56.234.20/22222  
  
-data_size_max 4000
```



```
-check_window 5  
  
-resp_conn_max 10
```

Creating an Application

The following example creates an Application object using the `peladm` online command. If you do not supply attributes explicitly, Gateway uses either default values from the operator interface or from Gateway internal default attributes.

Note: For the PEL LOTS command specifically, the Application name is limited to four characters, instead of the standard eight.

Example

```
peladm create_appli -alias fix255  
  
-comments "pelapplication"
```

[About PEL](#)

[Submitting a PEL Transfer Request](#)

[Working with Local Sites](#)

[Working with Local Sites \(command line\)](#)

[Working with Remote Sites](#)

[Working with Remote Sites \(command line\)](#)

[Working with Applications](#)

[Working with Applications \(command line\)](#)

PEL request counters for LOTS transfers

The following table describes the PEL request counters used when a LOTS transfer is submitted:

counts_0	Number of LOTS transfers
counts_1	Number of KEYS received for the last LOTS transfer
counts_2	Number of XFERS created for the last LOTS transfer
counts_3	Number of FAILURES to create xfers for the last LOTS transfer

counts_4	Reserved for future use
counts_5	Number of KEYS received for the all LOTS transfer
counts_6	Number of XFERS created for the all LOTS transfer
counts_7	Number of FAILURES to create xfers for the all LOTS transfer
counts_8	Reserved for future use
counts_9	Reserved for future use [end of table]

Note These counters can be used as symbolic variables in Decision Rules and Transfer Models.

Axway Gateway: Protocols

Submitting a PEL Transfer Request

[About PEL Transfer Requests](#)

[File submission process](#)

[Submitting a Transfer Request without an Application](#)

[Submitting a Transfer Request with an Application](#)

[Reply Codes](#)

[Protocol trace messages](#)

About PEL Transfer Requests

Submit Transfer Requests using the GUI, the `peltrans` online command, or user programs linked with APIs. Each of these interfaces transmits transfer parameters to Gateway.

The examples provided in this section use the `peltrans` online command, which has the following syntax:

```
peltrans <argument> value <argument> value <argument> value
```

Transfer Request parameters			
Keyword names (*)	Appli default	Internal default	Comments
-originator			Responder Site name (used for type = LOTS transfers only)

Transfer Request parameters			
-mode (-m)	I	I	Transfer mode: Initiator or Responder
-direction (-dir)	O	O	Transfer direction: O = out (Sender), I = in (Receiver) Sending or receiving a file
-compress (-comp)	U	U	U = no compression
-prio	1	0	Transfer priority
-f_org	S		File organization
-rec_fmt (-rf)	F	S	Record format
-rec_len (-rl)	0	512	Record length
-data_code (-dc)		B	B = binary
-x_rec_fmt (-xrf)		S	Transfer record format
-x_rec_len (-xrl)	0	512	Transfer record length
-x_data_code (-xdc)		B	Transfer data code
-padding		0	
-type	TRANS	TRANS	Type of processing
-interval			Interval between Remote Site polling (LOTS and POLL request types)
-text_file	N	N	
-snd_msg (-sm)			Transmitted unchanged: can be a Gateway Application
-sd			System-dependent parameter
-retry_count_max (-rc_max)	10	5	Max retries before canceling transfer
-purge_option (-po)	N	N	Purge file option for purge during transfer deletion
-user_processed (-usr_proc)	N	N	User processed flag

(*) Abbreviated keyword names in parentheses.

File submission process

Description

Gateway records Transfer Requests in the Mailbox, provided that they contain the minimum set of consistent parameters required for Gateway processing (including transfer monitoring, scheduling parameters, and file system attributes).

Typically, the Transfer Request contains only a subset of parameters. To record Transfer Requests in the Mailbox, Gateway completes the remaining transfer parameters from:

- User interface default values
- Corresponding values in the Application object and/or Site objects, if supplied
- Internal default values, if some parameters are still missing or are unresolved (such as transfer direction, which is set to *Both* in the default Application)

Gateway always verifies the consistency of Transfer Requests before it records them in the Mailbox. If Gateway cannot find a mandatory transfer parameter in the Transfer Request, the Application, or in the internal default values (such as the file receiver Site), it rejects the Transfer Request and does not add the record to the Mailbox.

Error reporting

When Gateway encounters a problem during the Transfer Request processing, it returns an error. The error format depends on the user interface used:

Interface	Error Format
Online commands	Error message is displayed directly on output
GUI	
API calls	Return value is 1 An error code is automatically loaded into the global variable ErrCod . This code can be used to retrieve the error message using a special API function: GikErrMsg .

Submitting a Transfer Request without an Application

For Gateway to process a Transfer Request submitted without using an Application, you must supply all the file system attributes that Gateway normally retrieves from the Application, in the Transfer Request itself.

Note: You can also supply file system attributes for Transfer Requests that you submit *with* an Application. In this case, Transfer Request parameters override the corresponding Application attributes.

Example

The following command creates a Transfer Request to send a file containing 80-byte fixed-length records. Platform-dependent parameters (such as `sd` in this example) must be supplied if requested for a given platform.

```
peltrans -org_alias locpel -dest_alias xpel  
  
-appli PEL1  
  
-file_component <name_of_file_to_send>  
  
-f_org S  
  
-rec_fmt F -rec_len 80  
  
-x_rec_fmt F -x_rec_len 80  
  
[-sd <system_dependent_param>]
```

Submitting a Transfer Request with an Application

In all modes, the Site and Application objects must belong to the same group. Refer to the table in [About PEL Transfer Requests](#) for parameter information.

Sending a file in Initiator mode

In Initiator mode, one Site represents the Initiator/Sender, while the partner Site is the Responder/Receiver.

The partner automatically receives the file, without submitting any further request.

Example: Sending a file using a TCP/IP network

The following command sends a file with variable-length records (maximum length: 255) and requests ASCII to EBCDIC code conversion.

The Transfer Request uses:

- Application: **var255**
- Remote Site: **tpelpc**

```
peltrans [-org_alias locpel] -dest_alias tpelpc  
  
-appli var255  
  
-file_component <name_of_file_to_send>  
  
-snd_msg tutu
```

Sending a File in Responder mode

The server partner can submit a Transfer Request in Responder/Sender mode. This operation simply registers the Request in the Gateway Mailbox.

The Remote Site initiates the actual file transmission by submitting a Transfer Request in Initiator/Receiver mode.

Example

In the following example, the Local Site **locpel** submits a Transfer Request in Responder/Sender mode, using an Application named **fb80**.

```
peltrans  [-org_alias locpel]

        -dest_alias xpelpc

        -appli fb80

        -direction O  -mode R

        -file_component <name_of_the_file_to_send>
```

Receiving a file in Initiator mode (PEL LOTS request)

An Initiator Site can submit a Transfer Request to receive a file from the partner Site.

The Site receiving the file is Initiator/Receiver, while the partner Site is Responder/Sender.

A Transfer Request in Initiator/Receiver mode is called a LOTS request in PEL terminology. Such a Request is created with its own local identifier, which is just a registration number, and does not correspond to any file data transmission.

However, end-of-transfer exits are scheduled for this type of Transfer Request, just as for Requests associated with actual file transfers.

You can receive up to ten files with the same LOTS Request, depending on the number of Transfer Requests in the Remote Site that meet the LOTS criteria. The Remote Site reply contains the list of Requests selected according to LOTS command criteria. Gateway automatically creates as many file Transfer Requests as the number received in the response to the LOTS command.

The Initiator Site receives the file if the Remote Site previously submitted a Transfer Request in Responder/Sender mode, on condition that:

- Both Transfer Requests use the same Application
- The Remote Site and the Application belong to the same group

Example

In the following example, the Remote Site **xpelpc** is the Initiator/Receiver. This Site submits a Transfer Request to receive files from the Local Site **locpel**.

```
peltrans  -org_alias locpel  -dest_alias xpelpc
```

```

-originator locpel

-appli fb80

-type LOTS

-direction I -mode I

```

The Responder (**locpel**) selects the files to transmit based on the transfer attributes supplied in this LOTS Transfer Request.

Gateway returns a Transfer Request identifier for the received Request (LOTS command), and creates a separate Transfer Request for each file transfer that the LOTS command generates.

Reply Codes

Errors associated with *NON response message

*NON Error number	Meaning
001	Responder out of sync
002	Command does not begin with the character ?
003	Incorrect syntax
004	Confidentiality breached in transmission
005	Confidentiality breached in reception
006	File not found on disk
007	Responder error
008	Responder name error name invalid
009	Application name invalid
00A	File reference error
00B	File not allocated
00C	No entry in the MAILBOX
00D	Incorrect file state
00E	The transfer cannot be restarted
00F	Number of records at zero in TRANS commands
00G	Compression not supported

Errors associated with *NDL response message

*NDL Message number	Meaning
001	*DDL not valid
002	Invalid number of blocks for acknowledgment
003	Invalid result
004	The transfer can not be restarted
999	ABORT

Protocol trace messages

Each protocol command or response corresponds to a protocol trace message that you can record in the log file. The protocol trace is an optional feature that you define for transfer activity for a given Site.

You can enable or disable trace printing for a Site object at either of the following stages by setting the Remote Site parameter `-trace_mode (-tm)` to *Y*(yes) or *N*(no):

- During Site creation
- On modifying attributes of an existing Site
- During Gateway processing (dynamically)

Enabling or disabling protocol tracing using this method is only valid for the current session. The update is lost when you stop Gateway.

[About PEL](#)

[Configuring Gateway for PEL](#)

[Working with File Transfer Requests](#)

[Working with Transfers \(command line\)](#)

Axway Gateway: Protocols

PeSIT Hors SIT

About PeSIT Hors SIT

[PeSIT Hors SIT](#)

[How PeSIT Hors SIT processing works](#)

PeSIT Hors SIT Introduction

PeSIT Hors SIT (or PeSIT) is an end-to-end public protocol designed to provide a reliable and normalized interface for user applications to exchange files and messages between heterogeneous systems, irrespective of the network communication type.

The PeSIT Hors SIT protocol is used on SNA in LU 6.2, TCP/IP and some proprietary communication networks such as DSA, and SPX/IPX.

There are four standard user profiles in the PeSIT protocol: SIT, Hors SIT, secured Hors SIT.

This section describes the implementation and use of the two standardized versions of the PeSIT Hors SIT protocol:

- Version D (November 1987) referred to as **PeSIT HS D**
- Version E (July 1989) - referred to as **PeSIT HS E**

PeSIT common services

Both PeSIT HS D and PeSIT HS E provide the following services:

- File transmission
- File reception
- Setting of synchronization points during transfer
- Restart of interrupted transfers from a synchronization point
- Dynamic synchronization (during transfer)
- Temporary suspension of transfer
- Data compression
- Generic file selection

File transmission

File transmission, also known as file-writing, enables you to transfer data to a remote partner, provided that you have established a logical link between the two users.

File reception

File reception, also known as file-reading, enables you to transfer file data from a partner represented by a Remote Site.

Setting of synchronization points

The data sender can initiate sequentially numbered synchronization points during the transfer. The data receiver can acknowledge the synchronization points to confirm that the data transferred up to that point has been correctly received and saved. This mechanism is used to restart a transfer after an incident.

When you establish the logical connection between partner Sites, the two Sites can negotiate the following:

- Synchronization acknowledgment window
- Interval between synchronization points
- Whether resynchronization is allowed

Restarting interrupted transfers

Transfer restart enables you to restart a transfer interrupted before its completion. The interruption closes the file and disables the protocol and network connections. It is always the Initiator who initiates the restart and the Receiver who decides from which synchronization point the transfer is to be restarted.

Dynamic synchronization

Dynamic synchronization allows you to ask the partner to restart the transfer from a previous synchronization point, if an incident was encountered during data transfer. The file remains open during this operation, which can be initiated by either Initiator or Responder.

Temporary suspension of transfer

Transfer suspension allows you to suspend the current transfer (closing and de-selecting the file) but retain the active connection with the Remote Site. The suspended transfer will be restarted later.

Data compression

Data can be compressed in order to reduce the size of data actually transferred.

Three types of compression are available:

- Horizontal compression: deletes repeated characters in a logical record
- Vertical compression: transmits only that data which differs from the previous record
- Horizontal and vertical compression: combines both of these methods

Generic file selection

When an existing file is remotely selected for read access, the file name sent by the Initiator is no longer just a nominative reference. With PeSIT HS E, it may also be a set of criteria that the server uses to select files. File selection can also be generic. The partner Sites are responsible for handling the fact that several files may meet the given selection criteria.

PeSIT HS E enhancement

In addition to the common PeSIT HS features described above, Gateway supports the following functions for PeSIT HS E only:

- Two-way transfers over the same logical connection
- Transmission error control using CRC (Cyclic Redundancy Checking) on each data block
- Message transfer which enables the transmission of application messages of arbitrary length, or end-to-end response acknowledgment messages
- Store-and-forward mode with automatic routing from one Site to another via intermediate Sites

Two-way transfers over the same logical connection

The PeSIT HS E protocol allows only one file transfer in a given session at a time. Nevertheless, Gateway can perform several concurrent transfers either as:

- *consecutive* transfers within the same session
- several *simultaneous* file transfers on *different* sessions, and in both directions

Transmission error control

Gateway provides a straightforward, efficient transmission error detection mechanism, CRC, for use over unreliable communication networks. The process involves adding two bytes resulting from a polynomial calculation on this data unit to each protocol unit of data sent.

Message transfer

This operation enables items of relatively small information to be exchanged between the Sites. A message is accompanied by a minimum of service information and may, for example, be used to acknowledge the correct processing of a received file.

Gateway enables you to transmit 254-character user messages, unchanged, to the partner Site.

Transfer and file identification

Gateway associates with each transfer a numeric transfer identifier, which is transmitted by the protocol.

The following parameters identify the file to be transferred (each parameter has a PI, Parameter Identifier)

Note: All PeSIT Hors SIT transfers are identified solely by these parameters.

Parameter	PI	Comment
Client Site name	PI 3	
Server Site name	PI 4	
Logical originator protocol identifier	PI 61	For routed transfers
Logical destination protocol identifier	PI 62	For routed transfers
File type	PI 11	Integer between 0 and 65635
File name	PI 12	Application logical file name
Transfer identifier	PI 13	Transfer ID number generated by the initial file originator

Transfer modes

During a session between two Sites using the PeSIT protocol, the dialog is always asymmetrical: the two Sites play different complementary roles. The requester of the session, the Initiator, also takes the initiative regarding transfers, while the Responder (server Site) executes the requests received from the Initiator.

The Initiator sends files to the Responder (write), or the Responder sends files to the Initiator (read).

A given partner can have the following roles:

- Initiator/Sender
- Initiator/Receiver
- Responder/Sender
- Responder/Receiver

Password management

During the pre-connection and connection phases, passwords can be exchanged and validated by the two conversing partners.

There are three types of password that you define in the Remote Site object:

- Pre-connection password
- Connection password sent by the Initiator (client: FPDU CONNECT PI 5)
- Connection acknowledgment password sent by the Responder (server: FPDU ACONNECT PI 5)

For more information on how to define passwords, refer to [CGates: Protocol Connection](#).

The Initiator can send ID and password details in the PeSIT pre-connection message. The Responder then validates them.

If the pre-connection message exchange is successful, the Initiator can send the connection FPDU with the connection password, which is subsequently validated by the Responder. The Responder can then send its own password back in the connection acknowledgment FPDU to identify itself to the Initiator.

Note: In the PeSIT Hors SIT protocol specification, password management is an optional feature. If your partner Site does not support the same features, leave the `check_password` and `check_password2` fields blank.

File and transfer attributes

The protocol negotiates the size of data blocks exchanged, as well as the compression type.

During the connection phase, the two Sites negotiate the synchronization point frequency and the acknowledgment data window.

How PeSIT Hors SIT processing works

Interaction diagrams

PeSIT provides a set of user services and a protocol dialogue used by two Sites to exchange files and messages. A service defines the interaction between two PeSIT end users.

Four types of service primitives implement this interaction:

Primitive	Notation	Description
Request	REQ	Request a given service
Indication	IND	Notify the arrival of a service request
Response	RSP	Answer a request
Confirmation	CNF	Confirm the arrival of a response to a previous request

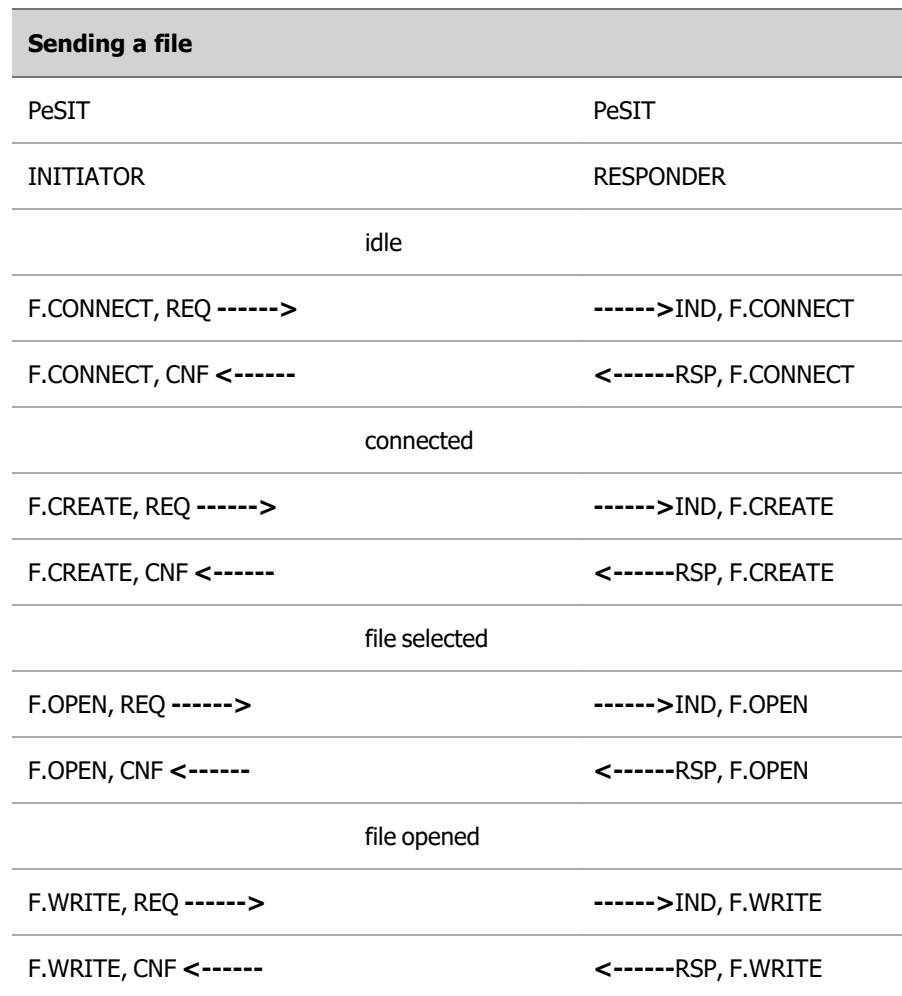
At the very minimum, a service involves a request submitted by a user and its corresponding indication to the partner. Some services also require a response and a confirmation to be complete.

The primitives provide functions for the following:

- File-writing
- File-reading
- Synchronization
- Transfer restart
- Resynchronization during the transfer
- Transfer suspension
- Data compression
- Error checking
- Message transfer

File transmission

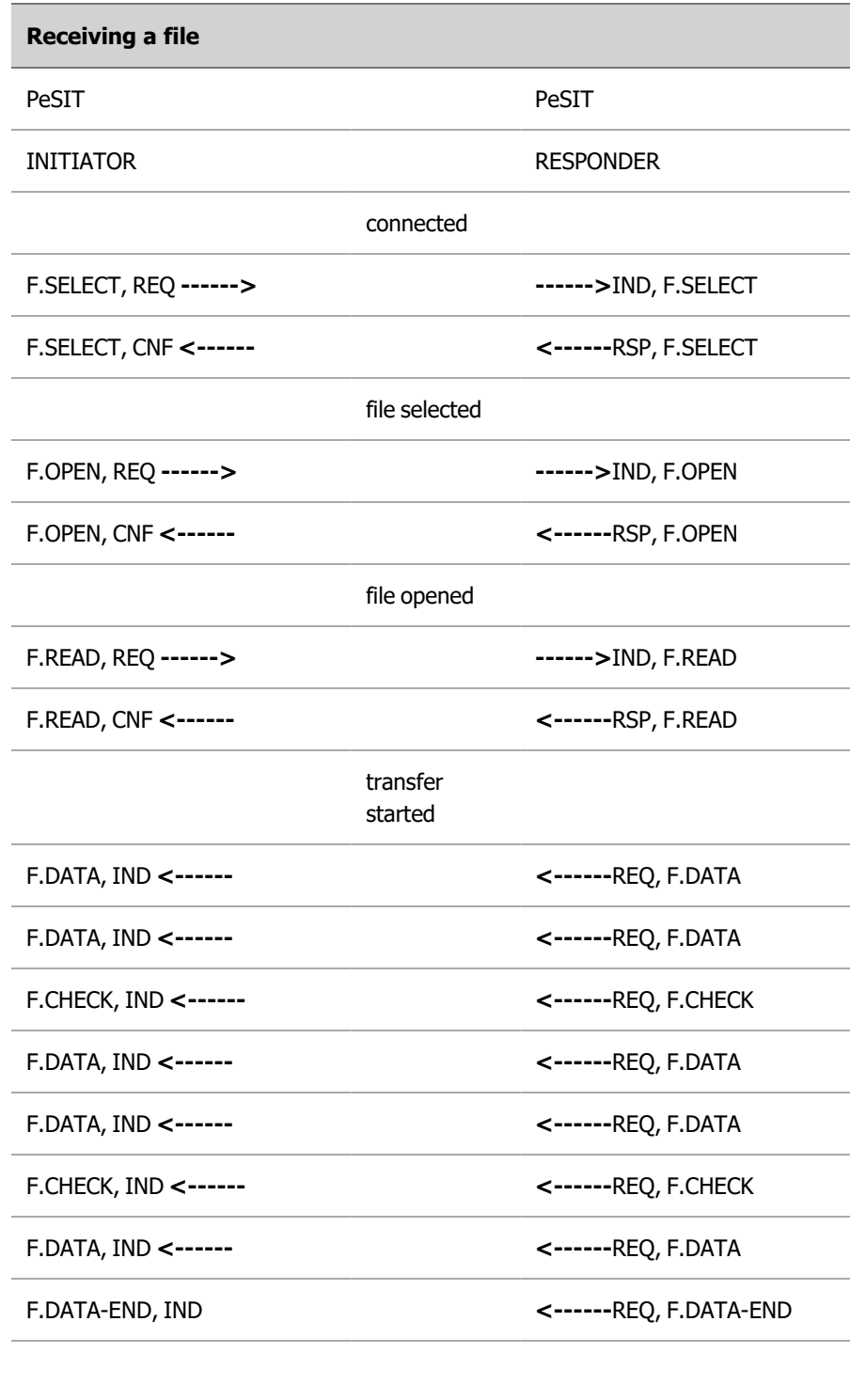
The following interaction diagram illustrates the communication flows between Initiator and Responder during file transmission.



Sending a file		
transfer started		
F.DATA, REQ ----->	data transfer	----->IND, F.DATA
F.DATA, REQ ----->		----->IND, F.DATA
F.CHECK, REQ ----->	synchronization point	----->IND, F.CHECK
F.DATA, REQ ----->		----->IND, F.DATA
F.DATA, REQ ----->		----->IND, F.DATA
F.CHECK, REQ ----->		----->IND, F.CHECK
F.DATA, REQ ----->		----->IND, F.DATA
F.DATA-END, REQ ----->		----->IND, F.DATA-END
F.TRANSFER-END, REQ ----->		----->IND, F.TRANSFER-END
F.TRANSFER-END, CNF <--- --		<-----RSP, F.TRANSFER-END
transfer ended		
F.CLOSE, REQ ----->		----->IND, F.CLOSE
F.CLOSE, CNF <-----		<-----RSP, F.CLOSE
file closed		
F.DESELECT, REQ ----->		----->IND, F.DESELECT
F.DESELECT, CNF <-----		<-----RSP, F.DESELECT
file deselected		
F.RELEASED, REQ ----->		----->IND, F.RELEASED
F.RELEASED, CNF <-----		<-----RSP, F.RELEASED
disconnected		

File reception

The following interaction diagram illustrates the communication flows between Initiator and Responder during file reception.



Receiving a file	
F.TRANSFER-END, REQ ----->	----->IND,F.TRANSFER-END
F.TRANSFER-END, CNF <-----	<-----RSP,F.TRANSFER-END
transfer ended	
F.CLOSE, REQ ----->	----->IND, F.CLOSE
F.CLOSE, CNF <-----	<-----RSP, F.CLOSE
file closed	
F.DESELECT, REQ ----->	----->IND, F.DESELECT
F.DESELECT, CNF <-----	<-----RSP, F.DESELECT
file deselected	

Message transmission

The following interaction diagram illustrates the communication flows between Initiator and Responder during message transmission.

Sending a message	
PeSIT	PeSIT
INITIATOR	RESPONDER
idle	
F.CONNECT, REQ ----->	----->IND, F.CONNECT
F.CONNECT, CNF <-----	<-----RSP, F.CONNECT
connected	
F.MESSAGE, REQ ----->	----->IND, F.MESSAGE
F.MESSAGE, CNF <-----	<-----RSP, F.MESSAGE
F.RELEASED, REQ ----->	----->IND, F.RELEASED

Sending a message

F.RELEASED, CNF <-----

<-----RSP, F.RELEASED

disconnected

Protocol messages

Message structure

The messages that carry out the dialogue between the Initiator and the Responder are referred to as FPDUs (**F**ile transfer **P**rotocol **D**ata **U**nits). Each FPDU comprises a header (six bytes) that specifies the type of data transferred.

The body of the FPDUs is formed of a series of parameters used to specify and negotiate all the transfer elements. Each parameter, named PI (Parameter Identifier), is assigned a number and value as defined in the PeSIT protocol specifications.

Each parameter group unit is identified by a numeric code named **PGI** (**P**arameters **G**roup **I**dentifier).

For more information on PeSIT HS protocol, refer to documentation published by the GSIT (Groupement pour un Système Interbancaire de Télécompensation): *PeSIT protocol, Technical Specifications - version E*.

Examples of message structures (LI is the length indicator)

Message with two parameters (PI)

FPDU Header

TML	PI	LI	VALUE	PI	LI	VALUE

Message with one PGI containing two parameters (PI)

FPDU Header

TML	PGI	LI	PI	LI	VALUE	PI	LI	VALUE

List of PI codes

When labeled *not used* the PIs are not sent, or are ignored on reception.

PI	Values	Meaning
1	0 = not used 1 = used	Use of a CRC (1 byte) Not used
2	byte 1: error type bytes 2-3: reason code	Diagnostic giving the type of error encountered (3 bytes)
3		24-character parameter identifying the Initiator and defined by Site object protocol identifier: • Initiator/Sender or Responder/ Receiver in the origin Site • Initiator/Receiver or Responder/ Sender in the destination Site
4		24-character parameter identifying the Responder and defined by the Site object protocol identifier: • Initiator/Sender or Responder/ Receiver: Destination Site • Initiator/Receiver or Responder/ Sender: Originator Site
5		Access control
6	1: version D 2: version E	Protocol version number. Defined in conffile by the parameter <code>ver_num</code>
7	b1b2=0: no synchronization point b3=0: no acknowledgment	Synchronization option (3 bytes) Bytes 1-2: synchronization interval (in KB) Byte 3: synchronization window Negotiated in connection phase
11	= 0: no specific action	Numeric parameter providing the file type as specified in the protocol documents Transfer Request <code>file_type</code> parameter
12	ASCII string	File name Transfer Request <code>file_name</code> parameter
13		Transfer identifier: numeric value generated by the Sender
14	bitmap (1 byte)	Requested attributes

PI	Values	Meaning
15	0 = new transfer 1 = transfer restarted	Transfer restarted
16	0 = ASCII 1 = EBCDIC 2 = BINARY	Data code
17	0 = urgent 1 = normal 2 = lowest priority	Transfer priority Supplied by Transfer Request parameters
18	0 = beginning of file	Restart point Supplied by Gateway
19	4 = error (restart to follow) 8 = suspension 12 = canceled by Responder 16 = canceled by Initiator	End-of-transfer code
20	0 = beginning of file	Synchronization point number Supplied by Gateway
21	byte 1: 0 = No compression 1 = compression byte 2: 1 = horizontal comp 2 = vertical comp 3 = both	Compression type: numeric value (2 bytes)
22	0 = Read 1 = Write 2 = RW	File access type
23	0 = F.RESTART authorized 1 = not authorized	Resynchronization specifying whether F.RESTART is authorized or not Defined in Site object <code>restart_allowed</code> parameter

PI	Values	Meaning
25	must be > 800 bytes	Maximum length of data block Supplied by configuration parameter <code>sdu_max_len</code>
27		Data byte count
28		Record count Mandatory if PAD is used
31	=0: fix =80h: variable	Record format (1 byte)
32		Record length (2 bytes)
33	=0: sequential =1: relative =2: indexed	File organization
37		File label: not used (256 bytes)
38		Key length in the record (2 bytes) Parameter present if PI 33 = 2
39		Key offset in the record Parameter present if PI 33 = 2
41	=0: KB =1: record	Space allocation unit Supplied by Transfer Request <code>alloc_unit</code> parameter
42		Maximum value of space reservation Supplied by Transfer Request <code>max_alloc_size</code> parameter
51		Date and time of transfer creation Supplied by Gateway
52		Date and time of last extraction Supplied by Gateway
61	24-byte char string	Initial sender of the file In routing mode, it is the Transfer Request Originator Site

PI	Values	Meaning
62	24-byte char string	Final receiver of the file In routing mode, it is the Transfer Request Destination Site
91		Message
99	Free format message	Usually supplied by Transfer Request <code>snd_msg</code> parameter. The supported message length is: • 512 bytes on PeSIT E • 64 bytes on PeSIT D It is possible to send PI 99 messages both : • on transfer-related phases: file creation/selection and deselection, and • on connection related phases: connect and release/disconnect. Only the values exchanged in transfer-related phases are recorded on the mailbox entry associated with the transfer. Values exchanged in the latest phase overwrite those exchanged in previous phases.

File transfer phases

The PeSIT protocol processes file transfers in following phases:

Phase	Processing
Connection/ Disconnection	Connection establishment and connection release initiated either by the user or the protocol
File selection/ deselection	File creation, selection/ de-selection, and message transfer
Opening/ Closure	Opening and closing of file services
Transfer	Reading / writing files, sending data, setting of the synchronization points, transferring, resynchronization
Interruption	Restart of a transfer, end-of- data transmission, end of transfer

The processing passes through these phases in the sequence illustrated below.

- 1- CONNECTION phase
- 2- FILE SELECTION phase
- 3- FILE OPENING phase
- 4- DATA TRANSFER phase
- 4- END OF DATA TRANSFER
- 3- FILE CLOSING
- 2- FILE DESELECTION

- 2- FILE SELECTION
- 3- FILE OPENING
- 4- DATA TRANSFER
- 4- END OF DATA TRANSFER
- 3- FILE CLOSING
- 2- FILE DESELECTION
- 1- DISCONNECTION

Each phase is a step in the file transfer process. It is not possible to leave a phase without closing all the phases it contains.

Axway Gateway: Protocols

Configuring the PeSIT Hors SIT environment

[About PeSIT Hors SIT configuration](#)

[Creating a Site](#)

[Creating an Application](#)

[Configuring Change Direction support](#)

[Configuring negative acknowledgments](#)

[Canceling transfers on connection refusal](#)

About PeSIT Hors SIT configuration

Before submitting a file transfer, you must define the following Gateway objects if they do not already exist:

- **Local Site**

On installation, you must create a Local Site corresponding to the alias name and protocol identifier of your Gateway system. This is the default Local Site for all protocols supplied by the product, including PeSIT. You can create a specific PeSIT Local Site if necessary

- **Remote Site**

Define your partner Site and communication attributes in the Remote Site object. For the PeSIT service implementation, a Remote Site object must exist for each end user

- **Application**

Define at least one Application

You can use either the `peladm` online command, as in the examples used here, or the GUI to create these objects.

Creating a Site

If you do not supply attributes explicitly, Gateway uses either default values from the operator interface or from Gateway internal defaults.

Some platforms may require values for optional attributes that differ from the defaults. Refer to your platform-specific documentation for more information.

If you do not specify a group for the Remote Site that you create, Gateway assigns the group name GDEFAULT.

Example 1: Creating a Local Site locpsit

```
peladm create_loc_site -alias locpsit  
  
-pi locpsit  
  
-comments "PHSD Local Site"
```

Example 2: Creating a Local Site acting in Responder mode only

```
peladm create_loc_site -alias locpsit1  
  
-pi locpsit1  
  
-no_i_s  
  
-no_r_r
```


Example 3: Creating a Remote Site using TCP/IP and the following attributes

- PeSIT Hors SIT E
- Remote IP address: 193.56.234.20
- Port number: 22222
- Maximum buffer size: 2048 bytes
- Synchronization acknowledgment window: 8 (KB)
- Maximum number of simultaneous server connections: 10
- Responder/Sender mode: disabled

```
peladm create_site -alias tcpphse

    -protocol PHSE

    -comments "Remote Site: PESIT version E, TCP"

    -comm_type TCP

    -dest_address 193.56.234.20/22222

    -data_size_max 2048

    -resp_conn_max 10

    -no_r_s
```

Creating an Application

Gateway completes any attributes that you do not define with default values from either the operator interface, or from internal default attributes.

If Application does not exist in this scenario	Gateway uses this default
For file reception in Responder mode	<i>DEFAULT_X</i> where X can take the values: • A for ASCII transfers • E for EBCDIC transfers • B for binary transfers
No <i>DEFAULT_X</i> Applications are defined	Application <i>DEFAULT_X</i> (if found) is used Note: Applications with these names must be defined in both partner Sites.

Example 1: Defining the default Applications named DEFAULT_A and DEFAULT_B

```
peladm create_appli -n DEFAULT_A
```

```
-text_file Y  
peladm create_appli -n DEFAULT_B
```

Example 2: Creating an Application to send a fixed-length file

This example creates an Application named **FIX255** to send a file that contains **255**-byte fixed-length records.

```
peladm create_appli -n fix255  
  
-rec_fmt F  
  
-rec_len 255  
  
-x_rec_fmt F  
  
-x_rec_len 255
```

Example 3: Creating an Application to send a variable-length file

This example creates an Application named **VAR255** to send a file that contains variable-length records with a maximum length of 255.

```
peladm create_appli -n fix255  
  
-rec_fmt V  
  
-rec_len 255  
  
-x_rec_fmt V  
  
-x_rec_len 255
```

Configuring Change Direction support

PeSIT Hors SIT Change Direction (CD) support is only available in PeSIT Hors SIT E.

The PeSIT Hors SIT protocol enables a client application (the connection Initiator) to call a server (the connection Responder), and then to send files to, or to ask to receive files from the server. The server application waits for client connection requests to receive files from the client, or to transmit files to the client upon a selection request.

The official protocol specification does not allow a server application to send files or a message, on its own initiative, to the client application. This is normally not a problem since a file transfer monitor can initiate as well as receive connection requests, thus assuming the roles of client and server alternatively during different sessions.

However, under more restrictive circumstances, such as a file transfer monitor situated within a DMZ (Demilitarized Zone), it may be forced to disable the client role, which means it is no longer allowed to initiate a connection request. In such cases, there is no way to send a PeSIT Hors SIT message (either an application or acknowledgment message) from the server to the client.

Note: PeSIT Hors SIT Change Direction Support is a proprietary extension to the official PeSIT Hors SIT E protocol specification.

PeSIT Hors SIT Change Direction Site parameters

To implement PeSIT Hors SIT Change Direction support, four specific parameters are available when defining a PeSIT Hors SIT E Remote Site.

Parameter	Value	Description
Change Direction Option	YES NO	Indicates whether the Remote Site supports the Change Direction extension or not. If you do not set this option, the following parameters are ignored.
Polling Activation Mode	YES NO	Indicates whether or not the Remote Site in question is polled when you start Gateway. This parameter also enables the CD connection to be dynamically activated or deactivated (see <code>pelctl</code> command below).
Polling Interval	mm (minutes)	Defines the time to wait before starting the next calling cycle.
Maximum Connections	Nn (integer)	Defines the upper limit of concurrent active connections in Change Direction mode. Note: The CD connections are counted separately from the normal Initiator and Responder connections established with this Site.

Example

Client Side Monitor

```
remote_server_pesit_hs_site
```

```
CD option : YES
```

```
CD mode : YES
```

```
CD interval : 60
```

```
CD max conn : 10
```

Server Side Monitor (within DMZ)

```
remote_client_pesit_hs_site
```

```
CD option : YES
```

```
CD mode : NO
```

```
CD interval : 0
```

```
CD max conn : 0
```

Online commands for Site parameters

Use the online commands:

- `peladm create_site` and `peladm update_site` to define the Remote Site features
- `pelctl start_site` and `pelctl stop_site` to enable the CD connection to be dynamically activated or deactivated.

The specific parameters required to configure this feature are:

Command	Parameter	Comment
<code>peladm create_site</code>	<code>-poll_option (-poll) Y N</code>	Define the Remote Site features as described here.
	<code>-poll_mode (-pm) Y N</code>	The default values are NO and zero. Other parameters are unchanged.
<code>peladm update_site</code>	<code>-poll_interval (-pollint) mm</code>	
	<code>-poll_conn_max (-pollmax) nn</code>	
<code>pelctl start_site</code>	<code>-poll_mode (-pm)</code>	The value set by these two commands overrides that defined in the Site object. However, this value is transient and valid only for the current Gateway session. At the next Gateway startup session, this field reverts to the value specified in the Site definition.
<code>pelctl stop_site</code>		

How it works

When you define a Remote Site to support the CD option, with `poll_mode=Y`:

- The client application automatically initiates a PeSIT Hors SIT connection with a modified version (with PI 6 bit 7 set to 1) to indicate that CD support is desired
- The server application can either accept or refuse the connection

Once the protocol connection is established and the two partners are successfully authenticated, the client application sends a SELECT request with a modified PI 14 (set to 0xFF) to indicate that a CD is requested.

Recognizing the modified SELECT request, the server application answers with an ACK.SELECT with a modified PI 33 (set to 0xFF) to acknowledge the CD request.

Thereafter, the client application switches to Responder mode and the server application to Initiator mode.

The server is then free to send files or messages as defined by the formal protocol specification.

If no files or messages are available, the server application can release the connection and the client application retries on the next cycle as defined by the scheduling parameters.

If the client application, after exchanging SELECT/ACK.SELECT, receives a file or message transmission indication, it either:

- immediately schedules another connection request to see if there are more files or messages to process (on condition that the maximum number of concurrent connections is not reached)
- launches the next cycle when one of the current connections is terminated

Configuring negative acknowledgments

Negative acknowledgment (PeSIT Hors SIT Negative EERP Support) is only available in PeSIT Hors SIT E.

The PeSIT Hors SIT E protocol supports an EERP (end-to-end) acknowledgment. In this positive acknowledgment, the final destination of a file transfer can send a message to the transfer originator to acknowledge reception of the file. However, the formal protocol specification does not specify how to send a negative acknowledgment to signal whether errors occurred. For example, errors detected or generated by a user application or by the intermediate Sites during Store-and-Forward processing.

Note: PeSIT Hors SIT Negative EERP Support is a proprietary extension to the official PeSIT Hors SIT E protocol specification.

PeSIT Hors SIT E Site Negative EERP parameters

To implement PeSIT Hors SIT negative EERP support, a new parameter is included in the definition of a PeSIT Hors SIT E Remote Site.

Negative acknowledgment option (`-nack_option`): YES | NO

The NACK (Negative acknowledgment) option indicates whether the Remote Site supports the negative EERP extension or not. If this option is not set to Yes, all negative EERP requests are denied.

Online commands for Site parameters

Use the online commands:

- `peladm create_site` and `peladm update_site` to define the Remote Site features
- `peltrans` to specify negative EERP

The specific parameters required to configure this feature are:

Command	Parameter	Default values
<code>peladm create_site</code>	<code>-nack_option (-nao) Y N</code>	N
<code>peladm update_site</code>		

Command	Parameter	Default values
peltrans	<code>-nack_by_user (-nack) Y N</code>	

How it works

When you define a Remote Site to support negative EERP:

- The client application initiates a PeSIT HS connection with a modified version (with PI 6 bit 8 set to 1) to indicate that negative EERP support is requested
- The server application can either accept or refuse the connection
- Once the protocol connection is established and the two partners are successfully authenticated, the client application can send a MESSAGE request with a modified PI 14 to indicate that this is a negative EERP

Note: PI 14 set to 0xFE indicates a user-initiated negative EERP. PI 14 set to 0xFF indicates a Gateway-initiated negative EERP.

The message contents are provided either by a user application or generated automatically by Gateway, as in the case of a normal EERP.

Recognizing the modified PI 14, the server application updates the file transfer status accordingly.

Canceling transfers on connection refusal

If required, you can cancel transfers to a Site that is reachable but refuses connections. To do this, you modify the `cancel_transfer_on_connection_refusal` advanced parameter in the Gateway configuration menu.

Refer to [Configuration: Gateway parameters - Advanced parameters and Trace levels](#).

The value of this parameter only affects transfers made using the PeSIT protocol.

[Working with Local Sites](#)

[Working with Local Sites \(command line\)](#)

[Working with Remote Sites](#)

[Working with Remote Sites \(command line\)](#)

[Working with Applications](#)

[Working with Applications \(command line\)](#)

[Executing a basic PeSIT transfer](#)

Submitting a PeSIT Hors SIT Transfer Request

[About the PeSIT Hors SIT Transfer Request](#)

[Submitting a Transfer Request without an Application](#)

[Submitting a Transfer Request with an Application](#)

[Sending a message](#)

[Sending an EERP acknowledgment](#)

[PeSIT HS E file reception requests](#)

About the PeSIT Hors SIT Transfer Request

Submit Transfer Requests using the `peltrans` online command, the GUI, or user programs linked with APIs. Each of these interfaces transmits transfer parameters to Gateway.

The examples provided in this section were created with the `peltrans` online command, which has the following syntax:

```
peltrans <argument> value <argument> value<argument> value
```

Command arguments are keywords, each followed by the corresponding value. Keywords are prefixed by a "-" character, and most of them have abbreviated names (as shown in parentheses in the table below).

The following table lists Transfer Request parameters relevant to the PeSIT protocol. For details of all available parameters, refer to [Working with Transfers \(command line\)](#).

Note: Set the size of data buffers transmitted on the network (defined by the `data_size_max` parameter in the Remote Site) greater than the maximum record size (defined by the `rec_len` and `x_rec_len` parameters in the Application).

Transfer Request parameters Sending and Receiving files				
Keyword name(*)	PI	Sending Internal default	Receiving Application default	Comments
<code>-org_alias (-oa)</code>	3 (61) (1)			Originator Site alias name, which must be a Remote Site defined with PeSIT protocol and the appropriate version
<code>-dest_alias (-da)</code>	4 (62) (1)			Destination Site alias name, which must be a Local Site name

-file_type (-ft)	11			Mandatory
-file_name (-fn)	12			Mandatory
-compress (-comp)	21	U	U	U = no compression
-hist_create_date (-hd)	51			PeSIT transfer creation date
-prio	17	0	1	Transfer priority
-f_org	33		S	File organization
-rec_fmt (-rf)	31	S	F	Record format
-rec_len (-rl)	32	512	0	Record length
-key_offset	39	0	0	
-key_len (-kl)	38	0	0	
-data_code (-dc)	16	B		B = binary
-x_rec_fmt (-xrf)	31	S		Transfer record format
-x_rec_len (-xrl)	32	512		Transfer record length
-x_data_code (-xdc)	16	B		Transfer data code
-interval				Interval between Remote Site polling (LOTS and POLL request types)
-text_file		N	N	Y/N
-snd_user	3			Store-and-Forward only: sender Id
-snd_appli	3			Store-and-Forward only: sender application Id
-snd_text	3			Store-and-Forward only: sender free text
-rcv_user	4			Store-and-Forward only: receiver Id
-rcv_appli	4			Store-and-Forward only: receiver application Id
-rcv_text	4			Store-and-Forward only: receiver free text
-maxi-files			1	Maximum number of files allowed

-maxi-reqs			1	Maximum number of requests allowed

(*) Abbreviations in parentheses

(1) PI value for Store-and-Forward mode

Submitting a Transfer Request without an Application

For Gateway to process a Transfer Request submitted without using an Application, you must supply all the file system attributes that Gateway normally retrieves from the Application, in the Transfer Request itself.

For this type of transfer, Gateway uses a default Application named *DEFAULT*. An Application with this name must exist in both Sites.

Note: You can also supply file system attributes for Transfer Requests submitted *with* an Application. In this case, Transfer Request parameters override the corresponding Application attributes.

Example

The following command creates a Transfer Request to send a file containing 80-byte fixed-length records. Platform-dependent parameters (such as *sd* in this example) must be supplied if requested for a given platform.

```
peltrans -org_alias locpsit -dest_alias x25phsd
        -appli DEFAULT
        -file_name DEFAULT -file_type 0
        -file_component <name_of_file_to_send>
        -f_org S
        -rec_fmt F -rec_len 80
        -x_rec_fmt F -x_rec_len 80
        [-sd <system_dependent_param>]
```

Submitting a Transfer Request with an Application

In all modes, the Site and Application objects must belong to the same group. Refer to the table in the Overview section for parameter information.

Sending a file: Initiator mode

In Initiator mode, one Site represents the Initiator/Sender, while the partner Site is Responder/Receiver.

The partner automatically receives the file, without submitting any further request.

Example: Sending a file using TCP/IP and PeSIT HS E

The following command sends a file with variable-length records. The maximum value length is 255.

The Transfer Request uses:

- Application **var255**
- Remote Site **tpelpc**

```
peltrans -org_alias localpsit  
          -dest_alias tcpphse  
          -appli var255  
          -file_name var255  
          -file_component <name_of_file_to_send>  
          -snd_msg send_message_text_sample
```

Sending a file: Responder mode

The Responder partner can submit a Transfer Request to send a file. Such requests are registered in the Mailbox and the Remote Site representing the partner initiates the actual file transmission, by submitting a Transfer Request in Initiator/Receiver mode.

Example

In the following example, the Local Site **locpsit** submits a Transfer Request in Responder/Sender mode, using the Application **fix255**.

```
peltrans -org_alias locpsit  
          -dest_alias x25phsd  
          -appli fix255  
          -file_name fix255  
          -direction O  
          -mode R  
          -file_component <name_of_file_to_send>\
```

Receiving a file: Responder mode

The Responder Site is the file receiver. This mode is used when the Remote Site submits a Transfer Request to send a file. The receiver side automatically receives the file, without submitting any further request.

Receiving a file: Initiator mode

A file-reading feature allows a Site to submit a Transfer Request to receive a file made available by the partner Site. The Site receiving the file is Initiator/Receiver, while the partner Site is Responder/Sender.

The Initiator Site actually receives the file if the Remote Site has previously submitted a Transfer Request in Responder/Sender mode.

Both requests must use the same Application, the same file type and file name protocol parameters.

Example

In the following example, the Remote Site **x25phsd** submits a Transfer Request to receive the file made available by the Local Site **locpsit**.

```
peltrans -org_alias locpsit
         -dest_alias x25phsd
         -direction I
         -mode I
         -appli fix255
         -file_name fix255
```

Note: This type of Transfer Request can only transmit one file.

Sending a message

PeSIT Hors SIT message transmission is only available in PeSIT Hors SIT E.

To send a PeSIT HORS SIT message, you must specify at least the following parameters:

- Request type (must be MSG)
- Initial message originator (origin alias)
- Final message destination (destination alias)
- Protocol file name
- Message to send

You can supply the message content either in a message file or as an in-line message. If both are specified, the in-line message is used.

Message Request parameters

The following table lists PeSIT HS relevant parameters for sending a message request using the **peltrans** command. Refer to [Working with Transfers \(command line\)](#) for information on all available parameters.

Sending a message: Transfer Request parameters			
Keyword names (*)	Default	PI	Comments
-type	MSG		Identifies transfer as a message request
-org_alias (-oa)		3 (61) ⁽¹⁾	Origin alias name
-dest_alias (-da)		4 (62) ⁽¹⁾	Destination alias name
-file_name (-fn)		12	Protocol file name
-snd_msg (-sm)		91	User message
-retry_count_max (-rc_max)	5		Maximum number of retries allowed

(*) Abbreviations in parentheses

(1) PI value for Store-and-Forward mode

Sending an EERP acknowledgment

The EERP acknowledgment feature is only available in PeSIT Hors SIT E.

To send a PeSIT HORS SIT end-to-end-response (EERP) acknowledgment, you must specify at least the following parameters:

- Request type (must be EERP)
- Transfer ID to reply to
- Message to send

You can supply the message content either in a message file or as an in-line message. If both are specified, the in-line message is used.

acknowledgment Request parameters

The following table lists PeSIT HS relevant `peltrans` parameters for an end-to-end-response acknowledgment message request.

For more information on the `peltrans` command, refer to [Working with Transfers \(command line\)](#).

EERP acknowledgment message request parameters			
Keyword names (*)	Internal Default	PI	Comments
-type	EERP		Identifies an acknowledgment request
-reply_to_xfer			ID of the transfer to reply to
-snd_msg (-sm)		91	User message

(*) Abbreviations in parentheses

PeSIT HS E file reception requests

With PeSIT HS E, you can specify a SELECT request to receive a variable number of files made available on the server.

Apart from the parameters required for a normal PeSIT HS transfer, three parameters govern the selection request behavior:

- `Max_file`: the maximum number of files to receive. Accepted values are from 0 to 32000.
- `Max_request`: the maximum number of attempts that can be made. Accepted values are from 0 to 32000
- `Interval`: the interval (in seconds) between two attempts

Use notes

- To receive a variable (indeterminate) number of files, specify:
`max_file = 0, max_request = n (with n > 0)`
to set an upper limit for the number of files to receive.
- The selection request ends when no more files are available or when the maximum number (n) is reached.
- To receive a fixed number of files, specify:
`max_file = m (with m > 0), max_request=0, interval = s`
- The selection request ends when *m* files have been received. If less than *m* files are made available at the moment of the request, successive attempts are made every *s* seconds, as specified by the interval parameter.
- If you specify `max_request = n (with n > m)`, the selection request ends when all *m* files have been received or *n* attempts have been made.
- For purposes of backward compatibility, if neither `max_file` nor `max_request` are specified, an attempt is issued to receive a single file.

Case of a transfer to an existing file name

To view in detail the actions Gateway performs on a pre-existing file before receiving the transfer via PeSIT or OFTP of a file with the same name, read: [Transfer to an existing file name on page 1192](#)

Axway Gateway: Protocols

Monitoring a PeSIT Hors SIT transfer

[PeSIT diagnostic codes](#)

[Protocol trace messages](#)

PeSIT diagnostic codes

The following protocol diagnostic return codes are found in end-of-transfer exits and in the log file:

PeSIT diagnostic codes	
000	Success
100	Transmission error
200	Insufficient file attributes
201	System resources temporarily insufficient
202	User resources temporarily insufficient
203	Non-priority transfer
204	File already exists
205	File does not exist
206	Receiving the file will exceed the disk quota
207	File busy
208	File too old (predates D-2 in the SIT context)
209	Message type not accepted on the specified installation
210	Presentation context negotiation failure
211	File cannot be opened
212	File cannot be closed normally
213	Blocking I/O error
214	Restart point negotiation failure
215	System error
216	Intentional premature shutdown
217	Too many synchronization points without acknowledgment
218	Resynchronization not possible
219	No more file space
220	Record longer than expected
221	End-of-transmission timeout
222	Too much data without synchronization point
223	Transfer ended abnormally
224	The file sent is bigger than the one specified in F.CREATE
225	Workstation application congestion: the file was received but CRS could not transfer it to the workstation application
226	Transfer refused
299	Other
300	Congestion in local communication system
301	Called party identification not known

302	Called party not attached to an SSAP
303	Congestion in the remote communication system (too many connections)
304	Initiator identification not allowed (security)
305	Negotiation failure: SELECT
306	Negotiation failure: RESYN
307	Negotiation failure: SYNC
308	Version number not supported
309	Too many connections already in progress for this CT
310	Network fault
311	Remote PeSIT protocol error
312	Requested service closed
313	Connection interrupted at the end of TD inactivity time
314	Unused connection disconnected to allow a new connection
315	Negotiation failure
316	Connection disconnected due to administration command
317	Timer failure
318	Mandatory PI missing or illegal PI content
319	Incorrect number of bytes or records
320	Too many resynchronizations for a transfer
321	Call standby number
322	Call back later
399	Other

Protocol trace messages

Each protocol command or response corresponds to a protocol trace message that you can record in the log file. The protocol trace is an optional feature that you define for transfer activity for a given Site.

You can enable or disable trace printing for a Site object at either of the following stages:

- During Site creation
- On modifying attributes of an existing Site
- During Gateway processing (dynamically)

Enabling or disabling protocol tracing using this method is only valid for the current session. The update is lost when you stop Gateway.

PeSIT specific user exits

Caution Please note that care should be taken when **customizing user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

Gateway provides the following PeSIT user exits that you can customize according to your requirements:

- [ExitPesitPreConnection](#):
PeSIT pre-connection user exit routine. On reception of a PeSIT pre-connection message, this exit routine is called to validate the message content (user ID and password).
- [ExitPesitMsg](#):
This exit routine is called on reception of an FPDU_MSG. On return, FPDU_ACKMSG is sent to the partner Site with diagnostic code (PI 2) equal to zero.

[Internal user exits](#)

Axway Gateway: Protocols

SFTP

[SFTP](#)

[Using SFTP in Gateway](#)

SFTP Introduction

The SSH File Transfer Protocol (SFTP) is an application-level protocol used as a generic protocol for communication between users, proxies, or gateways and other Internet systems.

Gateway supports SFTP versions 0, 1, 2, 3 and 6 in client and server modes. SFTP is defined as standard file transfer with SSH (Secure SHell) providing secure login over an insecure network. SFTP has no inherent identification mechanism and therefore relies on the SSH (version 2) protocol for user authentication and a secure connection.

The SSH and SFTP protocols are separate and technically different, but are used together exclusively in Gateway. The SFTP layer is above the three protocol layers used by SSH (the TLP, UAP and CP).

To [configure an SFTP/SSH link](#), you configure several compatible objects in Gateway:

- Remote Site
- Network Profile object (*netprof*)
- SSH Profile object (*sshprof*)

SSH provides the following services:

- **Authentication:** Verifies the claimed identity of the host server or client user via a private/public key pair, using either of these key exchange algorithms:
 - DSA
 - RSA
 - Keys included in an x.509 v3 certificate (client mode only)
 - Diffie-Hellman exchange (diffie-hellman-group1-sha1 or diffie-hellman-group14-sha1) or preliminary exchange of new groups (diffie-hellman-group-exchange-sha1 or diffie-hellman-group-exchange-sha256)
 - Elliptic Curves Diffie Hellman over predefined prime groups (ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521)

Note: Gateway does not support Kerberos or GSSAPI authentication in either SFTP server or client mode.

- **Integrity:** Adds a message abstract (also referred to as a *hash code* or *message digest*) to the data to ensure that it arrives at its destination intact. In SSH the hash code, referred to as **MAC** (Message Authentication Code), is added to each message sent using the standard hash functions **SHA-1** or **MD5**. See the *Security Guide* > [Certificates and keys](#) for further information (requires login).
- **Confidentiality** (also referred to as Privacy): Prevents third parties from accessing data that is not intended for them. Typically, data is protected using either **symmetrical** or **asymmetrical** encryption techniques. See the *Security Guide* > [Gateway security features](#) for further information (requires login).

Note: Although both protocols can serve the same purpose, SFTP with SSH and FTP with SSL do not function in the same way. See the *Security Guide* > [SSH protocol](#) and *Security Guide* > [SSL and TLS protocols](#) for further information

Using SFTP in Gateway

About SFTP processing

SFTP is generally used for remote file access rather than file transfer. SFTP dialogue is composed of client requests and server responses (the server never takes the initiative in the dialog). In Gateway, the terms client and server are replaced with Initiator and Responder.

The SFTP syntax is based on system primitives that allow access to files and directories on UNIX or Windows.

Use SFTP commands to:

- Open, create and close files
- Read and write files
- Display directory paths

- Read and modify file attributes
- Delete and rename files in both client and server mode (RDF only)
- Delete and create and rename directories (RDF only)
- Read and create symbolic links
- Restart transfers
- Transfer multiple files (MPUT and MGET)

Note: Gateway only supports those SFTP commands mentioned in this document.

Availability of protocol-specific commands

Gateway provides different sets of functions for client and server mode. On the server side, replies to client requests are dependent on environment compatibility. The table below summarizes the functions available for each mode.

Note: You cannot read or create symbolic links in the Gateway implementation of SFTP.

Function	Client and Server	Server Only – in Real File Directories (RFD)	Client Only
Open file	YES	-	-
Create file	YES	-	-
Close file	YES	-	-
Read file	YES	-	-
Write file	YES	-	-
Display directory path	YES	-	-
Read file attributes	YES	-	-
Modify file attributes (In Real File Directories (RFD) only)	YES	-	-
Delete file	-	YES	YES
Rename file	-	YES	YES Files can only be renamed in a post-transfer command.

Function	Client and Server	Server Only – in Real File Directories (RFD)	Client Only
Delete directory	-	YES	-
Create directory	-	YES	-

SFTP extensions

SFTP is customized in Gateway via extensions to the protocol, negotiated during the SFTP initialization phase. They are used only between Gateway Sites for transfer verification and data code conversion.

Extended requests contain a string field in the format `name@domain`, where the domain is the DNS of the organization defining the extension. All extensions provided by Axway are therefore in the format `name@axway.com`

Using this syntax, you can add extensions in the file attributes to:

- Indicate whether or not the Remote Site is a Gateway
- Introduce external functionality with exits

Authentication

Authentication is the process of verifying a claimed identity. The most widely used authentication mechanisms are:

- **Password request:** Associated with user login requests: When the user sends a login name, the password request is displayed.
- **Proof request:** Associated with asymmetrical encryption mechanisms. The user sends the public key (or any information relevant enough to get the public key, such as a certificate), and then receives a random encrypted message. To decrypt the message, the user must own the unique private key associated with the public key. This means that if the user can decrypt the message, the users identity is authenticated.

When using TLS and SSL protocols, the client encrypts the symmetrical key with the public key for the server. If the server can decrypt the key, the client is as claimed. Otherwise, dialogue is not possible and the connection is closed. When requested, the server uses a signature process to authenticate the client. See the *Security Guide* > [SSL and TLS protocols](#) for further information.

Code conversion

SFTP does not transform data during a transfer. However, you can use Gateway to send or receive data in a data code other than the server default.

Gateway code conversion allows you to transform data, particularly carriage returns, when using SFTP.

Example: Transforming a text file from UNIX format to Windows, and vice versa

To implement this type of transformation, you must create a Transfer Request in text mode and ensure that the SFTP Transfer Request file type matches that on the remote server.

- Set the Application field **Data code** (`-data_code`) to text mode (ASCII).
- Set the Transfer Request field **Newline convention** (`-newline_convention`) to the file type of the remote server:
 - Windows (or W)
 - UNIX (or U)

If you use this option regularly, you can also set the Remote Site field **Newline convention** (`-newline_convention`) to the remote server default file type:

- Windows (or W)
- UNIX (or U)

Indicating end of transfer

SFTP considers a transfer ended when you close the file in question. There is no particular command to indicate that a transfer is terminated or to validate a transaction.

Since the client does not send an ABORT type primitive, it is also difficult to detect the interruption of a transfer. Only when the client closes the dedicated SSH channel is the transfer deemed terminated.

The table below summarizes how transfer status is determined.

Transfer direction	Description
Client to server	The server has no information about the file that it will receive. The client decides: • How to send the file, for example, by blocks of data (sequenced or random), or without sending blank characters (that the server will complete automatically) • When the transfer is ended, without advising the server
Server to client	The server can consider that the transfer is invalid if the client has not read the entire file

Transfer direction	Description
Both	A transfer is considered ended when: • The client closes the SSH channel • A second file is opened before the end of the current transfer

Resuming interrupted transfers

SFTP does not have an explicit command to indicate that a file access request refers to an interrupted transfer. Resuming a file transfer is only possible because the read and write requests include an offset and a length in their parameters. By selecting packet size and defining the start location, an SFTP client can read or write:

- Parts of a file in any order
- Only part of a file, suspend the transfer and resume it later

You can use the following methods to resume a transfer:

- Extended syntax with file names to link an FXP_OPEN request to a suspended transfer.
- The VFD syntax (local identifier with file name) also allows you to link an open file request with a transfer.
- In monitored RFD mode, interpretation of file open options allows you to identify the Transfer Request type either as the resumption of a suspended Transfer or the creation of a new one.
- In RFD non-monitored mode, there is no transfer tracking.

Transfer phases

SFTP comprises a set of commands that allow you to manipulate files and directories. An SFTP client can read or write parts of a file in any order, and suspend and resume transfers.

The phases of a transfer (in an error-free scenario) are:

Transfer phases	
Initialization phase	Exchange of version number and protocol extensions (All)
	1 Client sends SSH_FXP_INIT to the server.
	2 Server sends SSH_FXP_VERSION to the client.

Transfer phases		
Transfer phase (Core)	3	If the client requests to read from or write to a file, it sends the open command (SSH_FXP_OPEN).
	4	The client receives the handle attribution (SSH_FXP_HANDLE).
Transfer phase (Case 1)	Client to server file transfer complete server file path known The write (SSH_FXP_WRITE) and status (SSH_FXP_STATUS) commands are repeated until the file is written.	
Transfer phase (Case 2)	Server to client file transfer - complete server file path known 5 The read (SSH_FXP_READ) and status (SSH_FXP_STATUS) commands are repeated until the file is read. The client can also search for the attributes of the remote file.	
Transfer phase (Case 3)	Directory path request If the client requests to read the directory path, the name and the attributes of any file or directory present in the open directory can be retrieved. The client can also request the attributes of the remote files in the directories.	
	5	The read (SSH_FXP_READ) and status (SSH_FXP_NAME) commands are repeated until the directory is read and directory file names are all listed.

[Axway Gateway specific SFTP commands](#)

[Configuring SFTP](#)

[SFTP security](#) using SSH

[Submitting an SFTP Transfer Request](#)

[Transferring multiple files with SFTP](#) (MPUT and MGET)

Axway Gateway: Protocols

Gateway-specific SFTP commands

[Recognizing a Gateway partner](#)

[Using an Application](#)

[Identifying a transfer](#)

[File filter and list query](#)

[Pre-transfer and post-transfer commands](#)

Recognizing a Gateway partner

Gateway extensions to standard SFTP behavior can only be used between Gateway entities. To identify the two parties, a Gateway client sends the following extensions during the initialization phase in the format described in [SFTP extensions](#).

- "SSH_FXP_OPEN@axway.com"
- "SSH_FXP_OPENDIR@axway.com"

A server that recognizes all or part of these primitives returns the parameters that it recognizes in its reply.

For each of these requests, the data field that accompanies them indicates the parameters that you can enter. For example, the field **name&appli&dest&org&snd_msg**, provided as a complement to *SSH_FXP_OPEN@axway.com*, indicates the set of parameters that the client can supply to identify a transfer.

Similarly, the data fields accompanying the server responses contain the parameters recognized by the server for information purposes.

Parameters that are not recognized by the extended syntax are ignored.

Using an Application

There are two methods for a server to find the Application to use for a file transfer:

- Set the Application name as an open file parameter, as described in [About Applications](#)
- Use an FTP Application exit

If you do not select either of these two methods, Gateway uses the *FTP_B* Application. This is the standard Application to use with SFTP because it does not apply any data code conversion or transformation. SFTP always transfers files in binary mode.

Identifying a transfer

Remote file access is performed with the command **FXP_OPEN**. The extended file name syntax illustrated in the example below is similar to the syntax used in FTP/HTTP Transfer Requests.

The table below describes the elements that Gateway uses as parameters for **FXP_OPEN**.

Mode	Client to server transfer	Server to client transfer
RFD (monitored or not)	Names as they are received on the system. Gateway applies the options (read, write, append, creat, trunc, excl) and takes the VFD rights for the user into account.	System file names are kept and are visible if this client has authorized access.
VFD	The server can modify file names when it receives files, taking into account the local identifier and file name.	Files are made available to the client with virtual (logical) names. By default, the file name presented to the client is in the format: "s">+".">+<filename> or <i>file name only</i> (depending on VFD file name display option) However, it is possible to retain the protocol file name.

Example

```
name=dir1/file1&org=site1&dest=site0
```

This syntax allows you to set additional parameters required for transfers between two Gateway entities:

- **name:** file name (the identifier of the connected user must not appear in the file names)
- **appli:** application name
- **dest:** destination
- **org:** originator
- **snd_msg:** accompanying messages

The slash ("/") is used as a directory separator, the period "." identifies the current directory and two dots ".." the parent directory. The client can request file names that it can see in the directory list.

File filter and list query

You can request the transfer of a list of files in a directory. The result of the operation is stored in a file that can be used for subsequent requests.

Gateway allows you to filter a directory listing. If both parties are Gateway systems, the Initiator of the request can filter file names using the extension **SSH_FXP_OPENDIR**.

A file filter is a string that contains **field=value** pairs, separated by the "&" character. This filter replaces the directory name in transfer commands.

The table below shows the accepted field values.

Field	Accepted values
name	Can be a regular expression that matches paths and protocol file names, or a transfer identifier format (<xfer_ident>.<proto_filename>)
date	The exact moment of the transfer, given in the following date format: YYYYMMDDhhmmss • Date fields (year, month, day of month) can be separated by a "/" character. • Time fields (hours, minutes, and seconds) can be separated by ":" characters. • Time can be provided without a date. The current day is then considered as the default date.
direction	Accepted values are IN, OUT, or BOTH. It is not case-sensitive.
endbefore or end_before	All transfers requested or performed later than the given date are not selected. The value is of date format.
startafter or start_after	All transfers requested or performed earlier than the given date are not selected. The value is of date format.
origin or org	Originator Site alias
dest or destination	Destination Site alias
state	List of states in Gateway format. Each selected transfer has one of the given states.
localident or local_ident	Long integer
xferident or xfer_ident	Transfer identifier

Example

The following command lists files on the server SITE1, in the `sftp/dir1` directory, dated between 03/03/2006 12:13:14 and 03/03/2007 12:13:14.

```
peltrans -oa SITE1
```

```
-dir_name "name=/sftp/dir1/&start_after=20060303121314&end_
before=20070303121314"

-type DIR -dir I -ap FTP_B
```

List query reply format

In a standard request for a directory listing, the information returned for each entry is the:

- short file name
- long file name
- file attributes

The long file name is a field used for presentation purposes only.

Example

```
drwxrwxrwx 9 LSFTP RSFTP 12345 2006 Mar 25 14:29 file1
```

where:

- *drwxrwxrwx* = rights
- *9* = Mailbox status (translated into Axway Transfer InterPel MVS format or '-' in RFD)
- *LSFTP* = originating site alias or '—NA—' in RFD
- *RSFTP* = destination site alias or '—NA—' in RFD
- *12345* = size in KB
- *2006 Mar 25 14:29* = last modification date
- *file1* = file name

Pre-transfer and post-transfer commands

With the SFTP protocol you can send commands or information before and after a transfer using the **SITE** and **QUOTE** commands.

SITE command

SITE commands are used to send the SFTP protocol extensions. They must be negotiated during the SFTP initialization phase.

Example

```
SITE command data
```

This **SITE** command generates the transmission of an **SSH_FXP_EXTENDED** type with the parameters `command@axway.com /data.`

QUOTE command

QUOTE commands allow you to send standard commands, such as renaming a file after a transfer. The actions that are available with the **QUOTE** command include:

Action	Command syntax	SFTP Message
Delete a file	REM <i>file_name</i>	SSH_FXP_REMOVE
Rename/Move a file	RENAME <i>file1 file2</i>	SSH_FXP_RENAME
Create a directory	MKDIR <i>dir_name</i>	SSH_FXP_MKDIR
Delete a directory	RMDIR <i>dir_name</i>	SSH_FXP_RMDIR
Set remote pathname rights	CHMOD [N]NNN <i>pathname</i> where 0<=N<=7	SSH_FXP_SETSTAT

SFTP servers do not enable you to change the current directory. This can only be done on the client side.

SITE and **QUOTE** commands are used for pre-transfer processing. To indicate that a command must be processed *after* the transfer, you must use:

- **AE_QUOTE**
- **AC_QUOTE**
- **AE_SITE**
- **AC_SITE**

Use the **AE_** prefix when the transfer status must remain *ENDED* if the command fails, and the **AC_** prefix if the status is set to *CANCELED* if an error occurs.

Note: The post-transfer command will not be processed if the transfer fails (status is not *ENDED*).

Example

```
AC_QUOTE RENAME TOTO1.TMP TOTO1.OK
```

This command renames the file `TOTO1.TMP` to `TOTO1.OK` after the file transfer has ended (when the status is *ENDED*).

Calling a script

You can place **SITE** and **QUOTE** commands in a script and call it using the following syntax:

```
peladm update_site -snd_msg "SCRIPT <script_site>"
```

```
peltrans -ftp_command "SCRIPT <script_xfer>"
```

In a script file, you must write each command on a separate line or use a semi-colon (;) to separate commands.

Example script

```
AC_QUOTE RENAME old_name new_name
```

```
AC_QUOTE REM filename
```

Note: Gateway does not permit calling a script within a script.

Axway Gateway: Protocols

Configuring SFTP

[About SFTP configuration](#)

[Defining a Remote Site \(-remote_agent\)](#)

[Creating a Transfer Request](#)

About SFTP configuration

To send and receive transfers via SFTP and perform processing on these transfers, you must configure the following elements in Gateway:

- Remote Site
- Network Profile
- SSH Profile
- Transfer Request

You can use the following default objects supplied in Gateway.

- Local Site
- Application

Configuration of the SFTP Remote Site and Transfer Request are described in this section. For more information on creating a Network profile and an SSH profile, see the [Security guide](#).

Defining a Remote Site (-remote_agent)

The Remote Site represents an SFTP client or server. You must define at least one Remote Site for each remote SFTP partner.

The SFTP Remote Site can have the following roles:

- Responder/Receiver
- Responder/Sender
- Initiator/Receiver
- Initiator/Sender

The following table summarizes the elements you define for Remote Responder and Remote Initiator Sites.

Element	Remote Responder Site	Remote Initiator Site
Local SSH profile	YES	-
User name and password	YES Send, if requested by the Remote Server	YES Check, if requested by the Remote Client
Local private key alias, your certificate alias, or both, to authenticate your Gateway to the Server	YES	-
Key or certificate parameters	YES Use to authenticate the server	YES Use to authenticate the client
Default location of files to be sent or received	YES	YES

Refer to [Remote Site: SFTP tab](#) for information on the specific SFTP parameters to use.

Creating a Transfer Request

To send or receive a file you must also create a Transfer Request, either via the GUI or using the `peltrans` command. The syntax is:

```
peltrans <argument> value <argument> value <argument> value
```

Command arguments are parameters, each of which is followed by the corresponding value. Parameters are prefixed by the "-" character, and most have abbreviated names.

The `peltrans` parameters specifically relevant to SFTP are given in the following table.

SFTP relevant Transfer Request parameters

Parameter	Default	Use this parameter to:
<code>-type (-ty)</code>	TRANS DIR	Request and format a list of available transfers
<code>-dir_type (-dt)</code>	LONG	
<code>-dir_name (-dn)</code>		Define the location of a remote file
<code>-ftp_command (-ftpc)</code>		Send commands to remote server using <code>QUOTE</code> or <code>SITE</code> command
<code>-newline_convention</code>	(no default)	Set the end-of-line code, depending on the remote system type. This parameter is valid for text file transfers in both send and receive mode. Although this parameter is also available in the Remote Site definition, the Transfer Request parameter takes priority. Enter one of the following values: • U for UNIX • W for Windows • Blank for Undefined

Related topics[Working with Remote Sites](#)[Working with Remote Sites \(command line\)](#)[Executing an SFTP transfer](#)[Transfers: Parameters List](#)

SFTP security

SSH Software provides a secure logon and encrypts the data transfer connection.

SFTP connects to, and communicates exclusively with, the SSH node. No identification is transmitted in SFTP because the authentication of the connecting parties takes place between the SSH nodes.

The connection and data transfer are only secure if solely accredited clients (by certificate or password) are allowed to connect. Bear in mind that if you allow trivial passwords, security is limited.

See the *Security Guide* > *Security*: [SSH protocol](#) for further information (requires login).

Defending against hammer attacks

A hammer attack can occur when a user floods the server with requests until it can no longer respond to any other users.

Transfer quotas

Gateway provides Transfer quotas as a way of guarding against hammer attacks on the SFTP protocol. You can configure them either using the exit routine **ExitProtocolConnection** or the Gateway Site, CGate, and CGateGroup objects.

Gateway provides four quota parameters for you to set transfer upload or download limits within a designated time period.

The following table describes transfer quota fields. By default, the fields are set to 0, which is equivalent to deactivating the quota restriction.

Transfer quota	Definition	Defense if exceeded
<code>max_upload_rate</code>	Upload quota in KB per minute.	All fields refuse the client the right to transfer in the appropriate direction (upload or download) until it returns within the quota limit.
<code>max_download_rate</code>	Download quota in KB per minute.	
<code>max_upload_requests_rate</code>	Maximum number of upload-type transfers that can be requested per minute.	
<code>max_download_requests_rate</code>	Maximum number of download-type transfers that can be requested per minute.	

Quota window size

The Gateway SFTP server manages client consumption of these transfer credits for all quotas in one sliding window. The window size is set in the configuration menu **Quota window size** field (specified in minutes). The default is 0, that is, all defenses are disabled irrespective of the quota limits already set.

Example

If, for example, the following elements are set:

- **Quota window size** = 10 minutes
- **max_upload_requests_rate** = 8 uploads per minute

Then the Gateway SFTP server verifies that the client does not exceed 80 uploads in any 10-minute interval. If the client sends a request that exceeds this limit, the server automatically activates the defenses, refusing all client new transfer requests. These defenses remain active for the number of minutes necessary for the volume of uploads over the preceding 10-minute period to return within the quota limit.

Monitoring User quotas

In the Gateway GUI, you can track user and quota consumption via the **Connected User record**. The interface displays:

- Number of uploads and downloads
- Volume of data transferred
- Percentage of **max_upload_rate** and **max_download_rate** quotas used
- Percentage of **max_upload_requests_rate** and **max_download_requests_rate** quotas used

Gateway also provides the online command `pelctl reset_connected_user` to:

- Reset recorded consumption to zero
- Delete the connection if it is no longer active

This command therefore serves as a mechanism to authorize a client to reconnect after being suspended.

Axway Gateway: Protocols

Submitting an SFTP Transfer Request

Defining the remote file location

In Initiator mode, the Transfer Request `dir_name` parameter defines the location of a remote file. This location represents the file destination for an outgoing transfer, or the path for an incoming transfer.

In UNIX and Windows, this parameter typically indicates a path. However, since a remote FTP server does not manage files with a hierarchical file system, or file separators, the FTP server can only concatenate `dir_name` and `file_name` providing no further analysis of the two strings.

Example

The following transfer command sends the file named `file` to directory `dir1`.

```
peltrans -mode I -direction O -dest_alias SFTP_SERVER -appli FTP_B  
        -file_component local_file -dir_name /dir1/ -file_name file
```

Axway Gateway: Protocols

Transferring multiple files with SFTP

[About transferring multiple files](#)

[Transferring multiple files](#)

[Displaying transfer details](#)

About transferring multiple files

Gateway's SFTP client can upload and download *multiple files* with a single Transfer Request.

This function is equivalent to the SFTP `MPUT` and `MGET` commands:

- `MPUT`: to upload multiple files from a local directory to a remote site
- `MGET`: to download multiple files from a remote site to a local directory

When you use this function, Gateway carries out the following functions:

1. Lists the files of the specified directory.
2. Parses the list.
3. Applies a filter to specified file attributes.
4. Selects the matching files.

5. Requests and attaches the individual file transfers ("children") to a multiple transfer ("master").
6. Uploads or downloads the matching files.

This function is compatible with UNIX and Windows platforms.

Transferring multiple files

Prerequisite: All files must be in the same directory.

Using the GUI

Submit a Transfer Request with the following parameters:

1. In the [General tab](#):
 - In **Type**, select **MULTI**
 - In **Direction**, select **Send** (MPUT or upload) or **Receive** (MGET or download)
2. In the [Attributes tab](#):
 - Complete the **Filter** field to specify the files to send or receive. You can filter on file properties such as name, size, time, etc.
 - Select the **Hash type** to determine the file duplication behavior.
 - (optional) Select the **Model to use for child transfers**.

Using online commands

Use the **peltrans** online command to submit a Transfer Request with the following parameters:

- Set **-type (-ty)** to **MULTI**.
- Set **-direction (-dir)** to **O** (MPUT or upload) or **I** (MGET or download).
- Use **-multi_filter (-muf)** to specify the files to send or receive. You can filter on file properties such as name, size, time, etc.
- Use **-multi_hash_type (-muh)** to determine the file duplication behavior.
- Use **-multi_model (-mum) <Model name>** to select a Model to apply to the *child* transfers.

Example 1: Submitting an MGET using the peltrans online command

```
peltrans -type MULTI -mode I -dir I -oa RS16_STD -da LOCAL -ap FTP_
A
      -dn public/available_things/
```

where:

dir_name (-dn) is the name of the directory on the remote computer.

Example 2: Submitting an MPUT using the peltrans online command

```
peltrans -type MULTI -da RS16_STD -oa LOCAL -mode I -dir O -ap FTP_
A
      -fc \Temp\doc\Rfc\HTTP -dn public/received_things/
```

where:

`file_component (-fc)` indicates the name of the local directory that contains the files to send. This parameter is normally used for the name of the file to send.

`dir_name (-dn)` is the name of the directory on the remote computer.

Displaying transfer details

Use the `peldsp display_trans` command to display the details of a Transfer. When you use this command on a master transfer id, you get the following additional lines:

```
x_multi_state='processed all ok'
x_multi_hash_type='1'
x_multi_model=''
x_multi_filter='name=*.h'
x_multi_nb_total='90'
x_multi_nb_ok='90'
```

For `x_multi_state`, there are three possible states:

State	Message displayed	Description
SUCCESS	processed all ok	All child transfers have been transferred successfully.
PARTIAL_SUCCESS	processed partial ok	Error on at least one child transfer.
ERROR	processed all ko	Errors on all child transfers.

This information is also displayed in the GUI.

[Working with File Transfer Requests](#)

[Working with Transfers \(command line\)](#)

[Transfers: Parameters List](#)

SMTP/POP3

SMTP/POP3 Introduction

SMTP and POP3 message transfer protocols are integrated in Gateway as file transfer processes. You can use Gateway to:

- Transfer email
- Send files as email attachments
- Send user messages in an email format
- Receive email-formatted files [s29]

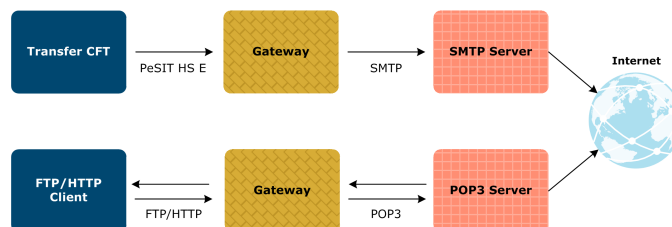
SMTP and POP3 are implemented in Gateway exclusively in client mode. This means that sending and receiving email occurs in a single direction for each protocol. In SMTP, the client (Initiator of the connection) can only *send* email. In POP3, the client can only *receive* email.

Gateway does not include SMTP and POP3 server functions. **You must use Gateway in conjunction with both an SMTP and POP3 server.**

Examples of SMTP/POP3 email processing

Email Transfer

Gateway can relay files received in PeSIT HS E to an electronic mailbox, as illustrated below:



Email Distribution

Gateway can make files received in an electronic mailbox available to HTTP and FTP clients.

Using SMTP in Gateway

SMTP is used to send email when Gateway is installed behind an SMTP server.

You can send the following types of email with Gateway:

- File transferred as an attachment (a maximum of one attached file is allowed) with or without a user message
- acknowledgment (of a transfer received in POP3)
- File already formatted as an email, unaltered by Gateway. This type of email "black box" can contain several attachments.
- User message (as in PeSIT)

Axway Gateway: Protocols

Configuring the SMTP environment

[About SMTP configuration](#)

[Defining a Local Site \(-local_agent\)](#)

[Defining a Remote Site \(-remote_agent\)](#)

[Defining an Application](#)

[Configuring SMTP defaults](#)

[Requesting an acknowledgment](#)

[Transfer identifiers](#)

[Limitations for SMTP parameters](#)

About SMTP configuration

Before submitting a file transfer, you must configure the SMTP environment by defining the following Gateway objects to be used, if they do not already exist:

- **Local Site**

On installation, you must create a Local Site corresponding to the `local_site_alias` and `local_site_proto_ident` parameters in the configuration file `<Gateway installation directory>/run_time/etc/conffile`. This is the default Local Site for all protocols supplied by the product, including SMTP. You can create a specific SMTP Local Site if necessary.

- **Remote Site**

Define communication attributes (such as SMTP server address, login and password) in Remote Site objects. The Remote Site represents an SMTP server to contact. At least one Remote Site must be defined for each remote SMTP server and user that Gateway uses to send email. The SMTP Remote Site is always Responder/Receiver.

- **Application**

The default Application is MAIL.

You can use either the GUI or the peladm online command (as in the examples used here) to create these objects.

Defining a Local Site (-local_agent)

The Local Site is the object that represents Gateway when connecting to an SMTP server. It provides the:

- Client identifier to the server via the EHLO command (or HELO if the server does not recognize extensions)
- Return email address to use by default if no address is provided when a transfer is submitted from this Site (From)

Defining a Remote Site (-remote_agent)

The Remote Site is used to:

- Connect to the server (Network address of the SMTP server)
- Establish a secure session with TLS
- Select the method to use for the identification phase with the server, and information relative to this method (user name and password for the login method)
- Obtain the destination email address to use by default if it was not specified when the transfer was submitted to this Site

You can define SMTP Remote Sites by creating either:

- One Remote Site that represents the SMTP server and identify the email recipients in the Transfer Request
- Several SMTP Remote Sites that reference the same SMTP server but have different default recipient addresses (the **To** field)

Note: Remote Site templates are not used in SMTP, because Gateway is always the Initiator in this protocol (no incoming transfers).

For the specific SMTP parameters to complete, refer to [Remote Site: SMTP tab](#).

Defining an Application

An Application is a set of parameters describing the structure of a file in terms of local storage and transfer format. In addition, the Application associated with a transfer indicates to Gateway whether it must create an email for the transfer or not.

Configuring SMTP defaults

You can configure default values for SMTP parameters in the Gateway [configuration menu](#).

Requesting an acknowledgment

Gateway only sends an acknowledgment request if, in the Transfer Request, you set the **Ack option** (`acknowledgment_option`) parameter to **To ack** (`TO_ACK`).

There are two types of acknowledgment in SMTP:

- DSN – (Delivery Status Notification) network (or protocol) acknowledgment
- MDN – (Mail Disposition Notification) application acknowledgment

The following table shows the type of acknowledgment request that Gateway sends for specific file types.

File type	acknowledgment request applicable
Attach file (<code>attach_extract_file</code>) parameter is set to Yes	MDN and DSN
Formatted email	DSN only

SMTP acknowledgments have the following characteristics in Gateway:

- **Initiation:** The sender requests acknowledgments in a transfer
- **Interaction with POP3:** acknowledgment requests sent in SMTP result in acknowledgments received in POP3. Inversely, reception of an acknowledgment request received via POP3 results in an acknowledgment sent in SMTP.
- **Multiple recipients:** If an email is sent to several recipients, network and application acknowledgments can be returned for each.
- **acknowledgment status:** If you are sending an email to multiple recipients, and DSN is acknowledged in error for at least one of the recipients, then the global SMTP acknowledgment status for a DSN is *negative acknowledgment*. It is *undefined* in all other cases.
- **Errors:** In case of error, both the DSN and the MDN are equally valid. However, only a positive MDN acknowledgment is considered a certain acknowledgment.

Transfer identifiers

A DSN acknowledgment carries a transfer identifier (ENVID) created by the SMTP client and relayed by the intermediary SMTP servers. When the final SMTP server receives an email it creates the acknowledgment file containing the transfer identifier and returns it to the sender (**From**).

On creation of an SMTP transfer, the transfer identifier (`xfer_ident`) used will have the format `<from_user.date_creation.local_ident@from_domain>` where:

Parameter	Meaning
<code>from_user</code>	User portion of the return address (From)
<code>date_creation</code>	Whole number representing the transfer creation date
<code>local_ident</code>	Transfer number (local identifier)
<code>from_domain</code>	Domain portion of the return address (From)

If Gateway creates the email, the SMTP transfer identifier `xfer_ident` is:

- Equivalent to the MIME **Message-Id**
- Used as the ENVID parameter of the protocol acknowledgment request (DSN)

Limitations for SMTP parameters

The following table summarizes SMTP parameter restrictions:

Parameter	Limits
User name	64 characters
Domain	64 characters
Path	256 characters, including all separators and punctuation
Command line	512 characters, including the command itself and the end-of-line <CRLF> characters. This maximum size is increased depending on the extensions implemented by the server.
Response string	512 characters, including the end-of-line characters <CRLF>
Text line (DATA)	1000 characters, including end-of-line characters (excluding the full stop copied at the beginning of a line)
Recipients	100 per transaction
Data	The SMTP server must be able to accept at least 64 KB of data

[Working with Local Sites](#)

[Working with Local Sites \(command line\)](#)

[Working with Remote Sites](#)

[Remote Site: SMTP tab](#)

[Working with Remote Sites \(command line\)](#)

[Working with Applications](#)

[Working with Applications \(command line\)](#)

Axway Gateway: Protocols

Configuring security for SMTP

[Establishing a secure link with SSL/TLS](#)

[Defining an authentication method](#)

Establishing a secure link with SSL/TLS

You can configure Gateway to use SSL or TLS security for SMTP:

1. In the configuration menu, select **Connectivity > Internet protocols > MAIL > SMTP**.
2. Click **Advanced....**
3. Check **Use SSL/TLS**.
4. Click **OK**.

You can configure Gateway to use TLS in implicit and explicit mode for an SMTP connection. Although most SMTP servers are secured explicitly, some accept implicit TLS connections, and SMTP servers of a proxy mailer may impose TLS in implicit mode.

Defining an authentication method

The authentication methods supported by Gateway for the SMTP protocol (all of which are standard SASL mechanisms) are:

- LOGIN
- ANONYMOUS
- PLAIN

Axway Gateway: Protocols

SMTP/POP3: sending an email

[Creating an SMTP transfer](#)

[Sending an acknowledgement \(EERP\) message](#)

[Resuming a transfer](#)

[Supported SMTP extensions](#)

Creating an SMTP transfer

Gateway sends email-type transfers in the same way as file transfers via any of the other Gateway protocols. To send an email, you must create a Transfer Request. Refer to [Managing Transfers](#).

Managing recipient addresses

Recipients list

You can specify either a single recipient or a *list of recipients* with an email. This list comprises:

Recipient	Field
Direct	To
Copied	Cc
Blind copy	Bcc

Note: If you do not define a recipient, by default Gateway uses the email address of the Remote Site that represents the SMTP server. Recipients defined in the Transfer Request overwrite this default address. Where no recipient is defined in either the Site or the Transfer Request, the transfer is rejected and an error is logged.

Multiple email recipients

Unlike the Gateway *Diffusion List*, SMTP allows you to send the same message to several recipients while physically sending the email only once to the server. Only one Mailbox record is created to monitor the SMTP send operation. This record contains the list of recipients and the acknowledgement counters (positive/negative) in the overflow zone.

Return address

You can also specify the return address (**From**) in the transfer. In the absence of a return address, Gateway uses the email address of the Local Site associated with the Remote Site.

Note: If neither of these addresses is defined, the transfer is rejected and an error is logged.

Address field length

The transfer record length is limited to 128 characters for each field (To, Cc, Bcc). If the field is insufficient to contain the corresponding list, Gateway completes the field with asterisks (*) and the value is stored in the overflow zone in the Mailbox record.

Completing the Subject field

In addition to the email **Subject** field, you can specify additional details in the **Add Gateway parameters to subject** field.

When you complete the **Subject** field via online commands, enclose the subject value in quotation marks (").

Adding Gateway parameters to the subject

You can define and transmit specific Gateway parameters (**Application**, **Origin** and **Destination**) in the MIME header Subject field when transferring an unformatted email (**Attach file = Yes**).

Note: When Gateway sends a pre-formatted email (**Attach file = No**), it does not use the **Subject** parameter and so does not take into account the values set for either the **Subject** or the **Add Gateway parameters to subject** (-gxs) parameters.

When you send an unformatted email via a Gateway Transfer Request, you can add parameters in the **Subject** field either manually or automatically when Gateway creates the email or the user message. The table below indicates the values to set for each field.

Mode	Add Gateway parameters to subject (-gxs)	Subject
Manual	No	Subject: //XFB appli=appliName; ORG=orgId; DEST=destId
Automatic	Yes	Leave empty

Completing the Attach file field

You can set the value for the **Attach file** parameter in these Gateway objects (taken into account in the following order of precedence):

- Transfer Request
- Application associated with the Transfer Request
- Model used to deposit the Transfer Request
- Site

If the values of all these parameters are undefined, the default value will be that of the **Attach file** parameter of the Remote Site.

Sending a pre-formatted email (Attach file = No)

Setting the **Attach file** to *No* indicates that the file transmitted is already formatted as an email. This functionality allows Gateway to act as an SMTP connector, transferring an email from an SMTP client to an SMTP server.

When Gateway sends a pre-formatted email, it does not modify the email that it is sending. Consequently:

- Gateway does not use the **Subject** parameter. It therefore cannot add transfer attributes to the **Subject** field. The **Add Gateway parameters to subject** (`-gxs`) parameter is set to *No* or *Undefined*.
- You cannot request an application acknowledgement (MDN). The source application is responsible for sending the MDN. If the acknowledgement option is set to *TO_ACK*, only the SMTP protocol acknowledgement (DSN) is requested.
- You must specify the origin and destination addresses in the Transfer or Remote Site, since the deposited file is not parsed.
- Gateway does not add the fields **To**, **Cc** and **Bcc** to the email. However, the SMTP protocol will use these fields to define the destination mailboxes. Only the **To** field is mandatory.

Note: If the **Attach file / Extract file** parameter is set to *No*, all the parameters except **To:** and **From:** in the new Transfer Request become unavailable.

*Sending a file and a user message (**Attach file** = Yes)*

When you send a file that is not formatted as an email, Gateway:

- Creates the email (the user message)
- Attaches the file to the email

To send a file as an attachment to an email, set the **Attach file** attribute of the transfer to *Yes*.

Use the parameters in the following table to send a file by email (as a user message with an email attachment).

Parameter	Required value	Description
Transfer type (<code>-type</code>)	TRANS	
Attach file / Extract file (<code>-attach_file</code>)	yes	
Remote Site (<code>-remote_agent</code>)		Site used to connect to the SMTP server. If no Site is defined, the default SMTP Site is used. (Refer to Defining a Remote Site (-remote_agent) for more information.)
File (<code>-file_component</code>)		File to attach to the email
Application (<code>-appli</code>)		Application to use for reading this file
Protocol file name (<code>-file_name</code>)		Name to assign to the file that you send
Recipients (<code>-to, -cc, -bcc</code>)		If no recipient is defined on creation of the transfer, the electronic address of the Remote Site (representing the SMTP server) is used as the unique destination address

Originating address (-from)		If no address is defined when you create the transfer, the electronic address of the Local Site associated with the Remote Site is used.
Email subject (-mail_subject)		
application (-appli)		
(-generate_xfb_subject) destination (-dest) origin (-org)	Y	Additional transfer attributes to add to the email subject
Acknowledgement request (-acknowledgment_option)	=TO_ACK	Request for application (MDN) and protocol (DSN) acknowledgments
User message (-snd_msg)		

Sending a user message (MSG)

Use the same syntax and parameters as described in [Sending a file and a user message](#), modifying the following values:

- Set the transfer type (-type) to *MSG*
- Do not define a file (-file_component)

Sending an acknowledgment (EERP) message

An acknowledgment is an email sent to indicate whether a file was received successfully or with errors. To send an acknowledgment in SMTP, the acknowledgment must be linked to a received email (a POP3 transfer).

The acknowledgments you can send are application type (MDN) acknowledgments. Since Gateway is not an SMTP server, the last SMTP server that receives the email must manage the acknowledgments.

To send an acknowledgement via the GUI:

1. Identify the transfer that you want to acknowledge.
2. Right-click **Transfer** in the left pane and select **Ack a transfer** from the context menu.
3. Complete the **Transfer Id to ack** field with the Transfer identifier.

To send an acknowledgement using the `peltrans` online command, use the following parameters:

- Transfer type (-type): *EERP*
- POP3 transfer identifier to acknowledge (-reply_to)

- Acknowledgement type: positive (`-ack_option=NO`) or negative (`-ack_option=YES`)
- SMTP Site to use: destination alias (`-dest`)

For syntax details, refer to [Transfers: Parameters List](#).

When Gateway sends the transfer, the acknowledgement is created automatically and sent to the return acknowledgement address indicated on the selected transfer (the `-ack_to` field of the POP3 transfer).

Resuming a transfer

SMTP and POP3 do not support the resumption of interrupted transfers. Where an incomplete transfer is recommenced, Gateway restarts from the beginning of the file.

Supported SMTP extensions

The following table summarizes the SMTP extensions supported in Gateway:

Extensions	Syntax	Description	Supported by Gateway
DSN	RET= ENVID= ORCPT= NOTIFY=	Send a request for protocol acknowledgement	X
SIZE	SIZE=000	Pre-allocation	
STARTTLS	STARTTLS	TLS implementation	X
AUTH	AUTH (SASL)	Identification mechanisms	X
	CHECKPOINT	Resuming a transfer	
	DELIVERBY	Requesting a transfer within a specified time period	

Axway Gateway: Protocols

Using POP3 in Axway Gateway

[About POP3](#)

[Setting up Gateway to work with POP3](#)

[Retrieving and processing emails](#)

About POP3

The Gateway implementation of POP3, Post Office Protocol Version 3, is exclusively in client mode and conforms to RFC 1939, incorporating the optional commands.

You can use the POP3 client in Gateway to:

- Connect to a POP3 server
- Retrieve email from a POP3 server
- Process retrieved emails
- Return acknowledgments

To be compatible, the POP3 server must also conform to RFC 1939 and support the commands USER, PASS and TOP.

Setting up Gateway to work with POP3

To retrieve and process email from a POP3 server, you must first configure the POP3 environment by defining the following:

- Remote Site for each user account on the POP3 server, thus enabling Gateway to connect to the POP3 Site and retrieve email for each specified user.
- Local Site
- Application object

You can use the Local Site and Application defined by default in Gateway.

Refer to [Configuring the POP3 environment](#).

Retrieving and processing emails

When you activate the polling option, Gateway retrieves emails by automatically polling the POP3 server for each defined Remote Site. You do not need to request a transfer.

For each email that it retrieves, Gateway:

- Identifies the email type by analyzing the MIME header. The following types are recognized:

MIME content-type	Axway MFT XFER record type	Description
Multipart/* : with a single attached file	TYPE=XFER FILE_NAME= <file name>	File transfer
Multipart/alternative text/*	TYPE=MSG FILE_NAME= <application name>	Message transfer
Multipart/report	TYPE=EERP	acknowledgment
All other types	TYPE=XFER FILE_NAME=RECOVER DUPLICATE	Unknown email format

- Processes the email according to its identified type:
 - Receives the whole email as a single received file (XFER)
 - Acknowledges certain recipients of a transfer (EERP)
 - Extracts the message of the email (MSG)
 - Extracts the message and the attachment (XFER)

When you receive acknowledgment requests via POP3, you return the acknowledgment via SMTP.

Axway Gateway: Protocols

Configuring the POP3 environment

[About POP3 configuration](#)

[Connecting to POP3 server](#)

[Retrieving emails](#)

[POP3 specific user exits](#)

[Types of email recognized by Gateway](#)

About POP3 configuration

You must create the following Gateway objects if they do not already exist:

- Remote Site
- Application

You can use either the GUI or the `pe1adm` online command to create these objects.

Defining a Remote Site

The Remote Site represents the POP3 server to contact. You must define one Remote Site for each email account on the POP3 server.

Defining an Application

An Application is a set of parameters that describes the structure of a file in terms of local storage and transfer format.

In addition to defining the formats you can use to receive a file, the Application associated with a transfer indicates to Gateway if it must extract a file from the email or not.

Connecting to POP3 server

Automatic Polling

Gateway retrieves emails by polling the POP3 server at defined intervals. To activate this polling mechanism, set the Polling Options in the [Remote Site POP3 tab](#).

Dynamically managing a polling cycle

While Gateway is running, it polls the POP3 server at the interval that you set in the Remote Site (`polling_interval` parameter). If necessary you can suspend polling for server maintenance, troubleshooting, or diagnostics purposes.

To suspend or resume the polling activity on a Remote Site, use the `pelctl` commands:

Note: These commands do not affect transfers that are already initiated or in progress.

- To suspend polling, use `pelctl stop_site -a pop3_<site_name> -pm`
- To resume polling, use `pelctl start_site -a pop3_<site_name> -pm`

The change to the polling status is confirmed in the log file.

Retrieving emails

The principal elements of email retrieval in Gateway are:

- Authentication mechanism:
 - AUTO
 - BASIC
 - APOP [RFC1939]
 - CRAM
- Automatic and periodic polling of POP servers to retrieve email
- Capacity to suspend and resume an email retrieval session
- Automatic deletion of email from the server after a POP3 session

How Gateway retrieves email

Gateway examines the MIME header of an incoming email and, if identified, creates the Transfer Request.

For every type of email received, except acknowledgment requests (DSN/MDN), a Transfer Request (of the Initiator/Receiver type) is created for possible monitoring or routing.

Otherwise Gateway processes the retrieved email according to the email type identified from the MIME header.

Rejecting the Transfer Request

If there is an error in parsing the MIME header, or a duplicate email is detected during the retrieval process, Gateway rejects the Transfer Request (if already created) and marks it as *CANCELED*. The status of the email is determined by the value of the `ERROR_MAIL_DISPOSAL` parameter for the Site in question.

The table below describes how Gateway handles the *CANCELED* Transfer Request.

ERROR_MAIL_DISPOSAL value	Action
LEAVE	The email is ignored and left on the POP3 server
DELETE	The email is deleted from the POP3 server
RECOVER	A new Transfer Request is created with the <code>extract_file</code> parameter set to <i>No</i>. The email (header and body) is retrieved (still encoded) with the identifying parameters: • <code>PROTOCOL=POP3</code> • <code>ORIGINATOR=POP3 Site identifier</code> • <code>DESTINATION=login user identification</code> • <code>FILE_NAME=RECOVER DUPLICATE</code> • <code>FILE_TYPE=DATA</code> • <code>XFER_ID=XFER ident.local_ident</code>

extract_file parameter

If Gateway accepts the Transfer Request, it uses the `extract_file` parameter to determine how to process the email. The value of this parameter is read from the objects below in the following order of priority:

- Application: returned by the exit function **ExitPop3GetAppli**
- Remote Site: for the POP3 server in progress

POP3 specific user exits

Gateway supplies the specific exit routine [ExitPop3GetAppli\(\)](#) for the POP3 protocol.

This exit allows you to select an Application to apply to the current transfer. Input and output parameters are described in the following table:

Parameter	Name	Description
Input	xfer.proto.originator	Protocol originator
	xfer.proto.destination	Protocol destination
	xfer.proto.from	Email From field
	xfer.proto.to	Email To field
	xfer.proto.xfer_ident	Message_id
	xfer.sup.mail_subject	Email subject
Output	xfer.sup.application	Application to use
	return code = 1	No Application is selected, use the default defined in the Site
	return code = 0	Use the Application in the return parameter

Types of email recognized by Gateway

Parsing the contents of the MIME header of the email, Gateway can identify four kinds of email:

Email type	Description
XFER (extract_file=yes: single file attached)	Envelope for a file transfer: a multi-part email with an attachment. The body of the message (optional), up to 254 characters, is considered a transfer user message. The first attachment* is considered to be the transferred file.
MSG (extract_file=yes: no file attached)	User Message: a one-part email without an attachment. The body of the message is considered to be the transferred message.
XFER (attach_extract_file=no)	Formatted email
DNS/MDN	acknowledgment: Delivery Status Notification (DSN)/Mail Disposition Notification (MDN).

(*) Gateway does not accept emails with more than one attachment.

[About Site objects](#)

[Remote Site POP3 tab](#)

[About Applications](#)

[Internal user exits](#)

Axway Gateway: Protocols

Processing email after retrieval

Mapping the Mailbox fields

Gateway extracts parameters (**DEST**, **ORG**, **APPLI**) from the subject (//**XFB**) if **Attach file / Extract file** in the Application or the Remote Site object is set to Yes.

Identifying an email

The **message-id** field identifies the email. Each message identifier must be unique as Gateway rejects any email whose message identifier corresponds to that of an email already received.

If the **message-id** field is missing, Gateway generates a hash on the fields **From**, **To**, **Orig-date**, **Subject** and **Cc**, and uses this as the message-id.

Processing different email types

Gateway processes the various email formats as described in the following table:

Email type	Gateway treatment
XFER Single attached file	Processes the body of the message, as a user message that accompanies the transfer. The attachment is considered to be the file transferred. Gateway extracts, decodes and saves the attached file in a physical file, disregarding all other elements. Gateway decodes and saves only the content of the email. If a technical error occurs during processing, Gateway: • assigns <i>Canceled</i> status to the Transfer Record, if it is already created • ignores the email, but does not delete it from the POP3 server • processes the next email
Multiple attached files	Rejects
MSG No attached files	Extracts and decodes the body of the message. If the message is less than 254 characters in length, Gateway stores the message in the request itself. Otherwise, it saves the message in a physical file.
extract_file=no	Saves the entire email (header and the content), without decoding it. Gateway updates the request in question at the end of the transfer.

DNS/MDN	Since Gateway only supports POP3 and SMTP in client mode, the acknowledgement mechanism for these protocols is different from that used for other protocols. When Gateway sends an acknowledgement in SMTP, the notification is received in POP3. Gateway extracts each notification and updates the corresponding SMTP transfer records in the Mailbox.

Diagnostics and Troubleshooting SMTP/POP3

[About diagnostics and troubleshooting](#)

[Mailbox content](#)

[Protocol Trace](#)

[Sample POP3 commands](#)

About diagnostics and troubleshooting

During a file transfer you can record messages, exchanges between Gateway and a POP3 server, relating to the transfer activities in the log file.

Mailbox content

To monitor SMTP/POP3, the Mailbox stores the following information for each transfer:

Field	Comment
<Subject>	After removal of any specific Gateway parameters such as appli , dest and org
appli	These parameters are saved in the fields Application , Destination and Origin of the Mailbox record
dest	
org	
<From>	Taken from the MIME Header field
<To>, <Cc> and <Bcc>	If there is more than one email recipient, these lists are saved in the overflow zone of the Mailbox. The overflow blocks strings in the form: To: (options) "description" <address@email>, \0 Cc: \0 Bcc: \0 \0 For each recipient, the <i>options</i> field will contain the complementary information (such as acknowledgement status).

xfer_ident	Corresponds to the Message-ID field of the MIME header, and is used to detect duplication of received messages. If this field is not found in the header, a hash of the MIME header (Date, From, To, Cc, Bcc, Subject) replaces it.

Protocol Trace

Activate the protocol trace for more detailed diagnostics when an unexpected error occurs.

The protocol trace is an optional feature that you define for transfer activity for a given Site.

You can enable or disable trace printing for a Site object at either of the following stages by setting the Remote Site parameter `-trace_mode (-tm)` to Y (yes) or N (no):

- During Site creation
- On modifying attributes of an existing Site
- During Gateway processing (dynamically)

Enabling or disabling protocol tracing using this method is only valid for the current session. The update is lost when you stop Gateway.

Sample POP3 commands

Use the following online commands to manage and launch POP3 in Gateway:

- `peladm` to create and update POP3 Sites and Applications
- `pelctl` to control the polling status of a Site
- `peldsp status_site` to view the status of the polling activity of a Remote Site

Note: The `peltrans` command is not used for the POP3 protocol, which operates by polling the POP3 Sites.

To create an SMTP/POP3 loop (to send an email in SMTP and receive it in POP3), you must create a Remote Site to represent each server partner, in this case the SMTP server and the POP3 server. You only need to create a transfer in SMTP.

Example: Creating an SMTP/POP3 Loop

Create an SMTP Remote Site, as follows:

```
peladm create_site -a site_smtp -pr SMTP -aef N -smtp_auth_method
none

-l_id user1 -l_pswd user1 -ct tcpip -pi smtp_
attach_no
```

This command creates an SMTP Remote Site named **site_smtp** with the following parameters:

- **Attach file / Extract file** set to *No* (sends a formatted email)
- Authentication method: none (id to server with login and password)
- Protocol identifier: *smtp_attach_no*

Create a POP3 Remote Site, as follows:

```
peladm create_site -a site_pop3 -pr POP3 -aef N -pop3_auth_method
basic

-l_id user1 -l_pswd user1 -ct tcpip -pi site_pop3
-pm Y

-pollint 1 -dp /dir1/dir2/ -routing N
```

This command creates a POP3 Remote Site named **site_pop3** with the following parameters:

- Polling mode: enabled
- Polling interval: 1 minute
- Storage directory: */dir1/dir2/*
- Protocol identifier: *site_pop3*

Send an SMTP transfer, as follows:

```
peltrans -appli MAIL -fc /dir1/dir2/SMTP_POP3/file.txt -da site_smtp
-mode I

-oa GW1 -smtp_from user2@xxx.yy.zz -to user1@xxx.yy.zz
```

This Transfer Request sends an email from user2@xxx.yy.zz to user1@xxx.yy.zz.

Axway Gateway: Protocols

EDIINT

[EDIINT](#)

[General operating principles](#)

[Generic EDIINT transfer example](#)

EDIINT Introduction

What is EDI?

Electronic Data Interchange (EDI) is a set of protocols for performing data exchanges between business organizations. EDI provides the processes that enable companies to buy and sell material, and to trade information. Businesses can use EDI, for example, to transfer:

- Purchase orders
- Invoices
- Shipping notices
- Insurance claims
- Medical records
- Clinical data
- Retail reports
- CAD drawings

Historically, companies who use EDI have relied on Value Added Networks (VANs) to exchange information. As the use of a VAN implies per transaction or per volume billing rates, the deployment of EDI in a high-volume business-exchange environment can be very expensive. The increased development of large and cost-free Internet bandwidth has led to the application of Internet protocols and security standards to the transport of EDI data.

What is EDIINT?

Electric Data Interchange over the Internet (EDIINT) is a Working Group of the Internet Engineering Task Force (IETF). This working group is chartered by the IETF to create a set of secure protocols for conducting inter-business information exchanges using Internet protocols.

Businesses that exchange large volumes of documents with partners require solutions that reduce information transfer costs. EDIINT makes use of existing Internet technology to significantly reduce the costs of EDI document transfers between trading partners.

Although EDIINT specifications evolved from EDI, they are not payload specific. EDIINT specifications are transport specifications that you can use to transmit any data as long you use the appropriate MIME content type.

EDIINT exists primarily as a solution for the upgrade of legacy EDI networks from VAN-based technology to the Internet. A major consideration of the IETF in developing EDIINT was ensuring adequate security for exchanges over the Internet. The IETF has invented an acronym to describe the EDIINT security solution, PAIN:

- **P**: Privacy
- **A**: Authentication

- **I:** Integrity
- **N:** Non-repudiation of message reception

To provide these security features, EDIINT encapsulates user data in S/MIME envelopes that enable encryption and signed receipts. Note, that the use of EDIINT security features remains optional. It is possible to transmit user data without applying encryption and/or without requesting or receiving a signed receipt.

What is an AS?

In EDIINT terminology, **AS** stands for **A**pplicability **S**tatement. An AS is a set of recommendations that shows how to apply existing standards to facilitate secure reliable EDI transport over the Internet. The recommendations include a small number of additions to the existing standards, primarily to provide more precise coordination between the existing standards.

EDIINT has currently produced three sets of AS recommendations:

- **AS1:** The AS1 specifications describe how current Internet standards can be used to achieve document transfers as a mail attachment. AS1 is based on SMTP and POP3. It was the first of the Application Statements to be developed, and does not provide the same level of reliability as AS2 and AS3.
AS1 security features include signatures for transferred objects and object-based encryption only.
- **AS2:** The AS2 specifications describe how to package the information in an envelope and transfers the document via HTTP using an S/MIME standard.
For AS2, S/MIME provides both authentication and privacy. AS2 provides for an authenticated acknowledgment on reception of a message.
Security in AS2 includes signatures for transferred objects, and both session and object encryption.
- **AS3:** The AS3 specifications offers data transfers similar to AS1 and AS2, but performs these transfers using FTP.
Security in AS3 includes signatures for transferred objects, and both session and object encryption.

All EDIINT AS versions use TCP/IP as the network-level protocol.

General operating principles

Message characteristics

EDIINT makes the distinction between the file you transfer using EDIINT protocols and the package that envelops the transferred file. EDIINT encapsulates user data and returned receipts in S/MIME envelopes and assigns headers to enable the handling of the envelope by the intended receiver. The

total volume of transferred data, contents and headers combined, is called a *message*. EDIINT, in fact, is composed of transport protocols that are indifferent to the type of data that is transported.

Network security

EDIINT protocols enable you to transfer documents over the Internet. To avoid malicious attacks you require a network security strategy, such as the deployment of a firewall.

Certificates

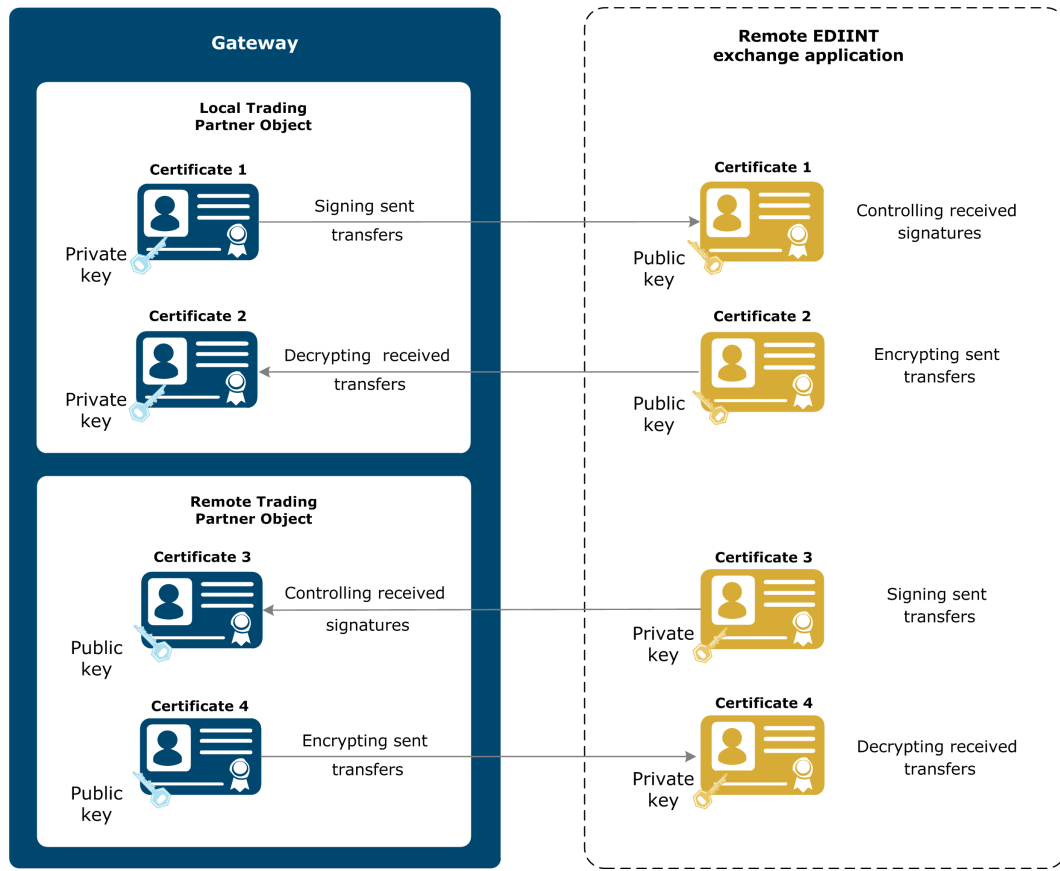
To use EDIINT you need to manage security certificates. You have the option between using self-signed certificates or using certificates offered by a Certificate Authority.

Before initializing an EDIINT exchange, the exchange partners must agree on the certificates they will use, and either exchange certificates or agree on a common PKI.

In Gateway, you have the option of using:

- No certificates
- A single certificate for signing and encryption operation (on each machine)
- Dual certificates: one for signing and another for encryption (on each machine)

The following diagram illustrates a dual certificate agreement between two exchange partners. In this case, a complete EDIINT exchange including transfer and receipt generation requires four certificates (with their associated public keys).



See the *Security Guide* > [Certificates and Keys](#) for further information (requires login).

For details on how to specify certificates and certificate types for an EDIINT transfer, refer to:

- [Working with Trading Partner objects](#)
- [Working with Trading Partner objects \(command line\)](#)

Transport layer protocols

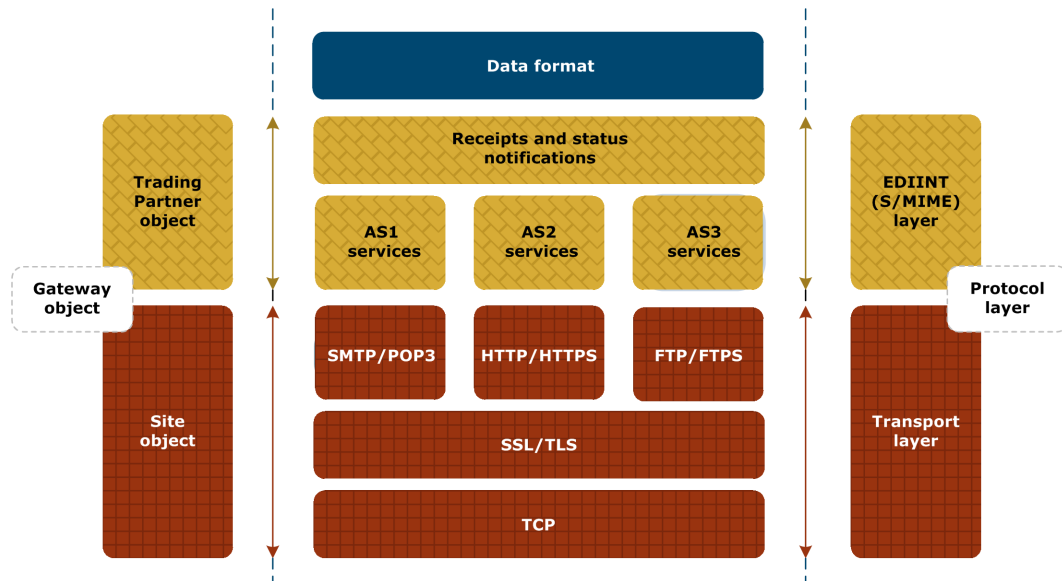
The transport-layer protocol you use with EDIINT depends on the AS version you are using:

- AS1: SMTP/POP3
- AS2: HTTP or HTTPS
- AS3: FTP or FTPS

Gateway includes native processes that handle SMTP/POP3, HTTP, HTTPS FTP and FTPS transfers. However, to execute Gateway transfers using EDIINT, you must activate support for dedicated EDIINT versions of these protocols in the Gateway configuration menu. Additionally, you must specify the EDIINT AS version in the related Site object, Trading Partner object and Transfer Request.

Managing service layers in Gateway

To specify the transport layer protocols for a Gateway EDIINT transfer, you create and manage [Site objects](#). In your Transfer Request you associate the Site object with a Trading Partner object. The Trading Partner object contains specifications for the S/MIME level handling of the EDIINT message. The following illustration shows the relationship of these two objects and the services they define.



Receipts

EDIINT uses S/MIME functions to return reception acknowledgments known as **receipts** to the entity from which a transfer originated. The sent message contains information specifying the type of receipt that is to be returned.

Receipt requests always come from the entity that sends the EDIINT message. The returned receipt can be positive or negative.

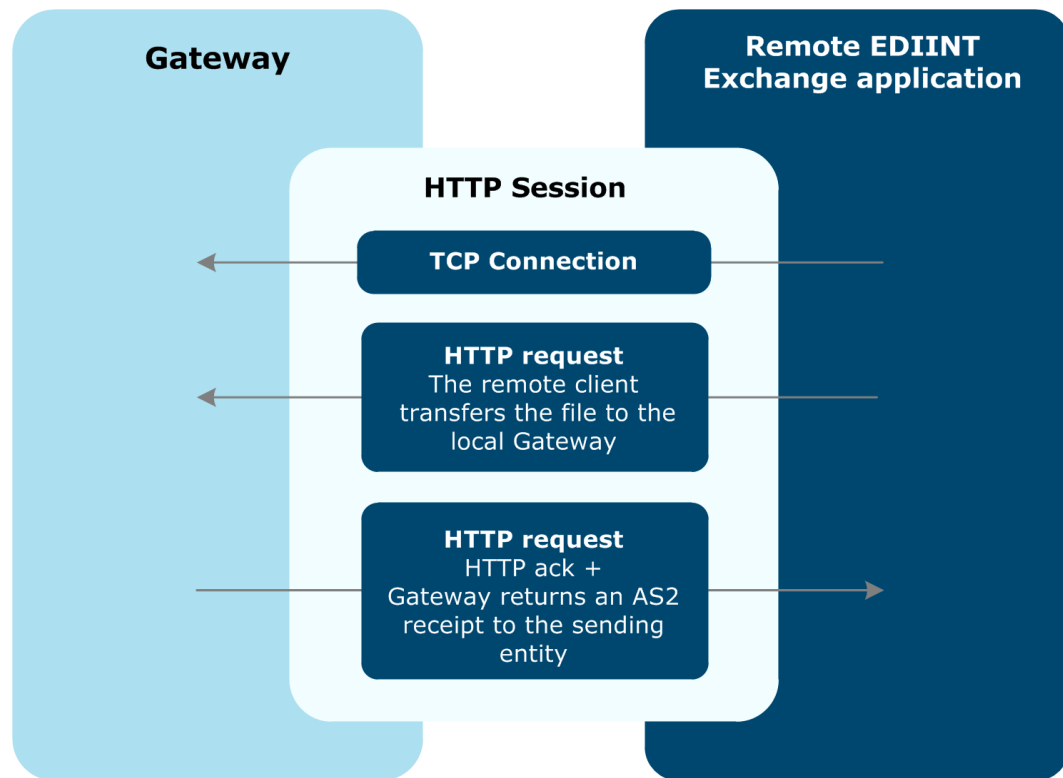
You can use the Gateway Trading Partner object to specify any of the following receipt types:

- **None:** No signature. This option excludes the possibility of later proof of reception
- **Signed**
- **Unsigned**

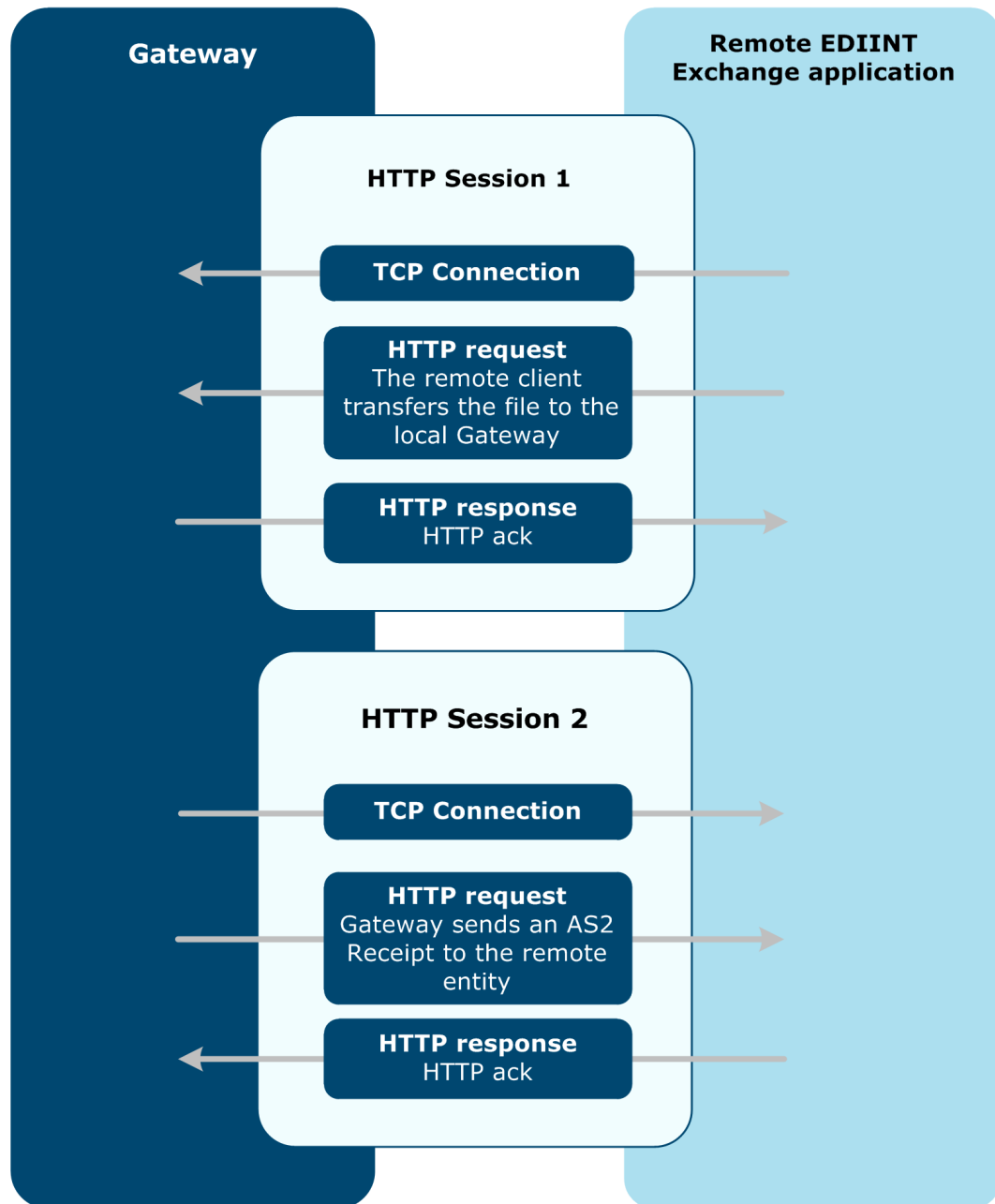
Synchronous and asynchronous receipt processes

Additionally, in the Trading Partner object, for AS2-type transfers you can choose between synchronous and asynchronous receipt processing:

- **Synchronous receipts:** The receiving partner returns a receipt immediately on file reception and unpacking in the same HTTP protocol session, using the standard HTTP response mechanism.



- Asynchronous receipts: The receiving partner generates a reception receipt in a protocol session independent of the original HTTP transfer session.



Managing receipts in Gateway

You manage receipt activation and receipt characteristics via the Gateway Trading Partner object.

Monitoring receipts in Gateway

Gateway provides monitoring features that enable you to follow the status of EDIINT receipts. The emission and reception of a receipt is itself a type of transfer. By selecting the transfer record in the Gateway Mailbox that corresponds to the transfer of the receipt, you can view details of the transfer, including status, signature, encryption and signing details.

Refer to [Viewing and managing Mailbox contents](#).

Encryption Algorithm

Gateway offers the following data encryption algorithms for EDIINT:

- **AES-128**
- **AES-256** (for AS2 only)
- **ARC4** (for AS2 only)
- **DES**
- **3DES**

In Gateway, you specify the encryption algorithm to use with the transfer, in the Trading Partner object that you associate with the transfer. Alternatively, you can choose not to encrypt the data.

Note: The partner in the transfer must imperatively support the same encryption algorithm. It is necessary to know before initiating the transfer, which algorithms the partner can use.

Managing encryption algorithms in Gateway

You manage encryption algorithms via the Gateway Trading Partner object.

Signing Algorithm

You can select to sign a file that is transferred via EDIINT, to reinforce the protection of the data integrity and security. The signature coupled with RSA key encryption enables the receiver to confirm that the message effectively came from the sender.

Gateway offers the choice of one of three signing algorithm options:

- **None:** no certificate (default)
- **SHA1 with RSA** (recommended by EDIINT)
- **MD5 with RSA**

In Gateway, you specify the signing algorithm to use for the transfer, in the Trading Partner object that you associate with the transfer.

Managing signing algorithms in Gateway

You manage signing algorithms via the Gateway Trading Partner object.

Compression

In EDIINT, compression reduces the size of a file or group of files in order to reduce transmission time.

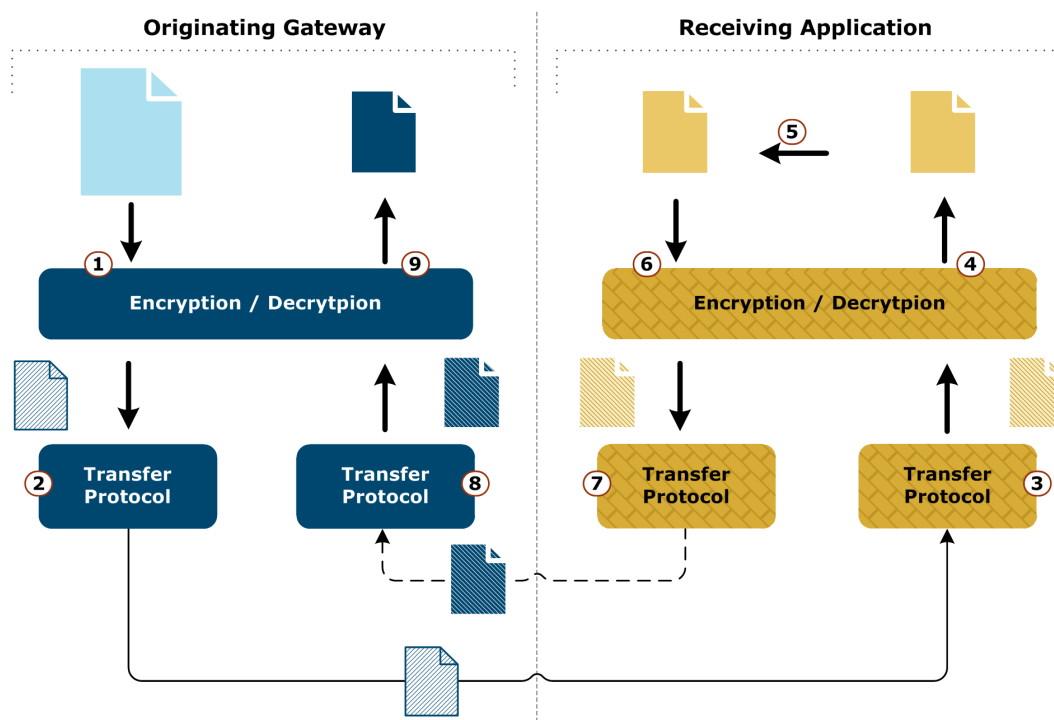
Gateway currently offers the ZLIB compression option, which you can activate in the Trading Partner object.

Managing compression in Gateway

You manage EDIINT compression via the Gateway Trading Partner object.

Generic EDIINT transfer example

The following example illustrates the sequence of events that occur in a typical EDIINT AS2 or AS3 transfer exchange.



The illustration shows the main steps of an EDIINT transfer:

1. The originating Gateway encodes and formats the data to be sent, constructing a MIME-format envelope. During this process, Gateway encrypts and signs the data using standardized S/MIME processes. The resulting S/MIME envelope and its content is an EDIINT data message.
2. The originating Gateway sends the secured envelope to the target business partner using one of the three following protocols:
 - SMTP/POP3 for AS1
 - HTTP or HTTPS for AS2
 - FTP or FTPS for AS3

3. The destination application receives the secured EDIINT data message. Before handling the message, the application checks to see that any *forcing* requirements (predetermined reception requirements) are fulfilled.
4. The destination application extracts data from the fields of the header and unpacks the data from the envelope, uncompressing and decrypting as necessary. These processes restore the data to its original format.
5. When the destination application has restored the data, if the originating Gateway has asked for a receipt, the receiving application constructs a receipt. This may be signed by the application that has received the EDIINT data message.
6. The destination application encrypts and formats the receipt.
7. The destination application sends the EDIINT receipt back to the originating Gateway via the protocol decided in the Trading Partner Agreement.
8. The Gateway that originated the EDIINT data message, receives the secured receipt. Before handling the return receipt, the application checks to see that any *forcing* requirements (predetermined reception requirements) are fulfilled.
9. Gateway extracts data from the fields of the header and unpacks the receipt from the envelope. If a signature was included in the envelope, Gateway decrypts the signature.

[Configuring and using EDIINT](#)

[Overview: Trading Partners](#)

[Transfer Examples: EDIINT AS2 transfer](#)

[Working with Trading Partner objects](#)

[Working with Trading Partner objects \(command line\)](#)

Axway Gateway: Protocols

Configuring and using EDIINT

[Configuring Gateway for EDIINT](#)

[Creating objects for EDIINT transfers](#)

[Using EDIINT on Gateway](#)

Configuring Gateway for EDIINT

Prerequisites

Before you can send and receive transfers via EDIINT and perform processing on these transfers, you need to:

- Confirm that you have a TCP/IP connection, and that your business exchange partners are connected to the network at the time you initiate or expect to receive a file transfer or acknowledgment
- Activate the appropriate supporting protocol processes in the [configuration menu](#)
- Ensure that you have access to the necessary Public Key Infrastructure (PKI). Each exchange partner must possess the public certificate for the remote and local partner as well as the associated private key. See *Security Guide* > [Certificates and keys](#) for further information (requires login).
- With each business exchange partner, you establish an EDI exchange contract known as a *Trading Partner Agreement*. In a Trading Partner Agreement you mutually establish:
 - your EDI identifiers, unique names for every exchange entity
 - the services you want to apply to your exchanges: signing, encryption, compression, and receipt management
 - target file directory locations for your exchanges
 - signing and encryption algorithms
 - for AS2, synchronization or non-synchronization of receipt emissions

When you configure your respective EDIINT compliant applications, you must scrupulously respect the terms of your Trading Partner Agreement.

Required objects

You require a different set of objects depending on whether you are sending or receiving a transfer. A complete EDIINT exchange normally comprises a file transfer phase and a receipt transfer phase. Between these two phases, the initiator and responder machines trade roles. For this reason, whether Gateway is acting as initiator or responder in an EDIINT exchange, you require the same set of objects to complete the file transfer/receipt transfer cycle.

Sending transfers

To successfully initiate an EDIINT transfer you require as a minimum the following Gateway objects:

- [Remote Site object](#)
to specify transport layer characteristics of the remote machine to which you are sending a file or a receipt
- [Local Trading Partner object](#)
to specify:
 - local identification
 - the signing certificate and signing algorithm on emission
 - optionally, the encryption certificate necessary for decrypting the returned receipt
- [Remote Trading Partner object](#)
to specify the remote compression, signature, encryption and encapsulation processes for the exchange with your business partner.

Receiving transfers

To successfully receive an EDIINT transfer from a remote business partner, you require as a minimum the following Gateway objects:

- [Remote Site object](#)
to specify transport layer characteristics of the remote machine from which you are receiving the file
- [Local Trading Partner object](#)
to specify:
 - local identification
 - the signing certificate and signing algorithm for controlling the message on reception
 - optionally, the encryption certificate necessary for encrypting the return receipt
- [Remote Trading Partner object](#)
to specify the remote compression, signature, encryption and encapsulation processes for exchanges with your business partner.
- [CGateGroup and CGate objects](#)
to control the remote connection to your machine by your business partner for transfer of the file or receipt

Additional required objects

The above objects take into account only the basic objects necessary for the transfer. Depending on your transfer type and business context, you will require additional Gateway objects to process incoming files. These objects can include, for example, VFD or RFD objects, SSL/TLS security objects, Decision Rule objects, and so on.

Creating objects for EDIINT transfers

As described in the previous section, you must create a certain number of Gateway objects before you can use Gateway to send and receive files via EDIINT:

- [Remote Site objects](#)
- [Local Trading Partner objects](#)
- [Remote Trading Partner objects](#)
- [CGate objects](#)

Creating a Remote Site object for EDIINT transfers

The Remote Site object provides Gateway with information about the remote machine to which you are sending your EDIINT messages and receipts.

You can create Remote Site objects in Gateway either via the GUI or command line.

Using command line

To create a Remote Site object via command line, use the [peladm create site](#) command. The parameter values depend on your transfer requirements.

The following `peladm create_site` parameters specifically concern EDIINT transfers.

Parameter	EDIINT-related value
<code>-protocol (-pr)</code>	Specify one of the following protocols: • AS1_POP3 • AS1_SMTP • AS2 • AS3

Using the GUI

To create a Remote Site object via the GUI:

1. In the left pane of the GUI main window, click to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node, then select **New** from the context menu to display the *New Remote Site* window.
Define the characteristics of the Remote Site object by entering values in the fields. Refer to [Working with Remote Sites](#).

The following Remote Site parameters specifically concern EDIINT transfers.

Tab	Field	Value
General	Protocol	Select the appropriate protocol: • AS1_POP3 • AS1_SMTP • AS2 • AS3

Creating a Local Trading Partner object for EDIINT transfers

The Local Trading Partner object provides Gateway with information about the EDIINT compression, signature, encryption and encapsulation operations that you use to transfer your message. It also enables you to specify how to handle the acknowledgments returned by remote partners.

You can create Local Trading Partner objects in Gateway either via the GUI or command line.

Using command line

To create a Local Trading Partner object via command line, use the [peladm create tradepart](#) command.

Using the GUI

To create a Local Trading Partner object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.
Gateway displays a context menu.
3. Select **New**.
Gateway displays a context menu.
4. Select the type of partner:
 - **AS1 partner**
 - **AS2 partner**
 - **AS3 partner**

Gateway displays the *Trading Partner* window.

In this window, you define the characteristics of the Local Trading Partner object by entering values in the appropriate fields. Refer to [Working with Trading Partner objects](#).

Creating a Remote Trading Partner object for EDIINT transfers

The Remote Trading Partner object provides Gateway with information about the compression, signature, encryption and encapsulation operations defined by the EDIINT protocol.

You can create Remote Trading Partner objects in Gateway either via the GUI or command line.

Using command line

To create a Remote Trading Partner object via command line, use the [peladm create tradepart](#) command.

Using the GUI

To create a Remote Trading Partner object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.
Gateway displays a context menu.
3. Select **New**.
Gateway displays a context menu.

4. Select the type of partner:

- **AS1 partner**
- **AS2 partner**
- **AS3 partner**

Gateway displays the *Trading Partner* window.

In this window, you define the characteristics of the Remote Trading Partner object by entering values in the appropriate fields. Refer to [Working with Trading Partner objects](#).

Creating a CGateGroup object for EDIINT transfers

To receive EDIINT messages or message receipts from a business partner on your local machine, you control remote connection attempts to your machine via the CGateGroup object and CGate object.

You can create CGateGroup and CGate objects in Gateway either via the GUI or command line.

Using command line

To create a CGateGroup object via command line, use the [peladm create cgategroup](#) command.

To enable the reception of a connection based on an EDIINT protocol, in the `protocol` attribute of this command, enter one of the following values to specify the EDIINT protocol format used by your remote partner for the acknowledgment transfer:

- AS2
- AS3

Using the GUI

To create a CGateGroup object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup...**

Gateway displays the *New CGateGroup* window. Complete this window to define the characteristics of your new CGateGroup.

Refer to [Working with CGates and CGateGroups](#).

In particular, when you create the CGateGroup, in the **Protocol** field, specify the protocol that is used by the connecting partner:

- **AS2**
- **AS3**

Next, create a CGate as described in the following section.

Creating a CGate object for EDIINT transfers

To specify the login name and password requirements for AS1 or AS2 connections to your local machine, create a CGate in the CGateGroup you created in the previous section.

You can create CGate objects in Gateway either via the GUI or command line.

Using command line

To create a CGate object via command line, use the [peladm create cgate](#) command.

Use the CGate object to define login requirements and directories with associated directory access rights for AS2 (HTTP) and AS3 (FTP) based transfers.

Using the GUI

To create a CGate object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGateGroup** sub-node, and from the context menu that appears, select **New CGateGroup...**

Gateway displays the *New CGate* window. Complete this window to define the characteristics of your new CGate.

Use the CGate object to define login requirements and directories with associated directory access rights for AS2 (HTTP) and AS3 (FTP) based transfers. Refer to [Working with CGates and CGateGroups](#).

Using EDIINT on Gateway

When you have carried out the configuration, confirmed your network connection and created the necessary objects, you can then:

- Submit a Transfer Request to send a file to a business partner via EDIINT, optionally requesting a receipt
- Receive EDIINT messages from remote business partners, and generate and return receipts to the sending partner

[Overview: Trading Partners](#)

[About EDIINT](#)

[Working with Trading Partner objects](#)

[Transfer Examples: EDIINT AS2 transfer](#)

RosettaNet

[The RosettaNet protocol](#)

[Gateway/RosettaNet message processing](#)

About the RosettaNet protocol

RosettaNet is an exchange protocol that provides standards for business exchanges.

Within RosettaNet, there are sets of RosettaNet *specifications*. These *RosettaNet Implementation Framework* (RNIF) specifications describe how documents can be packaged to create RosettaNet envelopes. The RosettaNet envelope can then be transmitted over the network, for example the Internet.

So far, two versions of the RNIF specifications have been published: RNIF V1.1 and RNIF V2.0. While both of these versions rely on XML as the standard format for payloads, the resulting envelopes differ slightly, and are called:

- RosettaNet Objects for RNIF V1.1
- RosettaNet Messages for V2.0

For more information on RosettaNet and RNIFs, refer to the www.rosettanet.org website.

RosettaNet PIP

Before you can understand how RosettaNet functions, you first need to become familiar with the concept of PIPs. A PIP, Partner Interface Process, is a set of processes that defines business exchanges between trading partners. This set of generic, standardized processes specifies business document structure and format for trading partners.

A variety of PIP are available, each of which outlines a different business process exchange. The user selects the PIPs that are appropriate for his business needs. RosettaNet then provides the packaging and security.

For example, suppose that Business Partner A wants to send a Purchase Order to Business Partner B using RosettaNet. (This exchange is described in detail in the PIP 3A4 *Request Purchase Order*.) According to this particular PIP, a process similar to the following sequence could occur:

- Partner A sends a *Purchase Order Request* message to Partner B
- Partner B must acknowledge the order within 2 hours (if not, Partner A resends the Purchase Order)
- Partner B responds to Partner A by sending back a *Purchase Order Confirmation* message within the predefined time period
- Partner A acknowledges the response by sending an ACK, *Receipt Acknowledgment Message*, to Partner B

The precise actions that occur depend on the specific PIP that these Business Partners chose. Additionally, for each of the above actions, the PIP can include information as to the level of security to be used, need for a signature, security for the message contents, need for an acknowledgment, time frames for these exchanges, and so on. In this sense, each PIP is unique and serves a separate business need.

RosettaNet exchanges

A RosettaNet exchange revolves around the concept of PIPs, as described in the previous section. The RNIF, RosettaNet Implementation Framework, deals with 'packaging' the PIP. This packaging process consists of encapsulating the XML-formatted PIP, the XML headers, and any attachments in a MIME package. When the security is added to this package, you have a RosettaNet message.

After the RosettaNet message is encapsulated and encrypted, it is then sent over the network. When it is received, the message is unpacked and decrypted. If the message contains an attachment, the receiver saves the attachment. Depending on the PIP and trading partner, the request is processed and a message is returned to the message originator.

All of these processes occur between defined Trading Partners. In this Trading Partner environment, RosettaNet PIPs aid in business transactions, and help determine the best course of action between business partners.

RosettaNet and Gateway

RosettaNet functions in Gateway at the transport layer. Gateway adds security and HTTP or SMTP/POP3 formatting. (Gateway does not manage or process PIPs.) It is one product of the Axway Platform offering RosettaNet services.

Using Axway products to deploy a full RosettaNet solution is outside the scope of this document. For more information on how to use Axway products with RosettaNet, contact your Axway Sales Representative at www.axway.com.

Gateway/RosettaNet message processing

This section explains the message processing that is performed by Gateway when a RosettaNet message arrives:

When an incoming transfer arrives, Gateway performs the following steps:

- Removes protocol envelope (HTTP/SMTP)
- Removes S/MIME envelope
- Performs a validation process

The XFBCONN interface

When applicable, the XFBCONN interface is used by Gateway to:

- Notify Integrator of the outgoing messages processing status
- Inform Integrator incoming message arrivals, and their associated parameters

For details of the functioning of the Integrator product, refer to the Axway Integrator documentation.

Outgoing RosettaNet message processing

Outgoing RosettaNet Request/Response

A RosettaNet message request is processed in multiple phases:

1. The transfer request undergoes validation checking (checks only the Preamble, DeliveryHeader and ServiceHeader). If the check is unsuccessful, the transfer is returned to the caller.
2. The transfer request is accepted and queued in the Gateway Mailbox. It is then scheduled for packaging.
3. After the RosettaNet message is successfully encoded, it is scheduled for network transport. If a communication error occurs, and transfer retry count is exhausted by successive retrials, the transfer is cancelled.
4. After the message is successfully sent to the partner, the transfer is flagged as ENDED.

Incoming RosettaNet message processing

The processing steps for an incoming RosettaNet message consist of removing the S/MIME envelope, and extracting the RosettaNet component parts and attachments from the package.

The RosettaNet component parts include:

- <Preamble>
- <DeliveryHeader> (version dependent)
- <ServiceHeader>

These elements must be validated for an incoming message. If an error occurs during the validation procedure, the message status is reported. The faulty message is then discarded. If no error occurs during these initial steps, the complete message is constructed as an XML file.

The incoming messaging processing occurs as follows:

1. For a signed message, the signature envelope is removed and the signature is verified.
2. The <Preamble> element is extracted and validated.
3. The <DeliveryHeader> element is extracted and validated.
4. If the <ServiceHeader> is encrypted, it is decrypted.
5. The <ServiceHeader> element is extracted and validated.
6. The <ServiceContent> element is extracted but is not validated.

At this point, Gateway has completed its processing. The message is sent to the next RosettaNet component. The process may continue by interpreting the message, sending a receipt acknowledgment, an exception, a Notification of Failure PIP, a response, and so on.

If a decoding error occurs, the MIME may be incomplete and contain only a partial RosettaNet message. An exception signal is returned to the partner *if* transport debug headers were sent by the partner, and are allowed by the local security policy. If transport debug headers were not sent by the partner, no exception signal is sent.

[Configuring the RosettaNet environment](#)

[Using RosettaNet](#)

Axway Gateway: Protocols

Configuring the RosettaNet environment

[Prerequisites](#)

[Required objects](#)

[Creating objects for RosettaNet transfers](#)

Prerequisites

Before you can send and receive transfers via RosettaNet and perform processing on these transfers, you need to:

- Confirm that you have a TCP/IP connection, and that your business exchange partners are connected to the network at the time you initiate or expect to receive a file transfer or acknowledgment.
- Activate the appropriate supporting protocol processes in the Gateway [configuration menu](#).
- Ensure that you have access to the necessary Public Key Infrastructure (PKI). Each exchange partner must possess the public certificate for the remote and local partner as well as the associated private key. See the *Security Guide* > [Certificates and keys](#) for further information (requires login).
- With each business exchange partner, you establish a RosettaNet exchange contract known as a *Trading Partner Agreement*. In a Trading Partner Agreement you mutually establish:
 - RosettaNet identifiers, unique names for every exchange entity
 - Target file directory locations for your exchanges
 - Signing and encryption algorithmsWhen you configure your respective RosettaNet compliant applications, you must scrupulously respect the terms of your Trading Partner Agreement.

Required objects

You require a different set of objects depending on whether you are sending or receiving a transfer. A complete RosettaNet exchange normally comprises a request phase and a file transfer phase. Between these two phases, the initiator and responder machines trade roles. For this reason, whether Gateway is acting as initiator or responder in a RosettaNet exchange, you require the same set of objects to complete the file transfer/receipt transfer cycle.

Sending transfers

To successfully initiate a RosettaNet transfer you require as a minimum the following Gateway objects:

- Remote Site object: to specify transport layer characteristics of the remote machine to which you are sending a file or a receipt
- Local Trading Partner object, to specify:
 - Local identification
 - Signing certificate and signing algorithm on emission
 - Optionally, the encryption certificate necessary for decrypting the returned receipt
- Remote Trading Partner object: to specify the remote compression, signature, encryption and encapsulation processes for the exchange with your business partner

Receiving transfers

To successfully receive a RosettaNet transfer from a remote business partner, you require as a minimum the following Gateway objects:

- Remote Site object: to specify transport layer characteristics of the remote machine from which you are receiving the file
- Local Trading Partner object, to specify:
 - Local identification
 - Signing certificate and signing algorithm for controlling the message on reception
 - Optionally, the encryption certificate necessary for encrypting the return receipt
- Remote Trading Partner object: to specify the remote compression, signature, encryption and encapsulation processes for exchanges with your business partner
- CGateGroup and CGate objects: to control the remote connection to your machine by your business partner for transfer of the file or receipt

Creating objects for RosettaNet transfers

This section describes how to create the following objects for RosettaNet transfers:

- Remote Site objects
- Local Trading Partner objects
- Remote Trading Partner objects
- CGateGroup objects
- CGate objects

Creating a Remote Site object for RosettaNet transfers

The Remote Site object provides Gateway with information about the remote machine to which you are sending your RosettaNet messages and receipts.

You can create Remote Site objects in Gateway either via the GUI or using command line.

Using the GUI

To create a Remote Site object via the GUI:

1. In the left pane of the GUI main window, click to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node, then select **New** from the context menu to display the *New Remote Site* window.
3. In this window, enter values in the fields to define the characteristics of the [Remote Site](#) object.

The following Remote Site attributes specifically concern RosettaNet Transfer Requests.

Tab	Field	Parameter value
General	Protocol	Select the appropriate protocol: • RN_HTTP • RN_SMTP • RN_POP3

Using command line

To create a Remote Site object via command line, use the [peladm create site](#) command. The precise values you enter depend on your transfer requirements. The following parameters specifically concern RosettaNet Transfer Requests.

Parameter	RosettaNet-related value
-protocol (-pr)	Specify one of the following protocols: • RN_HTTP • RN_SMTP • RN_POP3

Creating a Local Trading Partner object for RosettaNet transfers

The Local Trading Partner object provides Gateway with information about the RosettaNet compression, signature, encryption and encapsulation operations that you use to transfer your message. It also enables you to specify how to handle the acknowledgments returned by remote partners.

You can create Local Trading Partner objects in Gateway either via the GUI or using command line.

Using the GUI

To create a Local Trading Partner object via the GUI:

1. In the left pane of the GUI main window, click to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node, then select **New** from the context menu to display the *Trading Partner* window.

In this window, you define the characteristics of the Local Trading Partner object by entering values in the appropriate fields.

Refer to [Working with Trading Partner objects](#).

Using command line

To create a Local Trading Partner object via command line, use the [peladm create tradepart](#) command.

Creating a Remote Trading Partner object for RosettaNet transfers

The Remote Trading Partner object provides Gateway with information about the compression, signature, encryption and encapsulation operations defined by the RosettaNet protocol.

You can create Remote Trading Partner objects in Gateway either via the GUI or using command line.

Using the GUI

1. In the left pane of the GUI main window, click to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node, then select **New** from the context menu to display the *Trading Partner* window.

In this window, you define the characteristics of the Remote Trading Partner object by entering values in the appropriate fields.

Refer to [Working with Trading Partner objects](#).

Using command line

To create a Remote Trading Partner object via command line, use the [peladm create tradepart](#) command.

Creating a CGateGroup object for RosettaNet transfers

To receive RosettaNet messages or message receipts from a business partner on your local machine, you control remote connection attempts to your machine via the CGateGroup object and CGate object.

You can create CGateGroup and CGate objects in Gateway either via the GUI or using command line.

Using the GUI

To create a CGateGroup object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **CGate** sub-node, and from the context menu that appears, select **New CGateGroup...**

Gateway displays the *New CGateGroup* window. Complete this window to define the characteristics of your new CGateGroup.

Refer to [Working with CGates and CGateGroups](#).

In particular, when you create the CGateGroup, specify, in the **Protocol** field, the protocol that is used by the connecting partner:

- **RN_HTTP**
- **RN_SMTP**
- **RN_POP3**

Using command line

To create a CGateGroup object via command line, use the [peladm create cgategroup](#) command.

To enable the reception of a connection based on a RosettaNet protocol, in the `protocol` attribute of this command, enter one of the following values to specify the RosettaNet protocol format used by your remote partner for the acknowledgment transfer:

- RN_HTTP
- RN_SMTP
- RN_POP3

Creating a CGate object for RosettaNet transfers

To specify the login name and password requirements for AS1 or AS2 connections to your local machine, create a CGate in the CGateGroup you created in the previous section.

You can create CGate objects in Gateway either via the GUI or using command line.

Using the GUI

To create a CGate object via the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Click  to expand the **CGate** node.
3. Select the CGateGroup created in the previous section.
4. Right-click the CGateGroup, and from the context menu that appears, select **New > CGate....** Gateway displays the *New CGate* window. Complete this window to define the characteristics of your new CGate.
Use the CGate object to define login requirements and directories with associated directory access rights for AS2 (HTTP) and AS3 (FTP) based transfers.
Refer to [Working with CGates and CGateGroups](#).

Using command line

To create a CGate object via command line, use the [peladm create cgate](#) command.

Use the CGate object to define login requirements and directories with associated directory access rights for RN_HTTP based transfers.

[About RosettaNet](#)

[Working with Trading Partner objects](#)

[Working with Trading Partner objects \(command line\)](#)

[Using RosettaNet](#)

Axway Gateway: Protocols

Using RosettaNet

[Overview](#)

[Defining a RosettaNet outgoing request](#)

[Defining a RosettaNet Response, Exception Signal or acknowledgment](#)

Overview

This topic describes the Gateway parameters required for transfers using the RosettaNet protocol.

Prerequisites

Before you can use RosettaNet on Gateway, you must first:

- Configure Gateway for use with RosettaNet
- Confirm your network connection
- Create the necessary objects

For more information, refer to [Configuring the RosettaNet environment](#).

After completing these preliminary steps, you are ready to:

- Submit a Transfer Request to send a file to a business partner via RosettaNet
- Receive RosettaNet messages from remote business partners

RosettaNet monitoring

Each RosettaNet message (request, receipt acknowledgment, signal, and response) has a transfer record in Gateway. These messages are displayed in the Gateway log.

Defining a RosettaNet outgoing request

To submit an outgoing RosettaNet request, define the parameters listed in the table below. Parameters that have no value and description listed, have the same values and definitions as in other Gateway parameter lists.

Parameter	Value and definition
Trading Partner Info:	
TYPE (Mandatory)	TYPE_TRANS
TRADE_FORMAT	TRADE_FORMAT_RN
TRADE_ORG	TRADE_ORG and TRADE_ORG_ALIAS are mutually exclusive. If TRADE_ORG is specified, TRADE_FORMAT must also be defined.
TRADE_ORG_ALIAS	Only one of these can be specified. If both are specified, they are taken into account in the listed order. If none of these is defined, the default associated local partner defined in the destination partner is used.
TRADE_DEST (Mandatory)	TRADE_DEST, TRADE_DEST_ALIAS and EXT_DEA_ID are mutually exclusive.
TRADE_DEST_ALIAS (Mandatory)	If TRADE_DEST is specified, TRADE_FORMAT must also be present. Only one of these can be specified. If more than one is specified, they are taken into account in the listed order.
EXT_DEA_ID (Mandatory)	
RosettaNet Message Info:	

Parameter	Value and definition
FILE_COMPONENT (Mandatory)	FILE_COMPONENT is a MIME file containing the complete RosettaNet message. It contains up to five top level elements corresponding to the RosettaNet: • preamble header • delivery header (if RNIF V2.0) • service header • service content • attachments (if any)
TRADE_SIGN	Select from: • YES • NO If TRADE_SIGN is required and TRADE_SIGN_ALGO is not specified, the default value defined in destination trading partner is used.
TRADE_SIGN_ALGO	Select from: • MD5_RSA • SHA1_RSA If TRADE_SIGN is required and TRADE_SIGN_ALGO is not specified, the default value defined in destination trading partner is used.
TRADE_ENCRYPT (Applicable only for RN V2.0)	Select from: • YES • NO If TRADE_ENCRYPT is required and TRADE_ENCRYPT_ALGO is not specified, the default value defined in destination trading partner is used.

Parameter	Value and definition
TRADE_ENCRYPT_ALGO (Applicable only for RN V2.0)	Select from: • AES128 • AES256 • DES • 3DES • RC2_40 • RC4 If TRADE_ENCRYPT is required and TRADE_ENCRYPT_ALGO is not specified, the default value defined in destination trading partner is used. Note: Gateway supports RC4 with 128-bit key size. Gateway does not support RC4 with 40-bit or 64-bit key sizes.
TRADE_ENCRYPT_PARTS (Applicable only for RN V2.0)	Select from: • RN_PAYLOAD • RN_PAYLOAD_CONTAINER TRADE_ENCRYPT_PART indicates whether the whole payload container (service header plus service content) or only payload (service content) is to be encrypted.
RECEIPT_REQUEST_SYNC	Select from: • YES • NO RECEIPT_REQUEST_SYNC indicates whether it is a synchronous request.
RSEND_COUNT_MAX	10 retry attempts
RESEND_DELAY	300 (seconds)
Transport Info:	
REMOTE_AGENT	If REMOTE_AGENT is not specified, the one specified in the destination trading partner is used.
MAIL_SUBJECT (if transport is SMTP)	Select from: • FROM: if transport is SMTP • TO: if transport is SMTP
Schedule Info:	

Parameter	Value and definition
DATE_TO_BEGIN	The Schedule Information fields use the same values as for typical transfers.
DATE_TO_END	
PRIORITY	
RETRY_COUNT_MAX	
FROZEN_STATE	
Application Info:	
MODEL	The Application Information fields use the same values as for typical transfers.
PURGE_OPTION	
PARAM1	
PARAM2	
SENTINEL_TRANSFER_FILTER	
EX_SENTINEL_IDENT	
EX_SENTINEL_OBJECTNAME	

Defining a RosettaNet Response, Exception Signal or acknowledgment

To submit an outgoing RosettaNet response, exception signal, or acknowledgment, use the same values as for Defining a RosettaNet outgoing request.

[About RosettaNet](#)

[Configuring the RosettaNet environment](#)

Axway Gateway: Managing the Server

Starting and stopping the server

Server startup overview

Gateway is composed of a group of related executable programs. The programs activated at startup depend on the options chosen during installation and configuration.

UNIX environments

Specify whether or not the server daemons (for the Gateway user interface) are started at the same time as the product. By default, the server daemons start automatically.

Windows environments

Start Gateway in one of two ways:

- Manually: select **Start** in the Gateway program folder created by the installation process
- Automatically: configure Gateway as a service to automatically start on system start

After startup is completed, select **Process List** in the Gateway program folder to view the list of currently activated programs.

Before you start the Gateway server

Before starting Gateway, check the network resources needed for communication between Gateway and its partners.

IPC Queues

Gateway uses the OS's IPC queues for inter-process communication.

Refer to the *Installation Guide* > [Gateway kernel prerequisites](#) for their configuring to avoid delays during high-load.

TCP/IP subsystems

To check that you can reach remote systems from your computer, use the standard `ping` command with the hostname or IP address as argument. For more details, refer to the ping documentation supplied with your system.

SNA LU 6.2 subsystems

To check the SNA configuration, activate each LU 6.2 connection that Gateway uses.

If you are unable to activate LU-LU sessions, check the LU 6.2 local and remote parameters.

If LU-LU sessions are activated, and the session establishment failed at file transfer time, check that:

- The Remote Site is active, and
- The mode and the transaction name requested in *CPI symbolic destination* are defined in the Remote Site

If the problem persists, activate the audit function and the SNA trace.

Starting and stopping the Gateway server

You do not have to initialize this version of Gateway. The Installer executes the `gateinit` command for you.

Windows environments: Starting

To start Gateway, go to the Windows **Start** menu and select:

Programs > Axway Software > Axway [ProjectName]> Gateway > Start Gateway

You can configure Gateway to continue running between login sessions by installing it as a system service.

Windows environments: Stopping

To stop Gateway, go to the Windows **Start** menu and select:

Programs > Axway Software > Axway [ProjectName]> Gateway > Stop Gateway

If you started Gateway as a system service, you must stop it via a Service applet in the Control Panel. Alternatively, use the `net stop xxxxx` command.

The Axway MFT Navigator server and the Gateway GUI server stop at the same time as the product.

UNIX environments: Starting and stopping

The **gatestart** and **gatestop** utilities enable you to start and stop Gateway in the UNIX environment.

Before using Gateway program files, set the product environment for your UNIX session.

Execute the Shell script **profile**, located by default in:

```
<Gateway installation directory>/run_time/etc
```

Starting

To start Gateway, enter the command:

```
gatestart
```

This command starts Gateway, and by extension, activates the tracking functions of the Mailbox. Use this command when Gateway is stopped.

Note: Do not use the **pelmon start** command that was used in earlier versions of Gateway.

Stopping

To stop Gateway, enter the command:

```
gatestop
```

This command shuts down Gateway, and by extension, stops Mailbox tracking.

Note: Do not use the **pelmon stop** command that was used in earlier versions of Gateway.

Starting and stopping the Navigator server

In Windows and UNIX environments, the Gateway Navigator server (GUI server) starts and stops at the same time as Gateway.

Starting and stopping the Axway MFT Navigator server

The MFT Navigator server starts and stops at the same time as Gateway.

After you start the Gateway server

After starting Gateway, check the Gateway processes.

Windows environments

After startup is completed, select **Process List** in the Gateway program folder to view the list of currently activated programs.

UNIX environments

Status

Use the `gatestatus` command to obtain the current status of Gateway, and by extension, the Mailbox.

Note: Do not use the `pelmon status` command that was used in earlier versions of Gateway.

For a list of possible responses, refer to [About gatestatus messages](#).

Processes

After entering the `gatestart` or `gatestop` command, check that Gateway processes have correctly started or stopped.

Enter the command:

```
ps -e | grep -E " p_| q_| h_| notif| ipelapi"
```

When Gateway is started, the `ps` (or Process List) command must show a supervisor process (`p_sup`) and a system process (`p_sys`).

Other processes are launched depending on the networks, protocol resources used and product options (for example, Sentinel).

For TCP/IP networks

Use the following commands to specify the appropriate protocol for the TCP/IP network.

Command	Specifies the protocol for the TCP/IP network...
p_as1pop3sock	EDIINT AS1 POP3
p_as1smtpsock	EDIINT AS1 SMTP
p_as2sock	EDIINT AS2
p_as3sock	EDIINT AS3
p_fsitsock	PeSIT protocol
clp_ftpsock	FTP protocol
p_fodtsock	OFTP protocol
p_fpelsock	PEL protocol
p_fhttpsock	HTTP protocol
p_fpop3sock	POP3 protocol
p_sftpsock	SFTP protocol
p_smtpsock	SMTP protocol
h_trade_job	S/MIME, AS1, AS2, AS3

For SNA networks

Use the following command to specify the appropriate protocol for the SNA network.

Command	Specifies...
p_fsit62	the use of an SNA network with PeSIT protocol

Only the PeSIT E protocol is supported for SNA.

SNA processes are not started when you start Gateway. These processes are started when a connection occurs. For this reason, they are referred to as *dynamic processes*.

Multiple processes for the same protocol can be displayed if several connections occur. These processes stop when the connections are dropped.

Message queues (UNIX environments)

Gateway uses message queues. In the UNIX environment, use the `ipcs` command to check that the message queues are started.

```
ipcs -q | grep <OWNER>
```

After you stop the Gateway server

When you stop Gateway, you should also stop the list of processes being printed, and delete the message queues.

Checking the status of processes

Process traces can be found in the folder `p_home_dir/run_time/tmp`. Select the file that corresponds to your process to view messages that describe why processes are not launched.

The following table lists the types of trace file (where XX is the PID number of the process).

Trace file name	Trace type
SUP-<timestamp>-<pid>.out	SUPERVISOR traces
SYS-<timestamp>-<pid>.out	SYSTEM traces
FT_AS1_POP3-<timestamp>-<pid>.out	EDIINT AS1 POP3 protocol traces
FT_AS1_SMTP-<timestamp>-<pid>.out	EDIINT AS1 SMTP protocol traces
FT_AS2_X-<timestamp>-<pid>.out	EDIINT AS2 protocol traces
FT_AS3_X-<timestamp>-<pid>.out	EDIINT AS3 protocol traces
FT_PHSE-<timestamp>-<pid>.out	PeSIT E protocol traces

Trace file name	Trace type
FT_FTP-<timestamp>-<pid>.out	FTP protocol traces
TCP_PHSE-<timestamp>-<pid>.out	TCP traces with PeSIT E protocol
TCP_FTP-<timestamp>-<pid>.out	TCP traces with FTP protocol
PHECPICR.XX-<timestamp>-<pid>.out	SNA traces for outgoing call with PHSE protocol
PHECPICI.XX-<timestamp>-<pid>.out	SNA traces for incoming call with PHSE protocol

Axway Gateway: Managing the Server

Starting and stopping the server

Server startup overview

Gateway is composed of a group of related executable programs. The programs activated at startup depend on the options chosen during installation and configuration.

UNIX environments

Specify whether or not the server daemons (for the Gateway user interface) are started at the same time as the product. By default, the server daemons start automatically.

Windows environments

Start Gateway in one of two ways:

- Manually: select **Start** in the Gateway program folder created by the installation process
- Automatically: configure Gateway as a service to automatically start on system start

After startup is completed, select **Process List** in the Gateway program folder to view the list of currently activated programs.

Before you start the Gateway server

Before starting Gateway, check the network resources needed for communication between Gateway and its partners.

IPC Queues

Gateway uses the OS's IPC queues for inter-process communication.

Refer to the *Installation Guide* > [Gateway kernel prerequisites](#) for their configuring to avoid delays during high-load.

TCP/IP subsystems

To check that you can reach remote systems from your computer, use the standard `ping` command with the hostname or IP address as argument. For more details, refer to the ping documentation supplied with your system.

SNA LU 6.2 subsystems

To check the SNA configuration, activate each LU 6.2 connection that Gateway uses.

If you are unable to activate LU-LU sessions, check the LU 6.2 local and remote parameters.

If LU-LU sessions are activated, and the session establishment failed at file transfer time, check that:

- The Remote Site is active, and
- The mode and the transaction name requested in *CPI symbolic destination* are defined in the Remote Site

If the problem persists, activate the audit function and the SNA trace.

Starting and stopping the Gateway server

You do not have to initialize this version of Gateway. The Installer executes the `gateinit` command for you.

Windows environments: Starting

To start Gateway, go to the Windows **Start** menu and select:

Programs > Axway Software > Axway [ProjectName]> Gateway > Start Gateway

You can configure Gateway to continue running between login sessions by installing it as a system service.

Windows environments: Stopping

To stop Gateway, go to the Windows **Start** menu and select:

Programs > Axway Software > Axway [ProjectName]> Gateway > Stop Gateway

If you started Gateway as a system service, you must stop it via a Service applet in the Control Panel. Alternatively, use the `net stop xxxxx` command.

The Axway MFT Navigator server and the Gateway GUI server stop at the same time as the product.

UNIX environments: Starting and stopping

The `gatestart` and `gatestop` utilities enable you to start and stop Gateway in the UNIX environment.

Before using Gateway program files, set the product environment for your UNIX session.

Execute the Shell script `profile`, located by default in:

```
<Gateway installation directory>/run_time/etc
```

Starting

To start Gateway, enter the command:

gatestart

This command starts Gateway, and by extension, activates the tracking functions of the Mailbox. Use this command when Gateway is stopped.

Note: Do not use the `pelmon start` command that was used in earlier versions of Gateway.

Stopping

To stop Gateway, enter the command:

gatestop

This command shuts down Gateway, and by extension, stops Mailbox tracking.

Note: Do not use the `pelmon stop` command that was used in earlier versions of Gateway.

Starting and stopping the Navigator server

In Windows and UNIX environments, the Gateway Navigator server (GUI server) starts and stops at the same time as Gateway.

Starting and stopping the Axway MFT Navigator server

The MFT Navigator server starts and stops at the same time as Gateway.

After you start the Gateway server

After starting Gateway, check the Gateway processes.

Windows environments

After startup is completed, select **Process List** in the Gateway program folder to view the list of currently activated programs.

UNIX environments

Status

Use the `gatestatus` command to obtain the current status of Gateway, and by extension, the Mailbox.

Note: Do not use the `pelmon status` command that was used in earlier versions of Gateway.

For a list of possible responses, refer to [About gatestatus messages](#).

Processes

After entering the `gatestart` or `gatestop` command, check that Gateway processes have correctly started or stopped.

Enter the command:

```
ps -e | grep -E " p_| q_| h_| notif| ipelapi"
```

When Gateway is started, the `ps` (or Process List) command must show a supervisor process (`p_sup`) and a system process (`p_sys`).

Other processes are launched depending on the networks, protocol resources used and product options (for example, Sentinel).

For TCP/IP networks

Use the following commands to specify the appropriate protocol for the TCP/IP network.

Command	Specifies the protocol for the TCP/IP network...
p_as1pop3sock	EDIINT AS1 POP3
p_as1smtpsock	EDIINT AS1 SMTP
p_as2sock	EDIINT AS2
p_as3sock	EDIINT AS3
p_fsitsock	PeSIT protocol
clp_ftpsock	FTP protocol
p_fodtsock	OFTP protocol
p_fpelsock	PEL protocol
p_fhttpsock	HTTP protocol
p_fpop3sock	POP3 protocol
p_sftpsock	SFTP protocol
p_smtpsock	SMTP protocol
h_trade_job	S/MIME, AS1, AS2, AS3

For SNA networks

Use the following command to specify the appropriate protocol for the SNA network.

Command	Specifies...
p_fsit62	the use of an SNA network with PeSIT protocol

Only the PeSIT E protocol is supported for SNA.

SNA processes are not started when you start Gateway. These processes are started when a connection occurs. For this reason, they are referred to as *dynamic processes*.

Multiple processes for the same protocol can be displayed if several connections occur. These processes stop when the connections are dropped.

Message queues (UNIX environments)

Gateway uses message queues. In the UNIX environment, use the `ipcs` command to check that the message queues are started.

```
ipcs -q | grep <OWNER>
```

After you stop the Gateway server

When you stop Gateway, you should also stop the list of processes being printed, and delete the message queues.

Checking the status of processes

Process traces can be found in the folder `p_home_dir/run_time/tmp`. Select the file that corresponds to your process to view messages that describe why processes are not launched.

The following table lists the types of trace file (where XX is the PID number of the process).

Trace file name	Trace type
SUP-<timestamp>-<pid>.out	SUPERVISOR traces
SYS-<timestamp>-<pid>.out	SYSTEM traces
FT_AS1_POP3-<timestamp>-<pid>.out	EDIINT AS1 POP3 protocol traces
FT_AS1_SMTP-<timestamp>-<pid>.out	EDIINT AS1 SMTP protocol traces
FT_AS2_X-<timestamp>-<pid>.out	EDIINT AS2 protocol traces
FT_AS3_X-<timestamp>-<pid>.out	EDIINT AS3 protocol traces
FT_PHSE-<timestamp>-<pid>.out	PeSIT E protocol traces

Trace file name	Trace type
FT_FTP-<timestamp>-<pid>.out	FTP protocol traces
TCP_PHSE-<timestamp>-<pid>.out	TCP traces with PeSIT E protocol
TCP_FTP-<timestamp>-<pid>.out	TCP traces with FTP protocol
PHECPICR.XX-<timestamp>-<pid>.out	SNA traces for outgoing call with PHSE protocol
PHECPICI.XX-<timestamp>-<pid>.out	SNA traces for incoming call with PHSE protocol

Axway Gateway: Managing the Server

Importing and exporting objects

Overview

The **pelbase** command enables you to:

- Export and import objects between two Gateways
- Transfer objects between two different versions of Gateway on the same machine, for example when upgrading to a newer version

Using the pelbase command

To transfer objects from one Gateway to another, or between versions:

1. On the originating Gateway, execute the **pelbase export** command for the object or objects you want to export.

The result of each **pelbase export** command is a file describing the objects that you can transfer between machines or versions.

Use the object attribute parameters as object selection criteria. The command creates an output file with as many objects as meet all the criteria.

Use the **-output (-f)** parameter to specify the name of the resulting file. If you do not enter a value, Gateway prints the command results on the screen.

When exporting from Gateway 6.17.2 or later, specifying an encryption key using the `-encryption_key_file(-ekf)` is mandatory. See the [Security Guide](#) for how to generate an encryption key file.

2. On the target Gateway, execute the `pelbase import` command. Execute the command as many times as necessary to import each file you created with the `pelbase export` command.

If the objects were exported with passwords encrypted, use the `-encryption_key_file(-ekf)` parameter to specify the key used for password encryption.

If the objects were exported from a Gateway version which does not support password encryption for exported objects, use the `-import_pwd_cleartext` parameter to import clear-text passwords.

Password encryption applies for the following `pelbase` subcommands:

`pelbase export_site`

`pelbase export_cgate`

`pelbase export_sgate`

`pelbase export_tradepart`

and when importing the corresponding objects

pelbase subcommands

- [`pelbase export site`](#)
- [`pelbase export lsite`](#)
- [`pelbase export appli`](#)
- [`pelbase export list`](#)
- [`pelbase export model`](#)
- [`pelbase export purge model`](#)
- [`pelbase export user`](#)
- [`pelbase export profile`](#)
- [`pelbase export cgate`](#)
- [`pelbase export cgategroup`](#)
- [`pelbase export sgate`](#)
- [`pelbase export sgategroup`](#)
- [`pelbase export decisionrule`](#)
- [`pelbase export ruletable`](#)
- [`pelbase export proplist`](#)
- [`pelbase export tradepart`](#)
- [`pelbase export proxy`](#)

- `pelbase export_audit`: see [Working with Audits \(command line\)](#) on page 2398
- [pelbase import](#)

pelbase export_site

Syntax	<code>pelbase export_site</code> {-encryption_key_file} [-alias] [-proto_ident] [-group] [-protocol] [-comm_type] [-dest_address] [-format] [-user_fmt] [-output] [-append]

Description	Use this command to export fields and field values of one or more Sites. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Site attributes as selection criteria, as described below. For information about additional parameters for SWIFTNet output channel data, refer to SWIFTNet output channels.

Parameters

Mandatory

`-encryption_key_file(-ekf)`: Enter the path to the encryption key file to be used to encrypt passwords contained by the object.

Optional

`-alias (-a)`: Enter an alias that identifies the Site object you want to export.

Maximum: 31 alphanumeric characters.

`-proto_ident (-pi)`: Enter the protocol identifier of the Site object(s) you want to export.

Maximum: 25 alphanumeric characters.

`-group (-gr)`: Enter the group of the Site object(s) you want to export.

Maximum: 15 alphanumeric characters.

`-protocol (-pr)` : Enter the interchange protocol for the Site object(s) you want to export.

Enter one of the following values:

- **PEL**
- **ODETTE** (OFTP)
- **PHSE**
- **PHSD**
- **FTP**
- **HTTP**
- **FTP_HTTP**
- **POP3**
- **SFTP**
- **SMTP**
- **AS2**
- **AS3**
- **AS1_POP3**
- **AS1_SMTP**
- **RN_HTTP**
- **RN_POP3**
- **RN_SMTP**
- **SWIFTNET**
- **JMS**
- **X420**
- **X400**

`-comm_type (-ct)` : Enter the type of communications interface that corresponds to the Site object(s) you want to export:

- **PAD**
 - **TCPIP** (default)
 - **LU62**
 - **DSA**
-

`-dest_address (-da)` : Enter the Address to call (Remote Site network address).

In TCP/IP, specify the port number (example: 193.56.234.14/22000).

Maximum: 31 alphanumeric characters.

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
 - **C** = C-Shell. Example: set field = *value*
 - **U** = User format
-

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
 - %V = value of field
-

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_site
-encryption_key_file <decryption_key.dat>
-gr gdefault
-da SiteB
-f UpgradeSite
```

Gateway creates an export file named `UpgradeSite` containing all Site objects with SiteB as destination address.

pelbase export_lsite

Syntax

```
pelbase export_lsite [-alias] [-protocol_
ident] [-agent_alias] [-format] [-user_fmt] [-
output] [-append]
```

Description	Use this command to export the fields and field values of one or more Logical Sites. Gateway either displays the results on the screen or copies the results to an ASCII file. Use LSite attributes as selection criteria, as described below.

Note: It is not recommended to use Logical Sites. Logical Sites are still supported in this version of Gateway but are no longer accessible via the GUI.

Parameters <i>all optional</i>

<code>-alias (-a)</code> : Enter an alias that identifies the Logical Site object you want to export. Maximum: 31 alphanumeric characters.

<code>-protocol_ident (-pi)</code> : Enter the protocol identifier of the Logical Site object(s) you want to export. Maximum: 25 alphanumeric characters.

<code>-agent_alias (-ag)</code> : Enter the alias of an associated Site. Maximum: 31 alphanumeric characters

<code>-format (-of)</code> : Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format

<code>-user_fmt (-uf)</code> : User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field

<code>-output (-f)</code> : Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters.

<code>-append (-ap)</code> : Concatenation at the end of the file (if it exists). This parameter takes no value.

Example

Enter the command:

```
pelbase export_lsite  
-a SITLOG01  
-of U  
-uf 'The value of field %N is %V'  
-f ExpLsite
```

Gateway creates an export file named `ExpLsite` containing the definition of Logical Site SITLOG01.

pelbase export_appli

Syntax

```
pelbase export_appli [-name] [-group] [-site]  
[-format] [-user_fmt] [-output] [-append]
```

Description

Use this command to export the fields and field values of one or more Applications. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Appli attributes as selection criteria, as described below.

Parameters*all optional***-name (-n)** : Enter an Application name.

Maximum: 8 alphanumeric characters for PEL.

Maximum: 16 alphanumeric characters for all other protocols.

-group (-gr) : Enter the group of the Application object(s) you want to export.

Maximum: 15 alphanumeric characters.

-site (-si) : Enter the Site name of the Application object you want to export. Gateway selects Application objects having the same group as the Site you specify.

Maximum: 31 alphanumeric characters.

-format (-of) : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
- **C** = C-Shell. Example: set field = *value*
- **U** = User format

-user_fmt (-uf) : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
- %V = value of field

-output (-f) : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

-append (-ap) : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

pelbase export_appli**-n APPLI****-f ExpAppli**Gateway creates the `ExpAppli` export file. From Application objects named APPLI, it extracts all fields and corresponding values, and writes them to the file.

pelbase export_list

Syntax	pelbase export_list [-list_name] [-comments] [-format] [-user_fmt] [-output] [-append]
Description	Use this command to export the fields and field values of one or more Diffusion Lists. Gateway either displays the results on the screen or copies the results to an ASCII file. Use List attributes as selection criteria, as described below.
Parameters <i>all optional</i>	-list_name (-n) : Enter an existing Diffusion List name. Maximum: 31 alphanumeric characters -comments (-cts) : Enter user comments. Maximum: 80 alphanumeric characters. -format (-of) : Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format -user_fmt (-uf) : User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field -output (-f) : Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters. -append (-ap) : Concatenation at the end of the file (if it exists). This parameter takes no value. Example Enter the command: <pre>pelbase export_list -n LIST1 -f ExpList</pre> Gateway creates the <code>ExpList</code> export file. From Diffusion List objects named <code>LIST1</code>, it extracts all fields and corresponding values, and writes them to the file.

pelbase export_model

Syntax	pelbase export_model [-model] [-protocol] [-comments] [-format] [-user_fmt] [-output] [-append]
Description	Use this command to export the fields and field values of one or more Model objects. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Model attributes as selection criteria, as described below.

Parameters*all optional*

`-model (-ml)` : Enter an existing Model name.
Maximum: 31 alphanumeric characters.

`-protocol (-pr)` : The protocol identifier of the Model object (s) you want to export.

Specify one of the following values:

- **PEL**
 - **ODETTE** (OFTP)
 - **PHSE**
 - **PHSD**
 - **FTP**
 - **HTTP**
 - **POP3**
 - **SFTP**
 - **SMTP**
 - **TO_GTW**
 - **FROM_GTW**
 - **AS2**
 - **AS3**
 - **AS1_POP3**
 - **AS1_SMTP**
 - **RN_HTTP**
 - **RN_POP3**
 - **RN_SMTP**
 - **SWIFTNET**
 - **JMS**
 - **X420**
 - **GENERAL**
-

`-comments (-cts)` : Enter user comments.
Maximum: 80 alphanumeric characters.

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
 - **C** = C-Shell. Example: set field = *value*
 - **U** = User format
-

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
 - %V = value of field
-

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_model
-m1 MODEL1
-pr GENERAL
-f ExpModel
```

Gateway creates the `ExpModel` export file. In the database it locates Model objects named `MODEL1` that have `GENERAL` as protocol attribute values. From these Model objects, it extracts all fields and corresponding values, and writes them to the `ExpModel` file.

pelbase export_purge_model

Syntax

```
pelbase export_purge_model [-model] [-
comments] [-format] [-user_fmt] [-output] [-
append]
```

Description

Use this command to export the fields and field values of one or more Purge Model objects. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Purge Model attributes as selection criteria, as described below.

Parameters <i>all optional</i>	<code>-model (-ml)</code> : Enter an existing Purge Model name. Maximum: 31 alphanumeric characters. <code>-comments (-cts)</code> : Enter user comments. Maximum: 80 alphanumeric characters. <code>-format (-of)</code> : Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format <code>-user_fmt (-uf)</code> : User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field <code>-output (-f)</code> : Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters. <code>-append (-ap)</code> : Concatenation at the end of the file (if it exists). This parameter takes no value.

Example	Enter the command: <pre>pelbase export_purge_model -m1 PURGEMODEL1 -f ExpPurgeModel</pre> Gateway creates the <code>ExpPurgeModel</code> export file. In the database it locates Purge Model objects named <code>PURGEMODEL1</code>. From these Purge Model objects, it extracts all fields and corresponding values, and writes them to the <code>ExpPurgeModel</code> file.

pelbase export_user

Syntax	<pre>pelbase export_user [-username] [-group] [-profile] [-comments] [-disabled] [-administrator] [-allxfer] [-format] [-user_fmt] [-output] [-append]</pre>

Description	Use this command to export the fields and field values of one or more User objects. Gateway either displays the results on the screen or copies the results to an ASCII file. Use USER attributes as selection criteria, as described below.

Parameters*all optional*

`-user_name (-un)` : Enter an existing User login name.
Maximum: 31 alphanumeric characters.

`-group (-gr)` : Enter the Group of the User object(s) you want to export.
Maximum: 16 alphanumeric characters.

`-profile (-pr)` : Enter the Profile of the User object(s) you want to export.
Maximum: 16 alphanumeric characters.

`-comments (-cts)` : Enter user comments.
Maximum: 80 alphanumeric characters.

`-disabled (-dd)` : Enter a User status:

- **N**: User enabled
- **Y**: User disabled

`-administrator (-admin)` : Enter an administrator status:

- **N**: Non-administrator user
- **Y**: Administrator user

`-allxfer (-ax)` : Enter a User privilege status:

- **N**: User without *access all transfers* privilege
- **Y**: User with *access all transfers* privilege

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
- **C** = C-Shell. Example: set field = *value*
- **U** = User format

`-user_fmt (-uf)` : User format definition.
Enter a character string including the keywords %N and %V, where:

- %N = name of field
- %V = value of field

`-output (-f)` : Name of the output file.
If you do not specify this parameter, the standard output file is used.
Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_user  
-group FINANCE
```

Gateway exports all User entries that belong to the group FINANCE.

pelbase export_profile

Syntax

```
pelbase export_profile [-profile] [-comments]  
[-format] [-user_fmt] [-output] [-append]
```

Description

Use this command to export the fields and field values of one or more Profile objects. Gateway either displays the results on the screen or copies the results to an ASCII file. Use USER attributes as selection criteria, as described below.

Parameters*all optional*

`-profile (-pr)` : Enter the name of the Profile object(s) you want to export.

Maximum: 16 alphanumeric characters.

`-comments (-cts)` : Enter user comments.

Maximum: 80 alphanumeric characters.

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
- **C** = C-Shell. Example: set field = *value*
- **U** = User format

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
- %V = value of field

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_profile
-n OldProfile
-f ExpProfile
```

Gateway creates the `ExpProfile` export file. From the Profile object named `OldProfile`, it extracts all fields and corresponding values, and writes them to the `ExpProfile` file.

pelbase export_cgate

Syntax

```
pelbase export_cgate {-encryption_key_file (-ekf)} [-name] [-parent_group] [-protocol] [-format] [-user_fmt] [-output] [-append]
```

Description	Use this command to export the fields and field values of one or more CGates. Gateway either displays the results on the screen or copies the results to an ASCII file. Use CGate attributes as selection criteria, as described below.

Parameters*Mandatory*

`-encryption_key_file (-ekf)`: Enter the path to the encryption key file to be used to encrypt passwords contained by the object to be exported.

Optional

`-name (-n)`: Enter an existing CGate name.
Maximum: 32 alphanumeric characters.

`-parent_group (-pg)`: Enter the name of the parent CGateGroup with which the CGate is associated.
Maximum: 32 alphanumeric characters.

`-protocol (-pr)`: Enter the protocol identifier of the CGate object(s) you want to export.

Specify one of the following values:

- **FTP**
 - **HTTP**
 - **FTP_HTTP**
 - **PHSE**
 - **PHSD**
 - **ODETTE** (OFTP)
 - **PEL**
 - **POP3**
 - **SFTP**
 - **SMTP**
 - **AS2**
 - **AS3**
 - **AS1_POP3**
 - **AS1_SMTP**
 - **RN_HTTP**
 - **RN_POP3**
 - **RN_SMTP**
 - **JMS**
 - **X420**
 - ***** = undefined
-

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
 - **C** = C-Shell. Example: set field = *value*
 - **U** = User format
-

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
 - %V = value of field
-

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_cgate
  -encryption_key_file <decryption_key.dat>
  -n CGATE1
  -pg CGGROUP1
  -f ExpCGate
```

Gateway creates the `ExpCGate` export file. In the database it locates CGate objects named CGATE1 that have CGGROUP1 as `parent group` protocol attribute values. From these CGate objects, it extracts all fields and corresponding values, and writes them to the `ExpCGate` file.

pelbase export_categroup

Syntax

```
pelbase export_categroup [-name] [-format] [-user_fmt] [-output] [-append]
```

Description

Use this command to export the fields and field values of one or more CGateGroups. Gateway either displays the results on the screen or copies the results to an ASCII file. Use CGateGroup attributes as selection criteria, as described below.

Parameters <i>all optional</i>	<code>-name (-n)</code> : Enter an existing CGateGroup name. Maximum: 32 alphanumeric characters. <code>-format (-of)</code> : Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format <code>-user_fmt (-uf)</code> : User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field <code>-output (-f)</code> : Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters. <code>-append (-ap)</code> : Concatenation at the end of the file (if it exists). This parameter takes no value.
Example	Enter the command: <pre>pelbase export_cgategroup -n CGGROUPE1 -f ExpCGGroup</pre> Gateway creates the <code>ExpCGGroup</code> export file. In the database it locates CGate objects named CGGROUPE1. From these CGateGroup objects, it extracts all fields and corresponding values, and writes them to the <code>ExpCGGroup</code> file.

pelbase export_sgate

Syntax	<pre>pelbase export_sgate {-encryption_key_file} [-name] [-parent_group] [-format] [-user_fmt] [-output] [-append]</pre>
Description	Use this command to export the fields and field values of one or more SGates. Gateway either displays the results on the screen or copies the results to an ASCII file. Use SGate attributes as selection criteria, as described below.

Parameters	<i>Mandatory</i> <code>-encryption_key_file (-ekf)</code>: Enter the path to the encryption key file to be used to encrypt passwords contained by the object to be exported. <i>Optional</i> <code>-name (-n)</code>: Enter an existing SGate name. Maximum: 32 alphanumeric characters. <code>-parent_group (-pg)</code>: Enter the name of the parent SGateGroup with which the SGate is associated. Maximum: 32 alphanumeric characters. <code>-format (-of)</code>: Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format <code>-user_fmt (-uf)</code>: User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field <code>-output (-f)</code>: Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters. <code>-append (-ap)</code>: Concatenation at the end of the file (if it exists). This parameter takes no value.
Example	Enter the command: <pre> pelbase export_sgate -encryption_key_file <decryption_key.dat> -n SGATE1 -pg SGGROUP1 -f ExpSGate </pre> Gateway creates the <code>ExpSGate</code> export file. In the database it locates SGate objects named SGATE1 that have SGGROUP1 as <code>parent_group</code> protocol attribute values. From these SGate objects, it extracts all fields and corresponding values, and writes them to the <code>ExpSGate</code> file.

pelbase export_sgategroup

Syntax	pelbase export_sgategroup [-name] [-format] [-user_fmt] [-output] [-append]

Description	Use this command to export the fields and field values of one or more SGateGroups. Gateway either displays the results on the screen or copies the results to an ASCII file. Use SGateGroup attributes as selection criteria, as described below.

Parameters <i>all optional</i>	-name (-n) : Enter an existing SGateGroup name. Maximum: 32 alphanumeric characters. -format (-of) : Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format -user_fmt (-uf) : User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field -output (-f) : Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters. -append (-ap) : Concatenation at the end of the file (if it exists). This parameter takes no value.

Example	Enter the command: <pre>pelbase export_sgategroup -n SGGROUP1 -f ExpSGGGroup</pre> Gateway creates the <code>ExpSGGGroup</code> export file. In the database it locates SGate objects named SGGROUP1. From these SGateGroup objects, it extracts all fields and corresponding values, and writes them to the <code>ExpSGGGroup</code> file.

pelbase export_decisionrule

Syntax	pelbase export_decisionrule [-name] [-type_of_decisionrules] [-comments] [-execution_type] [-model] [-perl_script] [-prop_list_name] [-format] [-user_fmt] [-output] [-append]
Description	Use this command to export the fields and field values of one or more Decision Rule objects. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Decision Rule attributes as selection criteria, as described below.

Parameters*all optional*

`-name (-n)` : Enter a Decision Rule name.

`-type_of_decisionrules (-type)` : Type of Decision Rule to export.

Enter one of the following values:

- **XFER_CHANGE_STATE**
 - **TEMPORAL**
 - **FILE_STORING**
 - **XMS**
-

`-comments (-cts)` : Enter user comments.

Maximum: 80 alphanumeric characters.

`-execution_type (-et)` : The type of process triggered by the Decision Rule object(s) you want to export.

Enter one of the following values:

- **MODEL** = the default execution is defined by a Model
 - **AMTRIX**
 - **PERL**
 - **COMMAND**
 - **TO_XMS**
 - **DEFAULT**
 - **NONE**
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to Axway Integrator for processing
-

`-model (-ml)` : Model name.

`-perl_script (-ps)` : perl script file location.

`-prop_list_name (-pln)` : Name of the Property List to export.

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
 - **C** = C-Shell. Example: set field = *value*
 - **U** = User format
-

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
 - %V = value of field
-

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_decisionrule
-n OldRule
-f ExpRule
```

Gateway creates the `ExpRule` export file. From the Decision Rule object named `OldRule`, it extracts all fields and corresponding values, and writes them to the `ExpRule` file.

pelbase export_ruletable

Syntax

```
pelbase export_ruletable [-name] [-type_of_
rules_table] [-default_execution_type] [-
default_model] [-default_perl_script] [-xms_
connector_name] [-format] [-user_fmt] [-
output] [-append]
```

Description

Use this command to export the fields and field values of one or more Rule Table objects. Gateway either displays the results on the screen or copies the results to an ASCII file.

Parameters*all optional*

`-name (-n)` : Name of the Rule Table you want to export.

`-type_of_rules_table (-type)` : Type of Rule Table to export.

Enter one of the following values:

- **XFER_CHANGE_STATE**
 - **TEMPORAL**
 - **FILE_STORING**
 - **XMS**
-

`-default_execution_type (-det)` : The type of process triggered by the Rule Table object(s) you want to export.

Enter one of the following values:

- **MODEL** = the default execution is defined by a Model.
Specify the Model in the `default_model` parameter below.
 - **AMTRIX**
 - **PERL**
 - **COMMAND**
 - **TO_XMS**
 - **DEFAULT**
 - **NONE**
 - **LOG_CLEANING**
 - **MAILBOX_CLEANING**
 - **STAT_CLEANING**
 - **XIB** = redirect the Transfer to AMTriX/ Axway Integrator for processing
-

`-default_model (-dm)` : Enter the name of the default Model to use with the Rule Table.

`-default_perl_script (-dps)` : Enter the Perl script to apply by default to the current Transfer.

`-xms_connector_name (-xcn)` : Name of the Axway Messaging connector associated with the exported Rule Table.

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
- **C** = C-Shell. Example: set field = *value*
- **U** = User format

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords `%N` and `%V`, where:

- `%N` = name of field
- `%V` = value of field

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_ruletable
-n OldTable
-f ExpTable
```

pelbase export_proplist

Syntax

```
pelbase export_proplist [-properties_list] [-
comments] [-format] [-user_fmt] [-output] [-
append]
```

Description

Use this command to export the fields and field values of one or more Property List objects. Gateway either displays the results on the screen or copies the results to an ASCII file.

Parameters <i>all optional</i>	<code>-properties_list (-pl)</code> : Name of the Properties List you want to export. <code>-comments (-cts)</code> : Enter user comments. Maximum: 80 alphanumeric characters. <code>-format (-of)</code> : Output format. Enter one of: • S = Shell (default). Example: field = <i>value</i> • C = C-Shell. Example: set field = <i>value</i> • U = User format <code>-user_fmt (-uf)</code> : User format definition. Enter a character string including the keywords %N and %V, where: • %N = name of field • %V = value of field <code>-output (-f)</code> : Name of the output file. If you do not specify this parameter, the standard output file is used. Maximum: 127 alphanumeric characters. <code>-append (-ap)</code> : Concatenation at the end of the file (if it exists). This parameter takes no value.

Example	Enter the command: <pre> pelbase export_proplist -n OldPList -f ExpPList1 </pre>

pelbase export_tradepart

Syntax	<pre> pelbase export_tradepart {-encryption_key_ file} [-name] [-alias] [-is_local] [-trade_ format] [-format] [-user_fmt] [-output] [- append] </pre>
Description	Use this command to export the fields and field values of one or more Trading Partner objects. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Trading Partner attributes as selection criteria, as described below.

Parameters*Mandatory*

`-encryption_key_file (-ekf)`: Enter the path to the encryption key file to be used to encrypt passwords contained by the object to be exported.

Optional

`-name (-n)`: Enter the name of the Trading Partner object to be exported.

`-alias (-a)`: Enter the Local alias that identifies the Trading Partner object to be exported.

`-is_local (-loc)`: Enter one of the following values:

- **Y** = yes, local partner
 - **N** = no, remote partner
-

`-trade_format`: Format of the Trading Partner exchange.

Enter one of the following:

- **AS1**
 - **AS2**
 - **AS3**
 - **SMIME**
 - **RN**
 - **SWIFTNET**
 - **CMS**
-

`-format (-of)`: Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
 - **C** = C-Shell. Example: set field = *value*
 - **U** = User format
-

`-user_fmt (-uf)`: User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
 - %V = value of field
-

`-output (-f)`: Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_tradepart
  -encryption_key_file <decryption_key.dat>
  -n TradePart1
  -f TradePartExp
```

Gateway creates the `TradePartExp` export file. From the Trading Partner object named `TradePart1`, it extracts all fields and corresponding values, and writes them to the `TradePartExp` file.

pelbase export_proxy

Syntax

```
pelbase export_proxy [-name] [-comments] [-
protocol] [-format] [-user_fmt] [-output] [-
append]
```

Description

Use this command to export the fields and field values of one or more Proxies. Gateway either displays the results on the screen or copies the results to an ASCII file. Use Proxy attributes as selection criteria, as described below.

Parameters*all optional*

`-name (-n)` : Enter the Proxy name.

`-comments (-cts)` : Enter user comments.

Maximum: 80 alphanumeric characters.

`-protocol (-pr)` : The network layer protocol of the Proxy object(s) you want to export.

Enter one of the following values:

- **SOCKS4**
 - **FTP**
 - **HTTP**
 - **HTTPS**
-

`-format (-of)` : Output format. Enter one of:

- **S** = Shell (default). Example: field = *value*
 - **C** = C-Shell. Example: set field = *value*
 - **U** = User format
-

`-user_fmt (-uf)` : User format definition.

Enter a character string including the keywords %N and %V, where:

- %N = name of field
 - %V = value of field
-

`-output (-f)` : Name of the output file.

If you do not specify this parameter, the standard output file is used.

Maximum: 127 alphanumeric characters.

`-append (-ap)` : Concatenation at the end of the file (if it exists).

This parameter takes no value.

Example

Enter the command:

```
pelbase export_proxy
-n NetProxy
-f ExpProxy
```

Gateway creates the `ExpProxy` export file. From the Proxy object named `NetProxy`, it extracts all fields and corresponding values, and writes them to the `ExpProxy` file.

pelbase export_audit

See [Working with Audits \(command line\)](#) on page 2398

pelbase import

Syntax	<pre>pelbase import {-input} {-encryption_key_file} {-import_pwd_cleartext} [-site] [-lsite] [- appli] [-user] [-profile] [-list] [-model] [- proplist] [-purgemodel] [-cgategroup] [-cgate] [-decisionrule] [-ruletable] [-proxy] [- tradepart] [-xmsctor] [-sgategroup] [-sgate] [-error_free]</pre>

Description	Use this command to import the fields and field values of one or more objects contained in an importable object file. You can create importable files using the pelbase export subcommands. For information about additional parameters for SWIFTNet output channel data, refer to SWIFTNet queues and output channels.

Parameters*Mandatory*

`-input (-if)` : Enter the name of a file containing the objects you want to import.

`-encryption_key_file (-ekf)`: Enter the path to the encryption key file to be used to decrypt passwords contained by the object to be imported.

OR

`-import_pwd_cleartext`: Specifies a legacy import file which contains unencrypted passwords.

The parameters are mandatory when importing: Site, CGate, SGate and Trading Partner objects, and should not be used otherwise.

Note: `-import_pwd_cleartext` and `-ekf` are mutually exclusive.

`-site (-si)` : Enter this parameter without entering a value to import all Site objects contained in the file specified in the `input` parameter.

`-lsite (-ls)` : Enter this parameter without entering a value to import all Logical Site objects contained in the file specified in the `input` parameter.

Note: It is not recommended to use Logical Sites. Logical Sites are still supported in this version of Gateway but are no longer accessible via the GUI.

`-appli (-ap)` : Enter this parameter without entering a value to import all Application objects contained in the file specified in the `input` parameter.

`-user (-user)` : Enter this parameter without entering a value to import all User objects contained in the file specified in the `input` parameter.

`-profile (-pr)` : Enter this parameter without entering a value to import all Profile objects contained in the file specified in the `input` parameter.

`-list (-li)` : Enter this parameter without entering a value to import all Diffusion List objects contained in the file specified in the `input` parameter.

`-model (-ml)` : Enter this parameter without entering a value to import all Model objects contained in the file specified in the `input` parameter.

`-proplist (-pl)` : Enter this parameter without entering a value to import all Property List objects contained in the file specified in the `input` parameter.

`-purgemodel (-pm)` : Enter this parameter without entering a value to import all Purge Model objects contained in the file specified in the `input` parameter.

`-cgategroup (-cgg)` : Enter this parameter without entering a value to import all CGateGroup objects contained in the file specified in the `input` parameter.

Note: You must import CGateGroup objects *before* CGate objects.

`-cgate (-cg)` : Enter this parameter without entering a value to import all CGate objects contained in the file specified in the `input` parameter.

Note: You must import CGateGroup objects *before* CGate objects.

`-decisionrule (-rd)` : Enter this parameter without entering a value to import all Decision Rule objects contained in the file specified in the `input` parameter.

`-ruletable (-rt)` : Enter this parameter without entering a value to import all Rule Table objects contained in the file specified in the `input` parameter.

`-proxy (-px)` : Enter this parameter without entering a value to import all Proxy objects contained in the file specified in the `input` parameter.

`-tradeport (-tp)` : Enter this parameter without entering a value to import all Trading Partner objects contained in the file specified in the `input` parameter.

`-xmsector (-xc)` : Enter this parameter without entering a value to import all Axway Messaging connector objects contained in the file specified in the `input` parameter.

`-sgategroup (-sgg)` : Enter this parameter without entering a value to import all SGateGroup objects contained in the file specified in the `input` parameter.

Note: You must import SGateGroup objects *before* SGate objects.

`-sgate (-sg)` : Enter this parameter without entering a value to import all SGate objects contained in the file specified in the `input` parameter.

Note: You must import SGateGroup objects *before* SGate objects.

`-error_free`: Enter this parameter to create the different objects while ignoring the new parameters and without stopping the process at each error.

Example

Enter the following command to import the object contents of the `ImpObject` file:

```
pelbase import
-if ImpObject
-ekf <decryption_key.dat>
-site -lsite -model -appli -list
-user -profile -cgategroup -cgate -
error_free
```

Axway Gateway: Managing the Server

Upgrading the Mailbox

[Upgrading the Mailbox: pelmig](#)

The `pelmig` command enables you to upgrade the Mailbox file. The only supported migration is from 6.16 to 6.17. For more information, read [Upgrading to 6.17](#).

Upgrading the Mailbox: pelmig

Syntax

```
pelmig
{-input_version (-iv) { V6.16 | V6.17 }}
{-output_version (-ov) { V6.16 | V6.17 }}
```

Description	Use the pelmig command to upgrade the Mailbox file between different versions of Gateway. Notes: • Before upgrading the Mailbox file, you must first upgrade all Gateway objects (Sites, Applications, etc.). • The pelmig command requires that both the input and output Mailbox files are located in the <code>run_time/data</code> directory of the new version of Gateway. For this reason, it is recommended that you copy the input Mailbox to that directory and then run the pelmig command from that directory.
Parameters	<code>-input_version (-iv)</code> : Gateway version number from which you are upgrading (must be V6.16). <code>-output_version (-ov)</code> : Gateway version number to which you are upgrading (must be V6.17).
Example	To upgrade the Mailbox file from Gateway V6.16 to Gateway V6.17: 1. Copy and rename the file <code>xfer.dat</code> from Gateway V6.16 to Gateway V6.17 (name this file <code>xfer616.dat</code>). 2. Change directory to the <code>run_time/data</code> directory of Gateway V6.17. 3. Run the appropriate pelmig command. On a UNIX system, use the following command: <pre>cd <V617> . run_time/etc/profile cd run_time/data cp <V616>/run_time/data/xfer.dat xfer616.dat pelmig -im xfer616.dat -iv V6.16 -om xfer.dat -ov V6.17</pre>

Upgrading to an earlier version of Gateway

Although it is not recommended, you can use the **pelmig** command to upgrade the Mailbox file to an earlier version of Gateway. To make this possible, all Transfer Requests for upgrade must only use features that are available in both versions of Gateway.

By default, the **pelmig** command aborts the upgrade process if a Transfer Request that cannot upgrade is detected in the input Mailbox file. To change the default behavior, specify `-upgrade_mode partial` as input parameter. In partial mode, the **pelmig** command only upgrades Transfer Requests that are compatible with the target version of Gateway.

Message and exit status after upgrade

If upgrade is successful, the `pelmig` command displays the following statistical information in the output and then exits with a status value of zero:

- Total number of records successfully upgraded
- Total number of upgraded records changed by the user exit
- Total number of records not upgraded

Example output:

- 19 records upgraded
- 0 records updated
- 0 records ignored

Additional messages may also be written in the output depending on the value of the `verbosity` parameter supplied to `pelmig`. For example, when `verbosity` is either middle or high, the output also displays messages relating to changes applied to input data records or to information that was not moved to the upgraded records.

If an error occurs during upgrade, the `pelmig` command displays the cause of the error in the output and exits with a non-zero status.

For example, if the input Mailbox name cannot be opened for input because the specified name does not match an existing file, the following message will be displayed:

```
--2011/03/13 12:07:38 - ERROR from MigMailbox
--2011/03/13 12:07:38 - MbxOpen(xfer.dat) for reading returns error
--2011/03/13 12:07:38 - No such file or directory
--2011/03/13 12:07:38 - MAILBOX MIGRATION FAILED!
```

Axway Gateway: Managing the Server

Virtual Sites

Concerning virtual sites, please note that:

- Gateway 6.17.3 does not provide for the upgrading of virtual sites.
- `vsite03.dat` is automatically recreated.

[Importing and exporting objects](#)

Axway Gateway: Managing the Server

Executing server traces

The following online commands enable you to create trace files of Gateway activities without stopping and restarting the server:

- [pelctl trace entity](#)
- [pelctl trace process](#)

pelctl trace_entity

Syntax	<code>pelctl trace_entity {-entity_name} {-trace_level}</code>
Description	This command generates traces of a specified entity and records them to a file in <code><Gateway installation directory>/run_time/tmp</code>. This command initiates tracing dynamically. You do not need to restart the server for tracing to start. The name that Gateway attributes to the resulting file has the format: <pre>process_name-yyyyddmm-hhmmss-process_pid].out</pre> where <code>process_name</code> is the name of the Process that contains the entity in structure of the configuration file conffile located in <code><Gateway installation directory>/run_time/etc</code>.
Parameters	<i>(mandatory)</i> <code>-entity_name (-ename)</code>: Enter the name of the Gateway entity you want to trace. The entities that you can trace are listed in the configuration file conffile located in <code><Gateway installation directory>/run_time/etc</code>. <i>(mandatory)</i> <code>-trace_level (-l)</code>: Enter one of the following values: • 0: no trace • 1 • 2 • 3 • 4: maximum detail

Example

Enter the command:

```
pelctl trace_entity -ename NDSP -l 4
```

Gateway creates the trace file `sup-<timestamp>-<pid>.out` in the Gateway `/tmp` directory and records maximum trace detail of NDSP entity activities.

pelctl trace_process

Syntax

```
pelctl trace_process {-process_name} {-trace_level} [-all]
```

Description

This command generates traces of a specified Gateway process and records them to a file in `<Gateway installation directory>/run_time/tmp`.

This command initiates tracing dynamically. You do not need to restart the server for tracing to start.

The name that Gateway attributes to the resulting file has the format:

```
[process_name-yyyyymmdd-hhmmss-process_pid].out
```

Parameters

(mandatory)

`-process_name (-pname)` : Enter the name of the Gateway process you want to trace (for example, PEL, SUP, TCPPEL, ...).

The processes that you can trace are listed in the configuration file `conf file` located in `<Gateway installation directory>/run_time/etc`.

(mandatory)

`-trace_level (-l)` : Enter one of the following values:

- **0**: no trace
- **1**
- **2**
- **3**
- **4**: maximum detail

(optional)

`-all`: Enter this parameter to trace the process specified in the `-process_name` parameter and all entities that are associated with that process.

Example

Enter the command:

```
pelctl trace_process -pname SUP -l 4
```

Gateway creates the trace file `sup-<timestamp>-<pid>.out` in `<Gateway installation directory>/run_time/tmp` and records maximum trace details of SUP process activities.

Axway Gateway: Managing the Server

Managing rotation and archiving of trace files

This version of Gateway archives the trace files at regular intervals to prevent them becoming too large. This is achieved using *trace rotation* and *trace archiving* (also known as *trace collection*).

- **Trace rotation** is the act of closing the old trace files and creating the new trace files.
- **Trace archiving (or "collection")** is the act of moving the rotated (old) trace files into a configured archive folder.

Both of these processes are performed automatically by Gateway. You can modify the default parameters depending on your requirements.

You can manually trigger trace file rotation at any time. Gateway will move the old files into the configured archive directory without the need to restart.

Modifying trace rotation parameters

The parameters used for managing the trace rotation operation are stored in the `dtrace` section of the `dconfsave.ini` configuration file.

These parameters include:

Parameter	Description
<code>check_interval</code>	Determines the responsiveness of the trace rotation functionality. This is the maximum time (in seconds) on which a rotation is triggered after the rotation command was issued. Default = 300.
<code>check_file_path</code>	The directory containing an internal file used to manage the trace rotation operation. Default = <code>\$p_database</code> .

Parameter	Description
<code>check_file_name</code>	<i>For internal use only</i> The filename of the file used to signal trace rotation. Its last update timestamp is used to check if a rotation needs to be scheduled. Default = <code>dtrace.chk</code>.
<code>indirection_file_name</code>	<i>For internal use only</i> The full path to the indirection file used for saving the information related to trace files. It is used during trace rotation and collection to determine which files need to be archived. The default values is <code>\$p_database/indirection.ref</code>.

You can update these parameters using the `peluconf` online command. You do not need to restart the server to apply the parameters.

Example

```
peluconf [standalone] set -s dtrace check_interval 600
```

Modifying trace archiving parameters

You can use the following parameters in the `conf file` to configure the collector task which manages the archiving of trace files at a pre-scheduled interval.

Parameter	Description
<code>max_files_to_move</code>	The maximum number of files to be moved per iteration during a trace file collect operation. This should be a fraction of the <code>max_files_to_collect</code> parameter. Default = 10. (This parameter is designed to improve performance)
<code>max_files_to_collect</code>	The maximum number of trace files to be considered for a collection per iteration during preparation of the archive operation. Default = 100. (This parameter is designed to improve performance)
<code>collect_interval</code>	The interval (in seconds) between two successive trace file collector runs. Default = 600.

Parameter	Description
archive_ path	The directory where the archived traces are to be moved. Default = %p_home_dir%\run_time\tmp\archive. You can modify this value if required by editing the pelsetup.def file.

Triggering trace rotation

Use the `pelctl trace_rotate` command to trigger the rotation of trace files.

Syntax	<code>pelctl trace_rotate</code>
Description	Triggers the rotation of the trace files generated by Gateway. The rotated traces will be moved into a configurable directory. This command initiates trace rotation dynamically. You do not need to restart the server for the operation to start. Gateway creates the trace filenames in the format: <code>[process_name-yyyyymmdd-hhmmss-process_pid].out</code> where: <code>process_name</code> is the name of the process <code>yyyyymmdd-hhmmss</code> is the timestamp when the trace file was created <code>process_pid</code> is the pid of the process that created the trace file
Parameters	None
Example	<pre>pelctl trace_rotate</pre> Gateway moves the existing trace files into the archive directory.

Axway Gateway: Managing the Server

About gatestatus messages

[GateController and gatestatus](#)

[Possible gatestatus responses](#)

[List of gatestatus messages](#)

[Resetting the status prompted by the `gatestatus` command to the default value](#)

GateController and gatestatus

Gateway is designed as a modular system of loosely-coupled processes. The GateController process collects heartbeats and status messages from each of these individual processes. Despite being delivered to the operator through the `gatestatus` command, most of the messages GateController relays have nothing to do with GateController itself.

`gatestatus` is not the best way to monitor Gateway. It is a simple reference tool for a system administrator to check the functioning of Gateway. `gatestatus` messages do not necessarily indicate that there is problem, merely a condition that can only be cleared with user intervention.

Possible gatestatus responses

There are four classes of message that `gatestatus` can return:

- Gateway stopped
- Gateway running
- Gateway restarting
- Process crashed

These messages display the most recent status change for which GateController received a notification. To obtain more complete status information, including the time and date of the event, you should refer to the Gateway logs.

The best way to monitor Gateway is by tracking it in Sentinel. You can then automatically extract messages of interest and create alerts when something is wrong.

List of gatestatus messages

Process crash messages

Message	Printing Gateway status: Warning. TSD process raised a problem to GateController
Description	A dynamic Gateway process monitored by the TSD crashed or does not answer any more. It will be managed by its parent process.

Additional information	In this case, one of the child processes of the p_tsd process had to be restarted. The actions were all managed by the TSD module, and simply reported to GateController. Unless this happens frequently, it is harmless. One possible cause is insufficient resources due to system or user limits (ulimit). Gateway should resolve this issue on its own and automatically clear this message. For more details about the problem, look in <Gateway installation directory>/run_time/tmp/TSD-<timestamp>-<pid>.out
Message	Printing Gateway status: Minor error. TSD process raised a problem to GateController
Description	A dynamic Gateway process monitored by the TSD crashed or does not answer any more. It will be managed by its parent process.
Additional information	Gateway should resolve this issue on its own and automatically clear this message. For more details about the problem, look in <Gateway installation directory>/run_time/tmp/TSD-<timestamp>-<pid>.out
Message	Printing Gateway status: Minor error. SUP process raised a problem to GateController
Description	A dynamic Gateway process monitored by the SUP crashed or does not answer any more. It will be managed by its parent process.
Additional information	Gateway should resolve this issue on its own and automatically clear this message. For more details about the problem, look in <Gateway installation directory>/run_time/tmp/SUP-<timestamp>-<pid>.out
Message	Printing Gateway status: Minor error. TSD process raised a problem to GateController
Description	A dynamic Gateway process monitored by the TSD crashed or does not answer any more. It will be managed by its parent process. For more information refer to the log.
Additional information	Gateway should resolve this issue on its own and automatically clear this message. For more details about the problem, look in <Gateway installation directory>/run_time/tmp/TSD-<timestamp>-<pid>.out

Unable to retrieve messages

Message	Printing Gateway status Example: <pre>-- 2009/01/24 15:17:32 - ERROR from apigc_cli_build_cli_socket 2009/01/24 15:17:32 apigc_net_connect: SOCKET CONNECTION FAILED -- 2009/01/24 15:17:32 - ERROR from apigc_cli_init 2009/01/24 15:17:32 Client socket creation + connection failed. Retries are coming. -- 2009/01/24 15:17:35 - ERROR from apigc_cli_build_cli_socket 2009/01/24 15:17:35 apigc_net_connect: SOCKET CONNECTION FAILED 2009/01/24 15:17:39 apigc_net_connect: SOCKET CONNECTION FAILED 2009/01/24 15:17:44 apigc_net_connect: SOCKET CONNECTION FAILED 2009/01/24 15:17:48 apigc_net_connect: SOCKET CONNECTION FAILED 2009/01/24 15:17:52 apigc_net_connect: SOCKET CONNECTION FAILED GateController problem.</pre>
Description	GateController does not answer any more, but Gateway is still running.
Additional information	GateController is not running, but the pid file has not been cleaned up. Check for files named core in the Gateway and system core directories. Delete the pid file and attempt to restart Gateway.
Message	Printing Gateway status: GateController problem
Description	GateController does not answer any more, but Gateway is still running.
Additional information	GateController is not running, but Gateway is probably still trading. Restart Gateway when convenient.

Resetting the status prompted by the `gatestatus` command to the default value

The `pelmon reset` command enables you to reset the status prompted by the `gatestatus` command to the default value.

After entering this command, Gateway returns the message: **Printing Gateway status: running.**

Axway Gateway: Managing the Server

Reloading the profile

If you need to reload the environment variables for the operating system, you must reload the profile.

Depending on your operating system perform one of the following actions:

UNIX environments

1. Go to the directory: `<Gateway installation directory>/run_time/etc`
2. Execute the script: `profile`

Windows environments

1. Go to the directory: `<Gateway installation directory>\run_time\etc`
2. Execute the script: `profile.bat`

Axway Gateway: Messages and codes

Messages and codes

Gateway generates log messages that provide information about the processes and events that occur on the Gateway.

Structure of log messages

Each message is made up of three parts, each part displayed in a separate column:

Column	Contents
1	Time stamp (when the message was printed)
2	Message identification
3	Message text

The message identification items are displayed in two formats, depending on the user interface used. However, the message text does not change.

Message identification (column 2)

Messages are identified in a character string that comprises the following elements:

Element	Contents
Message number	Expressed using three bytes

Element	Contents
Gateway module (or process)	The name of the Gateway module or process at the source of this message. This name is abbreviated to one of: • FPSIT (PeSIT protocol) • FTP (FTP protocol) • HTTP (HTTP protocol) • SMTP (SMTP protocol) • POP3 (POP3 protocol) • SFTP (SFTP protocol) • ODT (OFTP protocol) • PEL (PEL protocol) • JCT (Job Control Table) • NET (Network module) • SNAC (SNA LU 6.2 module) • SUP (Supervisor module) • PCNX (Protocol connection - customization through the <code>cgate_log_level</code> configuration variable) • TSD (TCP Server Daemon) • VFD (Virtual file and directory access) • SECS (Security errors) • EXT (Exit) • USR (User) • CHK (Monitor checking) • NOT (Notif) • SR (Statistics and script resolution modules) • INT (Payload or Data integrity)
Message Severity	• S = Success • I = Information • W = Warning • E = Error
Event Name	A character string that abbreviates the name of the associated event. There is a one-to-one correspondence between message names and code numbers.

In the GUI, the Gateway module name, the code number and the type of message are concatenated. The message name is not displayed.

Example:

Depending on the interface, PEL information message number 131, named CONN_REQ, is displayed in one of the following formats:

- <time stamp> 131 PEL I CONN_REQ <text>
- <time stamp> PEL131I <text>

Message text (column 3)

The text of the message comprises a generic component. In turn, this component contains a (variable) number of context-dependent parameters.

Examples of message text

	Code	Module	Type	Message name	Message text
048	SUP	I	KEY		key=%1, site=%2
101	NET	I	CONN_REQ		(%1) outgoing connection request [dest_address="%2"]
045	PEL	E	ABORT_IND		%1(%2) [%3] transfer aborted. reason=%4

Each instance of a message replaces the context-dependent variables by their values before writing the message in the log file. In the example messages shown in the table above, variables are designated by the percentage symbol (%), followed by a digit that represents the rank of the parameter in the message.

Axway Gateway: Messages and codes

Access Management (AM)

This topic lists Access Management (AM) log messages.

101	AM I TRACE %1
<i>Type</i>	Information

Meaning	Generic message	
Parameters	1	Not used
Action	None	
102	AM W TRACE %1	
Type	Warning	
Meaning	Warning message	
Parameters	1	Incompatibility between CSD versions: "Gateway uses '%', Passport AM uses '%".
Action	Make sure that Gateway and Passport AM use the same CSD file version. Otherwise some inconsistencies may occur.	
103	AM E TRACE %1	
Type	Error	
Meaning	Generic error message	

<i>Parameters</i>	1	<i>Errors generated by the C API (XPAM), used for the communication with Passport AM or by Passport AM server:</i> • Error while initializing access management service: '%am_error%' • Error while checking user '%' authentication rights '%am_error%' • Error while checking user '%' authorization '%am_error%' • XPAM action '%' failed Note: If %am_error% is 'XPAM_ERR_SERVER', an additional log might be displayed. For more details regarding this message from PassPort, consult the PassPort documentation. <i>Error generated by: wrong username or password, user disabled or not defined in Passport AM server:</i> • User '%' authentication denied <i>Error generated by insufficient rights defined in Passport AM server for given user:</i> • User '%' is not authorized for the privilege: resource '%', action '%' <i>Internal Gateway errors:</i> • Internal error occurrence
<i>Action</i>		Activate traces for more information. If you cannot correct the problem or in case of internal errors (from Gateway, XPAM or Passport AM server), contact Axway Support.
104	AM I CONOK %1	
<i>Type</i>		Information
<i>Meaning</i>		Generic message
<i>Parameters</i>	1	Connection established with AM server
<i>Action</i>		None

Axway Gateway: Messages and codes

EDI errors

This topic lists EDI error log messages.

1000	Trading partner does not exist
<i>Action</i>	Enter a valid trading partner identifier (name & format).
1001	Originator trading partner does not exist
<i>Action</i>	Enter a valid trading partner identifier (name & format).
1002	Destination trading partner does not exist
<i>Action</i>	Enter a valid trading partner identifier (name & format).
1003	Ambiguous trading partner
<i>Meaning</i>	Trading format was not specified and more than one partner corresponds to the given name.
<i>Action</i>	Specify the trading format.
1004	Trading partner not local
<i>Action</i>	
1005	Invalid ciphering certificate
<i>Action</i>	Check trading partner configuration.
1006	Invalid signing certificate
<i>Action</i>	Check trading partner configuration.
1007	Invalid trading state of xfer
<i>Action</i>	None

1008	Invalid trading file format
<i>Action</i>	None
1009	Invalid MIC
<i>Action</i>	None
1010	Invalid acknowledgment URL
<i>Action</i>	None
1011	Invalid security level
<i>Action</i>	Check trading partner configuration.
1012	Can't send reply, receipt not requested
<i>Action</i>	None
1013	SMIME operation aborted
<i>Action</i>	

Axway Gateway: Messages and codes

Exits (EXT)

These EXIT messages are displayed only if the following line is present in the SUP entity definition in the Gateway configuration file:

```
trncc <infile>
```

001	EXT I XFER_RQ %1(%2) [%3]: %4

<i>Type</i>	Information	
<i>Meaning</i>	The transfer exit function labeled %4 has been called and is in progress	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Exit name
<i>Action</i>	None	

002 EXT I XFER_CF %1(%2) [%3]: %4 return code = %5

<i>Type</i>	Information	
<i>Meaning</i>	The transfer exit function labeled %4 has been executed and has returned the code number %5	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Exit name
	5	Return code. If 0 , the transfer continues normally, otherwise it is canceled.
<i>Action</i>	None	

003 EXT I XFER_ER %1(%2) [%3]: %4 %5 error = %6

<i>Type</i>	Information	
<i>Meaning</i>	The transfer exit function labeled %4 has returned an incorrect value for one of the transfer parameters it received in its arguments (field of ExitXfer_t structure). The function has supplied an incorrect value for the transfer parameter identified by %6.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Exit name
	5	Error label
	6	Identification of the incorrect field
Action	Check the source code of the exit concerned	
004	EXT I CONN_RQ %1(%2) [%3]: %4	
Type	Information	
Meaning	The connection exit function labeled %4 has been called and is in progress	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Exit name
Action	None	
005	EXT I CONN_CF %1(%2) [%3]: %4 return code = %5	
Type	Information	
Meaning	The connection exit function labeled %4 has been executed and has returned the code number %5	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Exit name
	5	Return code. If 0 , the transfer continues normally, otherwise it is canceled.
<i>Action</i>	None	

006	EXT I CONN_ER %1(%2) [%3]: %4 %5 error = %6	
<i>Type</i>	Information	
<i>Meaning</i>	The connection exit function labeled %4 has set an incorrect value for one of the transfer parameters (fields of ExitConn_t structure) it received in its arguments. The function has supplied an incorrect value for the parameter identified by %6.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Exit name
	5	Error label
	6	Identification of the incorrect field
<i>Action</i>	Check the exit source code	

Axway Gateway: Messages and codes

File Transfer Protocol (FTP)

This topic lists FTP log messages.

100	FTP E ERROR %1 (%2) %3	
<i>Type</i>	Error	

<i>Meaning</i>	An error condition has been detected. This message is used to report system errors (Input/Output, memory, network) or errors that cannot be explained by the protocol diagnostic code set.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Error text
<i>Action</i>	Refer to your platform documentation for system errors. For other errors, check protocol or Application parameters.	

101 FTP I TRACE %1

<i>Type</i>	Information	
<i>Meaning</i>	This message is used to trace FTP commands or FTP result codes sent to, or received from, a Remote Site. Each FTP command is in hexadecimal and text format (ASCII or EBCDIC, depending on host language).	
<i>Parameters</i>	1	FTP command or FTP result code
<i>Action</i>	None	

102 FTP I TRC_DATA %1

<i>Type</i>	Information	
<i>Meaning</i>	This message is used to trace data file or data list exchange with a Remote Site. All data is in hexadecimal and in text format (ASCII or EBCDIC, depending on host language if possible).	
<i>Parameters</i>	1	Data exchanged
<i>Action</i>	None	

103 FTP W TIMEOUT %1 (%2)

<i>Type</i>	Warning	
<i>Meaning</i>	This message appears if timeout occurred for a specific connection.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	None	

104	FTP E ERROR Ftp script file %1 doesnt exist or cannot be opened	
<i>Type</i>	Error	
<i>Meaning</i>	This message appears if a user uses a script file to send specific <code>SITE</code> or <code>QUOTE</code> commands in Initiator mode. For example: <code>SCRIPT</code> file in user message 1 from Transfer Request or Application.	
<i>Parameters</i>	1	Name of the script file
<i>Action</i>	Abort transfer, if needed, and re-send with a correct script file name. Alternatively, change read access on file.	

105	FTP E ERROR Command and %1 isnt correct, parsing of command line %2 canceled	
<i>Type</i>	Error	
<i>Meaning</i>	This message appears if timeout occurred for a specific connection.	
<i>Parameters</i>	1	Command in error
	2	Full user message or line in script
<i>Action</i>	Abort transfer, if needed, and restart the transfer with a correct user message 1 or line in script file.	

106	FTP I INFO Command '%1' ignored, not allowed to user	
<i>Type</i>	Information	
<i>Meaning</i>	Received an unknown FTP command.	
<i>Parameters</i>	1	Command in error
<i>Action</i>	Do not use this command any more.	

107	FTP E ERROR %1: data length %2 exceed %3	
Type	Error	
Meaning	An encoding/decoding error occurred during transfer.	
Parameters	1	Type of FTP encoding/decoding processing
	2	Length in error
	3	Maximum length as described in Transfer Request (or in Application used)
Action	Check data length in transfer or in Application.	
108	FTP E ERROR %1 parameter too long: '%2'	
Type	Error	
Meaning		
Parameters	1	Type of FTP encoding/decoding processing
	2	Length in error
Action		
109	FTP E ERROR Received file name '%1' is too long	
Type	Error	
Meaning		
Parameters	1	
Action		
110	FTP E ERROR %1(%2) Data connection forbidden: remote IP addresses are different for control and data sessions	
Type	Error	

<i>Meaning</i>	A bounce attack was detected. The data connection is restricted.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	If the requested transfer is not a bounce attack, enable the <code>change_data_addr</code> option in the Remote Site or the Proxy object to allow the transfer.	
111	FTP W TIMEOUT (%1) %2 Login deadline expired	
<i>Type</i>	Warning	
<i>Meaning</i>	Gateway has cut a connection from a partner, that was not followed by a completed login within a reasonable timeout. This mechanism avoids DOS-type attacks, that attempt to keep one connection slot occupied without proceeding to log in.	
<i>Parameters</i>	1	Remote alias
	2	Connection identifier (number)
<i>Possible causes</i>	• DOS attack • Network delays	
<i>Actions</i>	• Monitor calling addresses for the "login timed out" connections, check for repeated connection attempts not followed by login attempts. • Check network bandwidth.	
120	FTP W SEC_FAIL %1(%2) Security negotiation failure: security is disabled for control session	
<i>Type</i>	Warning	
<i>Meaning</i>	An AUTH command was received while security is restricted on the corresponding session.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	Check the configuration of the FTP client that sent this command, or configure Gateway to allow security.	

121	FTP W SEC_FAIL %1(%2) Security negotiation failure: security is mandatory for control session	
<i>Type</i>	Warning	
<i>Meaning</i>	The FTP client tried to connect in an unsecured way on a session where security is mandatory.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	Check the configuration of the FTP client that sent this command, or configure Gateway to allow unsecured communication.	
122	FTP W SEC_FAIL %1(%2) Security negotiation failure: security is disabled for data session	
<i>Type</i>	Warning	
<i>Meaning</i>	In server mode, the FTP client tried to negotiate a secured data session while this is restricted by configuration.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	Check the configuration of the FTP client that negotiated this feature, or configure Gateway to allow a secured transfer.	
123	FTP W SEC_FAIL %1(%2) Security negotiation failure: security is mandatory for data session	
<i>Type</i>	Warning	
<i>Meaning</i>	In server mode, the FTP client tried to negotiate an unsecured data session while security is required by configuration.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	Check the configuration of the FTP client that negotiated this feature, or configure Gateway to allow an unsecured transfer.	

124	FTP W SEC_FAIL %1(%2) Security negotiation failure: server refused '%3'	
<i>Type</i>	Warning	
<i>Meaning</i>	In client mode, the FTP server rejected a PROT command sent by Gateway, thus restricting the corresponding level of security for the next transfers. If the data policy of the Security Profile used is set to OPTIONAL, another option may be negotiated with the server.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	The rejected command: PROT P (secured transfer) or PROT C (plain transfer)
<i>Action</i>	If the negative server response is the last one of the negotiation, check the configuration of the FTP server that rejected the command, or configure Gateway to be compatible with its policy.	
130	FTP W VFD_ERR %1(%2) Could not move '%3' to '%4': '%5'	
<i>Type</i>	Warning	
<i>Meaning</i>	In server mode, an error occurred while trying to execute a requested move/rename sequence (RNFR-RNTO).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Source path
	4	Destination path
	5	Error message
<i>Action</i>	The action must be taken according to the error message.	
131	FTP W VFD_ERR %1(%2) Could not create directory '%3': '%4'	
<i>Type</i>	Warning	

<i>Meaning</i>	In server mode, an error occurred while trying to execute a requested directory creation command (MKD).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Directory
	4	Error message
<i>Action</i>	The action must be taken according to the error message.	

132	FTP W VFD_ERR %1(%2) Could not delete file '%3': '%4'	
<i>Type</i>	Warning	
<i>Meaning</i>	In server mode, an error occurred while trying to execute a requested file deletion command (DELE).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Directory
	4	Error message
<i>Action</i>	The action must be taken according to the error message.	

133	FTP W VFD_ERR %1(%2) Could not remove directory '%3': '%4'	
<i>Type</i>	Warning	
<i>Meaning</i>	In server mode, an error occurred while trying to execute a requested directory deletion command (RMD).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Directory
	4	Error message
<i>Action</i>	The action must be taken according to the error message.	

134	FTP W VFD_ERR %1(%2) Write permission denied for file '%3'. Current directory is '%4'	
Type	Warning	
Meaning	Gateway, in server mode, has refused the user, connected FTP client, or other, the right to receive the requested file in the defined directory.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	File name
	4	Directory
Action	Check that the user, the connected FTP client, or other, is authorized to upload this file. If the user is authorized, check the VFD rights defined in the CGate for the user, the connected FTP client, or other.	

135	FTP W VFD_ERR %1(%2) Read permission denied for file '%3'. Current directory is '%4'	
Type	Warning	
Meaning	Gateway, in server mode, has refused the user, connected FTP client, or other, the right to send the requested file to the defined directory.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	File name
	4	Directory
Action	Check that the user, the connected FTP client, or other is authorized to download this file. If the user is authorized, check the VFD rights defined in the CGate for this user, the connected FTP client, or other.	

210	FTP I SEND_BEG %1 (%2) [%3] Start sending ftp file: %4	
Type	Information	

<i>Meaning</i>	Beginning of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Xfer local ident
	4	Protocol file name
<i>Action</i>	None	

211 FTP I SEND_BEG %1 (%2) [%3] Start sending file: %4

<i>Type</i>	Information	
<i>Meaning</i>	Beginning of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
<i>Action</i>	None	

212 FTP I SEND_RST %1 (%2) [%3] Restart sending ftp file: %4

<i>Type</i>	Information	
<i>Meaning</i>	Restart of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
<i>Action</i>	None	

213	FTP I SEND_RST %1 (%2) [%3] Restart sending file: %4	
Type	Information	
Meaning	Restart of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
Action	None	
214	FTP I SEND_END %1 (%2) [%3] End sending ftp file: %4	
Type	Information	
Meaning	End of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
Action	None	
215	FTP I SEND_END %1 (%2) [%3] End sending file: %4	
Type	Information	
Meaning	End of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
<i>Action</i>	None	

216	FTP E SEND_ERR %1 (%2) [%3] Error sending (state %4) ftp file: %5	
<i>Type</i>	Error	
<i>Meaning</i>	Error on an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

217	FTP E SEND_ERR %1 (%2) [%3] Error sending (state %4) file: %5	
<i>Type</i>	Error	
<i>Meaning</i>	Error on an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Real file name on file system
<i>Action</i>	None	

218	FTP W SEND_STP %1 (%2) [%3] Stop sending (state %4) ftp file: %5	
<i>Type</i>	Warning	
<i>Meaning</i>	End of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

219	FTP W SEND_STP %1 (%2) [%3] Stop sending (state %4) file: %5	
<i>Type</i>	Warning	
<i>Meaning</i>	End of an FTP data file transfer in Initiator/Sender or Responder/Sender mode.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

230	FTP I RECV_BEG %1 (%2) [%3] Start receiving ftp file: %4	
<i>Type</i>	Information	
<i>Meaning</i>	Beginning of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
<i>Action</i>	None	

231	FTP I RECV_BEG %1 (%2) [%3] Start receiving file: %4	
<i>Type</i>	Information	
<i>Meaning</i>	Beginning of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
<i>Action</i>	None	

232	FTP I RECV_RST %1 (%2) [%3] Restart receiving ftp file: %4	
Type	Information	
Meaning	Restart of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
Action	None	
233	FTP I RECV_RST %1 (%2) [%3] Restart receiving file: %4	
Type	Information	
Meaning	Restart of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
Action	None	
234	FTP I RECV_RST %1 (%2) [%3] Restart receiving file: %4	
Type	Information	
Meaning	End of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
<i>Action</i>	None	

235	FTP I RECV_END %1 (%2) [%3] End receiving file: %4	
<i>Type</i>	Information	
<i>Meaning</i>	End of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
<i>Action</i>	None	

236	FTP E RECV_ERR %1 (%2) [%3] Error receiving (state %4) ftp file: %5	
<i>Type</i>	Error	
<i>Meaning</i>	End of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

237	FTP E RECV_ERR %1 (%2) [%3] Error receiving (state %4) file: %5	
Type	Error	
Meaning	Error on an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Real file name on file system
Action	None	
238	FTP W RECV_STP %1 (%2) [%3] Stop receiving (state %4) ftp file: %5	
Type	Warning	
Meaning	End of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
Action	None	
239	FTP W RECV_STP %1 (%2) [%3] Stop receiving (state %4) file: %5	

<i>Type</i>	Warning	
<i>Meaning</i>	End of an FTP data file transfer in Initiator/Receiver or Responder/Receiver mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

240 **FTP I XACK_REQ %1(%2) [%3] send file acknowledge, org:%4, dest:%5, protocol_file name:%6**

<i>Type</i>	Information	
<i>Meaning</i>	An FTP acknowledgment was sent.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol originator
	5	Protocol destination
	6	Protocol file name
<i>Action</i>	None	

241 **FTP I XACK_IND %1(%2) [%3] receive file acknowledge, org:%4, dest:%5, protocol_file name:%6**

<i>Type</i>	Information	
<i>Meaning</i>	An FTP acknowledgment was received.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol originator
	5	Protocol destination
	6	Protocol file name
<i>Action</i>	None	

242	FTP E CMD_ERR %1(%2) [%3] %4	
<i>Type</i>	Error	
<i>Meaning</i>	An error response (> 400) was received	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	The last command sent, followed by ":", and the error response received
<i>Action</i>	None	

250	FTP E VFD_ERR %1 (%2) VFD init error: %3 - %4	
<i>Type</i>	Error	
<i>Meaning</i>	An error occurred in FTP while initializing a VFD session	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	VFD error code
	4	VFD error message (refer to VFD Errors for more information)
<i>Action</i>	Depends on the VFD message	

260	FTP I LIST_BEG %1 (%2) [%3] Start receiving ftp listing: %4	
Type	Information	
Meaning	Beginning of an FTP listing.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
Action	None	
261	FTP I LIST_BEG %1 (%2) [%3] Start receiving listing: %4	
Type	Information	
Meaning	Beginning of an FTP listing.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Real file name on file system
Action	None	
264	FTP I LIST_END %1 (%2) [%3] End receiving ftp listing: %4	
Type	Information	
Meaning	End of an FTP listing.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
<i>Action</i>	None	

265	FTP I LIST_END %1 (%2) [%3] End receiving listing: %4	
<i>Type</i>	Information	
<i>Meaning</i>	End of an FTP listing.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Protocol file name
<i>Action</i>	None	

266	FTP E LIST_ERR %1 (%2) [%3] Error receiving (state %4) ftp listing: %5	
<i>Type</i>	Error	
<i>Meaning</i>	Error on an FTP listing.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

267	FTP E LIST_ERR %1 (%2) [%3] Error receiving (state %4) listing: %5	
Type	Error	
Meaning	Error on an FTP listing.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Real file name on file system
Action	None	
268	FTP W LIST_STP %1 (%2) [%3] Stop receiving (state %4) ftp listing: %5	
Type	Warning	
Meaning	End of an FTP listing	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
Action	None	
269	FTP W LIST_STP %1 (%2) [%3] Stop receiving (state %4) listing: %5	
Type	Warning	
Meaning	End of an FTP listing	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Transfer state
	5	Protocol file name
<i>Action</i>	None	

300	FTP I PX_LOAD %1(%2) Proxy %3 successfully loaded	
<i>Type</i>	Information	
<i>Meaning</i>	Proxy object has been found and loaded.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
<i>Action</i>	None	

301	FTP I PX_CNX %1(%2) Successfully connected to proxy %3	
<i>Type</i>	Information	
<i>Meaning</i>	A connection has been established between Gateway and proxy.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
<i>Action</i>	None	

310	FTP E PX_ERR %1(%2) Could not access proxy %3: error %4	

Type	Information	
Meaning	An error occurred while reading proxy object configuration.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
	4	Meaningless
Action		
311	FTP E PX_ERR %1(%2) Proxy %3 is not an FTP proxy	
Type	Information	
Meaning	The proxy object does not refer to FTP protocol.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
Action	Verify that the object used as proxy in the ftp site is defined with ftp protocol.	
312	FTP E PX_ERR %1(%2) Parsing error in proxy %3 script	
Type	Error	
Meaning	An error occurred while interpreting the script.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
Action	Correct command lines in the script.	

313	FTP E PX_ERR %1(%2) Protocol error while connecting to proxy %3: %4 %5	
Type	Error	
Meaning	Protocol error while connecting to proxy	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
	4	Error code
	5	Error message
Action	Verify protocol configuration.	
314	FTP E PX_ERR %1(%2) Proxy %3 script file %4 access error %5: %6	
Type	Error	
Meaning	Unable to access to script file name	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Proxy name
	4	Script file name
	5	Error code (errno)
	6	Error message (errno message)
Action	Verify name of script file.	
320	FTP I SEC_OK %1(%2) Data sessions mode switched to 'secure'	
Type	Information	
Meaning	Data session has been switched to secure mode.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	None	

321	FTP I SEC_OK %1(%2) Data sessions mode switched to 'clear'	
<i>Type</i>	Information	
<i>Meaning</i>	Data session has been switched to clear mode.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	None	

330	FTP I VFD_OK %1(%2) '%3' moved to '%4'	
<i>Type</i>	Information	
<i>Meaning</i>	A file name has been successfully renamed.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Old file name
	4	New file name
<i>Action</i>	None	

331	FTP I VFD_OK %1(%2) Created directory '%3'	
<i>Type</i>	Information	
<i>Meaning</i>	A directory has been successfully created.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Created directory name
<i>Action</i>	None	

332	FTP I VFD_OK %1(%2) Deleted file '%3'	
<i>Type</i>	Information	
<i>Meaning</i>	A file has been successfully deleted.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Deleted file name
<i>Action</i>	None	

333	FTP I VFD_OK %1(%2) Removed directory '%3'	
<i>Type</i>	Information	
<i>Meaning</i>	A directory has been successfully deleted.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Deleted directory name
<i>Action</i>	None	

Axway Gateway: Messages and codes

FTP reply codes

FTP code explanations

The RFC 959 reply codes are explained below:

1yz	Positive Preliminary reply The requested action is being initiated. Expect another reply before proceeding with a new command. (The user-process sending another command before the completion reply would violate protocol, but server-FTP processes should queue any commands that arrive while a preceding command is in progress.) This type of reply can be used to indicate that the command was accepted and the user-process may now recognize the data connections, for implementations where simultaneous monitoring is difficult. The server-FTP process can send at most, one 1yz reply per command.
2yz	Positive Completion reply The requested action has been successfully completed. A new request may be initiated.
3yz	Positive Intermediate reply The command has been accepted, but the requested action is suspended, pending receipt of further information. The user should send another command specifying this information. This reply is used in command sequence groups.
4yz	Transient Negative Completion reply The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again. The user should return to the beginning of the command sequence, if any. It is difficult to assign a meaning to transient, particularly when two distinct sites (Server- and User-processes) have to agree on the interpretation. Each reply in the 4yz category might have a slightly different time value, but the intent is that the user-process should try again. A rule of thumb in determining if a reply fits into the 4yz or the 5yz (Permanent Negative) category is that replies are 4yz if the commands can be repeated without any change in command form or in properties of the User or Server (for example, the command is spelled the same with the same arguments used; the user does not change his file access or user name; the server does not put up a new implementation).

5yz	Permanent Negative Completion reply The command was not accepted and the requested action did not take place. The User-process is discouraged from repeating the exact request (in the same sequence). Even some permanent error conditions can be corrected, so you may want to re-initiate the command sequence by direct action at some point in the future (for example, after the spelling has been changed, or the user has altered his directory status,...).

Second digit definitions

The following function groupings are encoded in the second digit:

x0z	Syntax These replies refer to syntax errors, syntactically correct commands that do not fit any functional category. They are un-implemented or superfluous commands.
x1z	Information These are replies to requests for information, such as status or help.
x2z	Connections Replies referring to the control and data connections.
x3z	Authentication and accounting Replies for the login process and accounting procedures.
x4z	File system These replies indicate the status of the Server file system in relation to the requested transfer or other file system action.

Third digit definitions

The third digit gives a finer gradation of meaning in each of the function categories.

FTP reply code list

The following is a list of FTP reply codes as defined in RFC 959, displayed in numeric order.

110	Restart marker reply

<i>Meaning</i>	In this case, the text is exact and not left to the particular implementation. It should read as follows: MARK yyyy = mmmm Where yyyy is User-process data stream marker, and mmmm server's equivalent marker (note the spaces between markers and "=").

120

<i>Meaning</i>	Service ready in nnn minutes.

125

<i>Meaning</i>	Data connection already open; transfer starting.

150

<i>Meaning</i>	File status ok; about to open data connection.

200

<i>Meaning</i>	Command is ok.

202

<i>Meaning</i>	Command not implemented, superfluous at this site.

211

<i>Meaning</i>	System status, or system help reply.

212

<i>Meaning</i>	Directory status.

213

<i>Meaning</i>	File status.

214

<i>Meaning</i>	Help message on how to use the server, or the meaning of a particular non-standard command. This reply is aimed at the user.

215

<i>Meaning</i>	NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.

220

<i>Meaning</i>	Service ready for new user.

221

<i>Meaning</i>	Service closing control connection. Logged out if appropriate.

225

<i>Meaning</i>	Data connection open; no transfer in progress.

226

<i>Meaning</i>	Closing data connection. Requested file action successful (for example, file transfer or file abort).

227

<i>Meaning</i>	Entering Passive Mode (h1,h2,h3,h4,p1,p2).

230

<i>Meaning</i>	User logged in, proceed.

250

<i>Meaning</i>	Requested file action ok, completed.

257

<i>Meaning</i>	"PATHNAME" created.

331

<i>Meaning</i>	User name ok, need password.

332

<i>Meaning</i>	Need account for login.

350

<i>Meaning</i>	Requested file action pending further information.

421

<i>Meaning</i>	Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.

425

<i>Meaning</i>	Cannot open data connection.

426

<i>Meaning</i>	Connection closed; transfer aborted.

450

<i>Meaning</i>	Requested file action not taken. File unavailable (for example, file busy).

451

<i>Meaning</i>	Requested action aborted: local error in processing.

452

<i>Meaning</i>	Requested action not taken. Insufficient storage space in system.

500

<i>Meaning</i>	Syntax error, command unrecognized. This may include errors such as 'the command line is too long'.

501

<i>Meaning</i>	Syntax error in parameters or arguments.

502

<i>Meaning</i>	Command not implemented.

503

<i>Meaning</i>	Bad sequence of commands.

504

<i>Meaning</i>	Command not implemented for that parameter.

530

<i>Meaning</i>	Not logged in.

550

<i>Meaning</i>	Requested action not taken. File unavailable (for example, file not found).

551

<i>Meaning</i>	Requested action aborted: page type unknown.

552

<i>Meaning</i>	Requested file action aborted. Exceeded storage allocation (for current directory or dataset).

553

<i>Meaning</i>	Requested action not taken. File name not allowed.

Command-reply correspondence table

COMMAND	SUCCESS	FAIL
USER	230,330	430-432,500-505
PASS	230	430-432,500-505
BYE	231,232	430-432,500-505
BYTE	200	500-506
SOCK	200	500-506
TYPE	200	500-506
MODE	200	500-506
RETR	250	450,451,500-506
Secondary Reply	252	452
STOR	250	451,451,500-506
Secondary Reply	252	452,453
APPE	250	451,451,500-506
Secondary Reply	252	452,453
RNFR	200	450,451,500-506
RNTD	253	450,451,500-505
DELE	254	450,451,500-506
LIST	250	450,453,500-506
Secondary Reply	252	452
ALLO	200	500-506
STAT	100,150,151	450,451,500-506
REST	200	500-506
ABOR	201,202	500-505

COMMAND	SUCCESS	FAIL
Spontaneous Replies	0xx,300,301, 251,255	400,401,434-436

Axway Gateway: Messages and codes

Hypertext Transfer Protocol (HTTP)

This topic lists HTTP log messages.

101	HTTP I TRACE %1
<i>Type</i>	Information
<i>Meaning</i>	This message is used to trace both received and sent data.
<i>Parameters</i>	None
<i>Action</i>	None

102	HTTP I TRACE %1(%2) %3 - %4 - %5 "%6" - length %7 - user-agent: %8
<i>Type</i>	Information
<i>Meaning</i>	An HTTP request has been received and is being processed. The request does not match a monitor transfer.

Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Request method used by the HTTP client (GET, PUT, POST)
	4	Path requested (URI accessed)
	5	HTTP return code
	6	Reason associated with the return code
	7	Length of the data sent by the client
	8	"User-Agent" field sent by the client. This field is typically used to recognize the kind of client connected.
Action	None	
103	HTTP I TRACE %1(%2) [%9] %3 - %4 - %5 "%6" - length %7 - user-agent: %8	
Type	Information	
Meaning	A Transfer Request has been received and is being processed.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Request method used by the HTTP client (GET, PUT, POST)
	4	Path requested (URI accessed)
	5	HTTP return code
	6	Reason associated with the return code
	7	Length of the data sent by the client
	8	"User-Agent" field sent by the client. This field is typically used to recognize the kind of client connected.
	9	Local transfer number
Action	None	

104	HTTP I TRACE (%1) %2 %3	
Type	Information	
Meaning	This message is used to trace received and sent HTTP headers.	
Parameters	1	Local connection identifier (number)
	2	Direction (R for read and S for sent)
	3	Header field
Action	None	
105	HTTP I TRACE HTTP %1"server is using identification method: %2	
Type	Information	
Meaning	This message is displayed when the server is started and each time the identification method is changed.	
Parameters	1	Server using the authentication method
	2	New authentication method
Action	None	
110	HTTP E ERROR %1(%2) Can't load file %3	
Type	Error	
Meaning	A file could not be loaded (configuration file or directory listing template file).	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	File path
Action	None	
111	HTTP E ERROR %1(%2) [%3] %4: error %5 (%6)	

Type	Error	
Meaning	An unexpected error occurred while manipulating a resource.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Current operation being made or resource being manipulated
	5	Error number or string
	6	Description (if available)
Action	Check configuration of the resource being used.	
112	HTTP E ERROR %1(%2) No directory listing template available	
Type	Error	
Meaning	Directory could not be listed because no listing template is defined.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
Action	Fill in the list template field of the selected CGate or the default list template in the HTTP configuration	
130	HTTP W VFD_ERR %1(%2) Could not move '%3' to '%4': '%5'	
Type	Warning	
Meaning	Directory could not be moved.	

<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Old directory name
	4	New directory name
	5	Error message
<i>Action</i>	Action depends on the error message.	

131	HTTP W VFD_ERR %1(%2) Could not create directory '%3': '%4'	
<i>Type</i>	Warning	
<i>Meaning</i>	Directory could not be created.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Directory name
	4	Error message
<i>Action</i>	Action depends on the error message.	

132	HTTP W VFD_ERR %1(%2) Could not delete '%3': '%4'	
<i>Type</i>	Warning	
<i>Meaning</i>	Directory could not be deleted.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Directory name
	4	Error message
<i>Action</i>	Action depends on the error message.	

133	HTTP W VFD_ERR %1(%2) Could not list '%3': '%4'	
Type	Warning	
Meaning	Directory could not be listed.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Directory name
	4	Error message
Action	Action depends on the error message.	
134	HTTP W VFD_ERR %1(%2) Could not read '%3': '%4'	
Type	Warning	
Meaning	Directory could not be read.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Directory name
	4	Error message
Action	Action depends on the error message.	
135	HTTP W VFD_ERR %1(%2) Write permission denied for file '%3'. Current directory is '%4'	
Type	Warning	
Meaning	File could not be written.	

<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	File name
	4	Directory message
<i>Action</i>	Check VFD access rights.	

150	HTTP I VFD_OK %1(%2) '%3' moved to '%4'	
<i>Type</i>	Information	
<i>Meaning</i>	Directory was moved.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Old directory name
	4	New directory name
<i>Action</i>	None	

151	HTTP I VFD_OK %1(%2) Created directory '%3'	
<i>Type</i>	Information	
<i>Meaning</i>	Directory was created.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Directory name
<i>Action</i>	None	

152	HTTP I VFD_OK %1(%2) Deleted file '%3'	

<i>Type</i>	Information	
<i>Meaning</i>	File was deleted.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	File name
<i>Action</i>	None	

153 HTTP I VFD_OK %1(%2) Deleted directory '%3'

<i>Type</i>	Information	
<i>Meaning</i>	Directory was deleted.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Directory name
<i>Action</i>	None	

211 HTTP I SEND_BEG %1(%2) [%3] %4 Start sending file: %5

<i>Type</i>	Information	
<i>Meaning</i>	The file is starting to be sent.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file being sent
<i>Action</i>	None	

215	HTTP I SEND_END %1(%2) [%3] %4 End sending file: %5	
Type	Information	
Meaning	The file has been completely sent.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file sent
Action	None	
217	HTTP E SEND_ERR %1(%2) [%3] %4 Error sending file: %5	
Type	Error	
Meaning	An error occurred while sending the file.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file being sent
Action	None	
231	HTTP I RECV_BEG %1(%2) [%3] %4 Start receiving file: %5	
Type	Information	
Meaning	A file starts to be received.	

<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file being received
<i>Action</i>	None	

235	HTTP I RECV_END %1(%2) [%3] %4 End receiving file: %5	
<i>Type</i>	Information	
<i>Meaning</i>	The file has been completely received.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file received
<i>Action</i>	None	

237	HTTP E RECV_ERR %1(%2) [%3] %4 Error receiving file: %5	
<i>Type</i>	Error	
<i>Meaning</i>	An error occurred while receiving the file.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file being received

<i>Action</i>	None

240	HTTP I LIST_BEG %1(%2) [%3] %4 Start receiving listing: %5
<i>Type</i>	Information
<i>Meaning</i>	Start of file reception.
<i>Parameters</i>	1 Connected Site alias
	2 Local connection identifier (number)
	3 Local number of the transfer (0 if not available)
	4 Protocol destination
	5 Physical path of the file being received
<i>Action</i>	None

241	HTTP I LIST_END %1(%2) [%3] %4 End receiving listing: %5
<i>Type</i>	Information
<i>Meaning</i>	The file has been completely received.
<i>Parameters</i>	1 Connected Site alias
	2 Local connection identifier (number)
	3 Local number of the transfer (0 if not available)
	4 Protocol destination
	5 Physical path of the file being received
<i>Action</i>	None

242	HTTP E LIST_ERR %1(%2) [%3] %4 Error receiving listing: %5
<i>Type</i>	Error

Meaning	An error occurred while receiving the file.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Protocol destination
	5	Physical path of the file being received
Action	None	
300	HTTP I TRACE %1(%2) [%3] Sending request: %4 (length %5)	
Type	Information	
Meaning	A request is being sent to the server.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Request sent (for example, GET / HTTP/1.0)
	5	Length of data to be sent
Action	None	
301	HTTP E RECV_ERR %1(%2) [%3] Request refused by server (%4 '%5')	
Type	Error	
Meaning	Request has been rejected by the server (returned code >= 300).	

<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Return code
	5	Returned description
<i>Action</i>	None	

302	HTTP I TRACE %1(%2) [%3] Receive response: %4 (length %5)	
<i>Type</i>	Information	
<i>Meaning</i>	A response has been received.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Answer (for example, HTTP/1.0 200 OK)
	5	Length of the data to be received
<i>Action</i>	None	

303	HTTP E ERROR %1(%2) [%3] Invalid server response: %4	
<i>Type</i>	Error	
<i>Meaning</i>	Some elements could not be recognized or accepted in the response from the server.	
<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Answer (for example, HTTP/1.0 200 OK)
<i>Action</i>	None	

304	HTTP E AUTH_ERR %1(%2) [%3] Auth rejected by server (%4 '%5')	
Type	Error	
Meaning	The server returned a negative answer to the first request used (identification).	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Return code (>= 300)
	5	Returned description
Action	None	

305	HTTP W REDIR_W %1(%2) [%3] Receive response: %4 (length %5); please check/update the configuration	
Type	Warning	
Meaning	A "HTTP 301 Moved Permanently" was received.	
Parameters	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Answer (for example, HTTP/1.0 301 Moved Permanently)
	5	Length of the data to be received
Action	Update the configuration to reflect the new address.	

306	HTTP I NTLM_1 %1(%2) [%3] %4 NTLM%5 Authentication %6	
Type	Information	
Meaning	An NTLM authentication operation was performed.	

<i>Parameters</i>	1	Connected Site alias
	2	Local connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Direction (Receiving or Sending an NTLM authentication message)
	5	NTLM protocol version (NTLM or NTLM2)
	6	Type of the NTLM authentication message
<i>Action</i>	None.	
400	HTTP W TIMEOUT %1(%2) Login deadline expired	
<i>Type</i>	Warning	
<i>Meaning</i>	Gateway has cut a connection from a partner, that was not followed by a completed login within a reasonable timeout. This mechanism avoids DOS-type attacks, that attempt to keep one connection slot occupied without proceeding to log in.	
<i>Parameters</i>	1	Remote protocol identifier
	2	Connection identifier
<i>Actions</i>	• Monitor calling addresses for the "login timed out" connections. Check for repeated connection attempts not followed by login attempts • Check network bandwidth	

Axway Gateway: Messages and codes

HTTP reply codes

Introduction to HTTP code

This is a summary of the RFC 913.

The Status-Code element is a 3-digit integer result code of the attempt to understand and satisfy the request. These codes are fully defined in section 10 of the RFC. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automata and the Reason-Phrase is intended for the human user. The client is not required to examine or display the Reason-Phrase.

The first digit of the Status-Code defines the class of response. The last two digits do not have any categorization role. There are 5 values for the first digit:

- 1xx: Informational - Request received, continuing process
- 2xx: Success - The action was successfully received, understood, and accepted
- 3xx: Redirection - Further action must be taken in order to complete the request
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
- 5xx: Server Error - The server failed to fulfill an apparently valid request

The individual values of the numeric status codes defined for HTTP/1.1, and an example set of corresponding Reason-Phrases, are presented below. The reason phrases listed here are only recommendations: they MAY be replaced by local equivalents without affecting the protocol.

Code	Reason
100	Continue
101	Switching Protocols
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified

Code	Reason
305	Use Proxy
307	Temporary Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-out
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable

Code	Reason
504	Gateway Time-out
505	HTTP Version not supported

HTTP status codes are extensible. HTTP applications are not required to understand the meaning of all registered status codes, though such understanding is obviously desirable. However, applications **MUST** understand the class of any status code, as indicated by the first digit, and treat any unrecognized response as being equivalent to the x00 status code of that class, with the exception that an unrecognized response **MUST NOT** be cached. For example, if an unrecognized status code of 431 is received by the client, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 status code. In such cases, user agents **SHOULD** present to the user the entity returned with the response, since that entity is likely to include human-readable information which will explain the unusual status.

HTTP reply codes detail

Code 100 Continue

The client **SHOULD** continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server. The client **SHOULD** continue by sending the remainder of the request or, if the request has already been completed, ignore this response. The server **MUST** send a final response after the request has been completed. See section 8.2.3 for detailed discussion of the use and handling of this status code.

Switching Protocols

The server understands and is willing to comply with the clients request, via the Upgrade message header field (section 14.42), for a change in the application protocol being used on this connection. The server will switch protocols to those defined by the responses Upgrade header field immediately after the empty line which terminates the 101 response.

The protocol **SHOULD** be switched only when it is advantageous to do so. For example, switching to a newer version of HTTP is advantageous over older versions, and switching to a real-time, synchronous protocol might be advantageous when delivering resources that use such features.

Successful 2xx

This class of status code indicates that the clients request was successfully received, understood, and accepted.

200 OK

The request has succeeded. The information returned with the response is dependent on the method used in the request, for example:

- GET an entity corresponding to the requested resource is sent in the response
- HEAD the entity-header fields corresponding to the requested resource are sent in the response without any message-body
- POST an entity describing or containing the result of the action
- TRACE an entity containing the request message as received by the end server

201 Created

The request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field. The response SHOULD include an entity containing a list of resource characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. The origin server MUST create the resource before returning the 201 status code. If the action cannot be carried out immediately, the server SHOULD respond with 202 (Accepted) response instead.

A 201 response MAY contain an ETag response header field indicating the current value of the entity tag for the requested variant just created.

202 Accepted

The request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place. There is no facility for re-sending a status code from an asynchronous operation such as this.

The 202 response is intentionally non-committal. Its purpose is to allow a server to accept a request for some other process (perhaps a batch-oriented process that is only run once per day) without requiring that the user agents connection to the server persist until the process is completed. The entity returned with this response SHOULD include an indication of the requests current status and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled.

203 Non-Authoritative Information

The returned meta-information in the entity-header is not the definitive set as available from the origin server, but is gathered from a local or a third-party copy. The set presented MAY be a subset or superset of the original version. For example, including local annotation information about the resource might result in a superset of the meta-information known by the origin server. Use of this response code is not required and is only appropriate when the response would otherwise be 200 (OK).

204 No Content

The server has fulfilled the request but does not need to return an entity-body, and might want to return updated meta-information. The response MAY include new or updated meta-information in the form of entity-headers, which if present SHOULD be associated with the requested variant.

If the client is a user agent, it **SHOULD NOT** change its document view from that which caused the request to be sent. This response is primarily intended to allow input for actions to take place without causing a change to the user agents active document view, although any new or updated meta-information **SHOULD** be applied to the document currently in the user agents active view.

The 204 response **MUST NOT** include a message-body, and thus is always terminated by the first empty line after the header fields.

205 Reset Content

The server has fulfilled the request and the user agent **SHOULD** reset the document view which caused the request to be sent. This response is primarily intended to allow input for actions to take place via user input, followed by a clearing of the form in which the input is given so that the user can easily initiate another input action. The response **MUST NOT** include an entity.

206 Partial Content

The server has fulfilled the partial GET request for the resource. The request **MUST** have included a Range header field (section 14.35) indicating the desired range, and **MAY** have included an If-Range header field (section 14.27) to make the request conditional.

The response **MUST** include the following header fields:

- Either a Content-Range header field (section 14.16) indicating the range included with this response, or a multi-part/byte ranges Content-Type including Content-Range fields for each part. If a Content-Length header field is present in the response, its value **MUST** match the actual number of OCTETs transmitted in the message-body.
- Date
- ETag and/or Content-Location, if the header would have been sent in a 200 response to the same request
- Expires, Cache-Control, and/or Vary, if the field-value might differ from that sent in any previous response for the same variant

If the 206 response is the result of an If-Range request that used a strong cache validator (see section 13.3.3), the response **SHOULD NOT** include other entity-headers. If the response is the result of an If-Range request that used a weak validator, the response **MUST NOT** include other entity-headers; this prevents inconsistencies between cached entity-bodies and updated headers. Otherwise, the response **MUST** include all of the entity-headers that would have been returned with a 200 (OK) response to the same request.

A cache **MUST NOT** combine a 206 response with other previously cached content if the ETag or Last-Modified headers do not match exactly, see 13.5.4.

A cache that does not support the Range and Content-Range headers **MUST NOT** cache 206 (Partial) responses.

Redirection 3xx

This class of status code indicates that further action needs to be taken by the user agent in order to fulfill the request. The action required **MAY** be carried out by the user agent without interaction with the user if and only if the method used in the second request is GET or HEAD. A client **SHOULD** detect

infinite redirection loops, since such loops generate network traffic for each redirection.

Note: previous versions of this specification recommended a maximum of five redirections. Content developers should be aware that there might be clients that implement such a fixed limitation.

300 Multiple Choices

The requested resource corresponds to any one of a set of representations, each with its own specific location, and agent-driven negotiation information (section 12) is being provided so that the user (or user agent) can select a preferred representation and redirect its request to that location.

Unless it was a HEAD request, the response SHOULD include an entity containing a list of resource characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. Depending upon the format and the capabilities of the user agent, selection of the most appropriate choice MAY be performed automatically. However, this specification does not define any standard for such automatic selection.

If the server has a preferred choice of representation, it SHOULD include the specific URI for that representation in the Location field; user agents MAY use the Location field value for automatic redirection. This response is cacheable unless indicated otherwise.

301 Moved Permanently

The requested resource has been assigned a new permanent URI and any future references to this resource SHOULD use one of the returned URIs. Clients with link editing capabilities ought to automatically re-link references to the Request-URI to one or more of the new references returned by the server, where possible. This response is cacheable unless indicated otherwise.

The new permanent URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s).

If the 301 status code is received in response to a request other than GET or HEAD, the user agent MUST NOT automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

Note: When automatically redirecting a POST request after receiving a 301 status code, some existing HTTP/1.0 user agents will erroneously change it into a GET request.

302 Found

The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

The temporary URI SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s).

If the 302 status code is received in response to a request other than GET or HEAD, the user agent **MUST NOT** automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

Note: RFC 1945 and RFC 2068 specify that the client is not allowed to change the method on the redirected request. However, most existing user agent implementations treat 302 as if it were a 303 response, performing a GET on the Location field-value regardless of the original request method. The status codes 303 and 307 have been added for servers that want to make unambiguously clear which kind of reaction is expected of the client.

303 See Other

The response to the request can be found under a different URI and **SHOULD** be retrieved using a GET method on that resource. This method exists primarily to allow the output of a POST-activated script to redirect the user agent to a selected resource. The new URI is not a substitute reference for the originally requested resource. The 303 response **MUST NOT** be cached, but the response to the second (redirected) request might be cacheable.

The different URI **SHOULD** be given by the Location field in the response. Unless the request method was HEAD, the entity of the response **SHOULD** contain a short hypertext note with a hyperlink to the new URI(s).

Note: Many pre-HTTP/1.1 user agents do not understand the 303 status. When interoperability with such clients is a concern, the 302 status code may be used instead, since most user agents react to a 302 response as described here for 303.

304 Not Modified

If the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server **SHOULD** respond with this status code. The 304 response **MUST NOT** contain a message-body, and thus is always terminated by the first empty line after the header fields.

The response **MUST** include the following header fields:

- Date, unless its omission is required by section 14.18.1
If a clockless origin server obeys these rules, and proxies and clients add their own Date to any response received without one (as already specified by [RFC 2068], section 14.19), caches will operate correctly.
- ETag and/or Content-Location, if the header would have been sent in a 200 response to the same request
- Expires, Cache-Control, and/or Vary, if the field-value might differ from that sent in any previous response for the same variant

If the conditional GET used a strong cache validator (see section 13.3.3), the response **SHOULD NOT** include other entity-headers. Otherwise (i.e., the conditional GET used a weak validator), the response **MUST NOT** include other entity-headers; this prevents inconsistencies between cached entity-bodies and updated headers.

If a 304 response indicates an entity not currently cached, then the cache **MUST** disregard the response and repeat the request without the conditional.

If a cache uses a received 304 response to update a cache entry, the cache **MUST** update the entry to reflect any new field values given in the response.

305 Use Proxy

The requested resource **MUST** be accessed through the proxy given by the Location field. The Location field gives the URI of the proxy. The recipient is expected to repeat this single request via the proxy. 305 responses **MUST** only be generated by origin servers.

Note: RFC 2068 was not clear that 305 was intended to redirect a single request, and to be generated by origin servers only. Not observing these limitations has significant security consequences.

306 (Unused)

The 306 status code was used in a previous version of the specification, is no longer used, and the code is reserved.

307 Temporary Redirect

The requested resource resides temporarily under a different URI. Since the redirection **MAY** be altered on occasion, the client **SHOULD** continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

The temporary URI **SHOULD** be given by the Location field in the response. Unless the request method was HEAD, the entity of the response **SHOULD** contain a short hypertext note with a hyperlink to the new URI(s), since many pre-HTTP/1.1 user agents do not understand the 307 status. Therefore, the note **SHOULD** contain the information necessary for a user to repeat the original request on the new URI.

If the 307 status code is received in response to a request other than GET or HEAD, the user agent **MUST NOT** automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

Client Error 4xx

The 4xx class of status code is intended for cases in which the client seems to have erred. Except when responding to a HEAD request, the server **SHOULD** include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents **SHOULD** display any included entity to the user.

If the client is sending data, a server implementation using TCP **SHOULD** be careful to ensure that the client acknowledges receipt of the packet(s) containing the response, before the server closes the input connection. If the client continues sending data to the server after the close, the server's TCP stack will send a reset packet to the client, which may erase the client's unacknowledged input buffers before they can be read and interpreted by the HTTP application.

400 Bad Request

The request could not be understood by the server due to malformed syntax. The client **SHOULD NOT** repeat the request without modifications.

401 Unauthorized

The request requires user authentication. The response **MUST** include a WWW-Authenticate header field (section 14.47) containing a challenge applicable to the requested resource. The client **MAY** repeat the request with a suitable Authorization header field (section 14.8). If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user **SHOULD** be presented the entity that was given in the response, since that entity might include relevant diagnostic information. HTTP access authentication is explained in HTTP Authentication: Basic and Digest Access Authentication [43].

402 Payment Required

This code is reserved for future use.

403 Forbidden

The server understood the request, but is refusing to fulfill it.

Authorization will not help and the request **SHOULD NOT** be repeated. If the request method was not HEAD and the server wants to make public why the request has not been fulfilled, it **SHOULD** describe the reason for the refusal in the entity. If the server does not want to make this information available to the client, the status code 404 (Not Found) can be used instead.

404 Not Found

The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent. The 410 (Gone) status code **SHOULD** be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address. This status code is commonly used when the server does not want to reveal exactly why the request has been refused, or when no other response is applicable.

405 Method Not Allowed

The method specified in the Request-Line is not allowed for the resource identified by the Request-URI. The response **MUST** include an Allow header containing a list of valid methods for the requested resource.

406 Not Acceptable

The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

Unless it was a HEAD request, the response SHOULD include an entity containing a list of available entity characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. Depending upon the format and the capabilities of the user agent, selection of the most appropriate choice MAY be performed automatically. However, this specification does not define any standard for such automatic selection.

Note: HTTP/1.1 servers are allowed to return responses which are not acceptable according to the accept headers sent in the request. In some cases, this may even be preferable to sending a 406 response. User agents are encouraged to inspect the headers of an incoming response to determine if it is acceptable.

If the response could be unacceptable, a user agent SHOULD temporarily stop receipt of more data and query the user for a decision on further actions.

407 Proxy Authentication Required

This code is similar to 401 (Unauthorized), but indicates that the client must first authenticate itself with the proxy. The proxy MUST return a Proxy-Authenticate header field (section 14.33) containing a challenge applicable to the proxy for the requested resource. The client MAY repeat the request with a suitable Proxy-Authorization header field (section 14.34). HTTP access authentication is explained in HTTP Authentication: Basic and Digest Access Authentication [43].

408 Request Timeout

The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time.

409 Conflict

The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request. The response body SHOULD include enough information for the user to recognize the source of the conflict. Ideally, the response entity would include enough information for the user or user agent to fix the problem; however, that might not be possible and is not required.

Conflicts are most likely to occur in response to a PUT request. For example, if versioning were being used and the entity being PUT included changes to a resource which conflict with those made by an earlier (third-party) request, the server might use the 409 response to indicate that it cannot complete the request. In this case, the response entity would likely contain a list of the differences between the two versions in a format defined by the response Content-Type.

410 Gone

The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. Clients with link editing capabilities SHOULD delete references to the Request-URI after user approval. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found) SHOULD be used instead. This response is cacheable unless indicated otherwise.

The 410 response is primarily intended to assist the task of web maintenance by notifying the recipient that the resource is intentionally unavailable and that the server owners desire that remote links to that resource be removed. Such an event is common for limited-time, promotional services and for resources belonging to individuals no longer working at the servers site. It is not necessary to mark all permanently unavailable resources as gone or to keep the mark for any length of time that is left to the discretion of the server owner.

411 Length Required

The server refuses to accept the request without a defined Content-Length. The client MAY repeat the request if it adds a valid Content-Length header field containing the length of the message-body in the request message.

412 Precondition Failed

The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server. This response code allows the client to place preconditions on the current resource meta-information (header field data) and thus prevent the requested method from being applied to a resource other than the one intended.

413 Request Entity Too Large

The server is refusing to process a request because the request entity is larger than the server is willing or able to process. The server MAY close the connection to prevent the client from continuing the request.

If the condition is temporary, the server SHOULD include a Retry-After header field to indicate that it is temporary and after what time the client MAY try again.

414 Request-URI Too Long

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret. This rare condition is only likely to occur when a client has improperly converted a POST request to a GET request with long query information, when the client has descended into a URI black hole of redirection (e.g., a redirected URI prefix that points to a suffix of itself), or when the server is under attack by a client attempting to exploit security holes present in some servers using fixed-length buffers for reading or manipulating the Request-URI.

415 Unsupported Media Type

The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

416 Requested Range Not Satisfiable

A server SHOULD return a response with this status code if a request included a Range request-header field (section 14.35), and none of the range-specifier values in this field overlap the current extent of the selected resource, and the request did not include an If-Range request-header field. (For byte-ranges, this means that the first-byte-pos of all of the byte-range-spec values were greater than the current length of the selected resource.)

When this status code is returned for a byte-range request, the response SHOULD include a Content-Range entity-header field specifying the current length of the selected resource. This response MUST NOT use the multipart/byteranges content-type.

417 Expectation Failed

The expectation given in an Expect request-header field could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.

Server Error 5xx

Response status codes beginning with the digit 5 indicate cases in which the server is aware that it has erred or is incapable of performing the request. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. User agents SHOULD display any included entity to the user. These response codes are applicable to any request method.

500 Internal Server Error

The server encountered an unexpected condition which prevented it from fulfilling the request.

501 Not Implemented

The server does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.

502 Bad Gateway

The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

503 Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay MAY be indicated in a Retry-After header. If no Retry-After is given, the client SHOULD handle the response as it would for a 500 response.

Note: The existence of the 503 status code does not imply that a server must use it when becoming overloaded. Some servers may want to simply refuse the connection.

504 Gateway Timeout

The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request.

Note: some deployed proxies are known to return 400 or 500 when DNS lookups time out.

505 HTTP Version Not Supported

The server does not support, or refuses to support, the HTTP protocol version that was used in the request message. The server is indicating that it is unable or unwilling to complete the request using the same major version as the client, other than with this error message. The response SHOULD contain an entity describing why that version is not supported and what other protocols are supported by that server.

Axway Integrator messages

This topic lists log messages relating to Axway Integrator.

520	AMTS_ROUT %1[%2] transfer routed to AMTRIX/XIB. file=%3	
<i>Type</i>	Information	
<i>Meaning</i>	A file handled by Gateway was successfully routed to Integrator.	
<i>Parameters</i>	1	Local protocol identifier
	2	Transfer identifier
	3	Path of the routed file
<i>Action</i>	None	

522	AMTS_API %1[%2] XFBCONN connector error. Verb=%3. Reason=%4 [%5]	
Type	Error	
Meaning	An error occurred while connecting with Integrator	
Parameters	1	Local protocol identifier
	2	Transfer identifier
	3	Failed operation
	4	Error code
	5	Error description
Action	Contact Axway Support	
523	AMTS_ROUT %1[%2] transfer queued for routing to AMTRIX/XIB. file=%3	
Type	Information	
Meaning	A transfer was sent to Integrator but it is queued for routing	
Parameters	1	Local protocol identifier
	2	Master transfer identifier
	3	Path of the routed file
Action	Contact Axway Support.	
524	AMTS_INFO %1[%2] %3	
Type	Information	
Meaning	Message used for debug purposes	

<i>Parameters</i>	1	Local Protocol identifier
	2	Master transfer identifier
	3	Generic connector diagnostic message
<i>Action</i>	Contact Axway Support	
525	AMTS_ERRO %1[%2] %3	
<i>Type</i>	Error	
<i>Meaning</i>	Message used for debug purposes	
<i>Parameters</i>	1	Local Protocol identifier
	2	Transfer identifier
	3	Generic connector diagnostic message
<i>Action</i>	Contact Axway Support	

Axway Gateway: Messages and codes

JMS

This topic lists JMS log messages.

101	JMS I TRACE %1
<i>Type</i>	Information
<i>Meaning</i>	Technical information displayed when using trace mode on the Site.
<i>Parameters</i>	1 Message
<i>Action</i>	None
102	JMS W TRACE %1
<i>Type</i>	Warning

Meaning	Third party warning: not used so far.	
Parameters	1	Message
Action	None	
103	JMS E TRACE %1	
Type	Error	
Meaning	Technical/system error encountered.	
Parameters	1	Error message
Action	If necessary, correct the encountered error.	
104	JMS I OPEN Queue %1 (read) opened	
Type	Information	
Meaning	Activity tracing: Queue has been opened for reading (input).	
Parameters	1	Queue alias
Action	None	
105	JMS I OPEN Queue %1 (write) opened	
Type	Information	
Meaning	Activity tracing: Queue has been opened for writing (output).	
Parameters	1	Queue alias
Action	None	
106	JMS I CLOSE Queue %1 (read) closed	
Type	Information	

Meaning	Activity tracing: Queue, which was opened for reading, has been closed.	
Parameters	1	Queue alias
Action	None	
107	JMS I CLOSE Queue %1 (write) closed	
Type	Information	
Meaning	Activity tracing: Queue, which was opened for writing, has been closed.	
Parameters	1	Queue alias
Action	None	
108	JMS W CLOSE Queue %1 (read) went down (%2 error)	
Type	Warning	
Meaning	Queue, which was opened for reading, has been abnormally closed.	
Parameters	1	Queue alias
	2	Origin of the error: <i>System, Local, Remote</i> or <i>Protocol</i>
Action	None	
109	JMS W CLOSE Queue %1 (write) went down (%2 error)	
Type	Warning	
Meaning	Queue, which was opened for writing, has been abnormally closed.	
Parameters	1	Queue alias
	2	Origin of the error: <i>System, Local, Remote</i> or <i>Protocol</i>
Action	None	

110	JMS I SEND Transfer %1 sent on queue %2	
Type	Information	
Meaning	Activity tracing: Transfer has been sent to the queue.	
Parameters	1	Transfer number
	2	Queue alias
Action	None	
111	JMS E SEND_ERR Transfer %1 failed on queue %2	
Type	Error	
Meaning	Transfer could not be sent to the queue.	
Parameters	1	Transfer number
	2	Queue alias
Action	None	
112	JMS I RECV Transfer %1 received on queue %2	
Type	Information	
Meaning	Message received on queue.	
Parameters	1	Transfer number
	2	Queue alias
Action	None	
113	JMS I RECV_ERR Transfer creation failed on queue %1: %2	
Type	Information	
Meaning	Transfer could not be created for the incoming message.	

<i>Parameters</i>	1	Queue alias
	2	Error message
<i>Action</i>	None	

Axway Gateway: Messages and codes

Job Control Table (JCT)

In the following messages, the parameter **%1** is the internal job name (JOBNAME). This parameter enables you to locate information that is associated with the process, and to find other log messages concerning the job process. You can use this information to better understand a message and diagnose an error, for example, by displaying the exit file, `$p_home_dir/run_time/tmp/JOBNAME-<timestamp>-<pid>.out`, that is associated with the job.

Dynamic processes

001	JCT E FORKFAIL [%1] Process creation failure: %2	
<i>Type</i>	Error	
<i>Meaning</i>	Error occurred during process creation. The local file system error is returned in parameter %1.	
<i>Parameters</i>	1	System error message
	2	Job name
<i>Action</i>	Refer to your platform documentation for information on the local file system error. If the system error message does not help resolve the problem, contact Axway Support.	
002	JCT I JOBQUEUED [%1] Job [%2] queued	
<i>Type</i>	Information	
<i>Meaning</i>	A user job (user command executed in background) cannot be processed immediately, since the maximum number of running jobs has been reached (max_job configuration parameter).	

Parameters	1	Job name
	2	Job command associated with the job (the command may be shorted, with only the beginning of the command displayed)
Action	None: The job is queued and will be automatically scheduled when the number of running jobs decreases.	
003	JCT I JOBTERML [%1] Job [%2] pid %3 terminated	
Type	Information	
Meaning	The process executing the job has completed and exited.	
Parameters	1	Job name
	2	Job command associated with the job (the command may be shorted, with only the beginning of the command displayed)
	3	System process identifier
Action	None	
004	JCT I JOBSUBM [%1] Job [%2] submitted with pid %3	
Type	Information	
Meaning	The job process was submitted.	
Parameters	1	Job name
	2	Trace
	3	System process identifier
Action	None	
010	JCT W JOBSIG [%1] Job [%2] pid %3 terminated on signal %4	
Type	Warning (UNIX only)	
Meaning	The job terminated abnormally on the indicated signal.	

<i>Parameters</i>	1	Job name
	2	User command associated with the job (the command may be shorted, with only the beginning of the command displayed)
	3	System process identifier
	4	Signal
<i>Action</i>	Analyze the message for job type and signal.	

011	JCT W JOBSTOP [%1] Job [%2] pid %3 stopped on signal %4	
<i>Type</i>	Warning (UNIX only)	
<i>Meaning</i>	The job stopped abnormally on the indicated signal.	
<i>Parameters</i>	1	Job name
	2	Job command associated with the job (the command may be shorted, with only the beginning of the command displayed)
	3	System process identifier
	4	Signal
<i>Action</i>	Analyze the message for job type and signal.	

012	JCT W JOBEXIT [%1] Job [%2] pid %3 exited with status %4	
<i>Type</i>	Warning (UNIX only)	
<i>Meaning</i>	The program stopped. When a job stops with the status displayed, you can use the status to diagnose the error.	
<i>Parameters</i>	1	Job name
	2	Job command associated with the job (the command may be shorted, with only the beginning of the command displayed)
	3	System process identifier
	4	Status
<i>Action</i>	Contact Axway Support.	

013	JCT W JOBMESS [%1] Job [%2] pid %3 messed up with wait status %4	
<i>Type</i>	Warning	
<i>Meaning</i>	Job has stopped with no available information to diagnose the situation.	
<i>Parameters</i>	1	Job name
	2	Job command associated with the job (the command may be shorted, with only the beginning of the command displayed)
	3	System process identifier
	4	Status
<i>Action</i>	Contact Axway Support.	

User processes

051	JCT E FORKFAIL [%1] Process creation failure : %2	
<i>Type</i>	Error	
<i>Meaning</i>	An error occurred during batch creation.	
<i>Parameters</i>	1	System error message
	2	Batch command associated with the job process (may be truncated)
<i>Action</i>	Refer to your platform documentation for more information. If the system error messages does not help resolve the problem, contact Axway Support.	

052	JCT I CMDQUEUED [%1] Batch command [%2] queued	
<i>Type</i>	Information	
<i>Meaning</i>	A batch command cannot be processed immediately because the maximum number of running batches has been reached.	

Parameters	1	System error message
	2	Batch command associated with the job process (may be truncated)
Action	None	
053	JCT I CMDTERML [%1] Batch command [%2] pid %3 completed (exited with status 0)	
Type	Information	
Meaning	The process executing the batch command has finished and exited.	
Parameters	1	Job name
	2	Batch command associated with the job process (may be truncated)
	3	System process identifier
Action	None	
054	JCT I CMDSUBM [%1] Batch command [%2] submitted with pid %3	
Type	Information	
Meaning	The batch command was submitted.	
Parameters	1	Job name
	2	Batch command associated with the job process (command may be truncated)
	3	System process identifier
Action	None	
060	JCT W CMDSIG [%1] Batch command [%2] pid %3 terminated on signal %4	
Type	Warning	

Meaning	The batch has terminated abnormally on the indicated signal.		
Parameters	1	Job name	
	2	Batch command associated with the job process (command may be truncated)	
	3	System process identifier	
	4	Signal	
Action	Analyze the message for job type and signal.		
061	JCT W CMDSTOP [%1] Batch command [%2] pid %3 stopped on signal %4		
Type	Warning		
Meaning	The batch stopped abnormally on the indicated signal.		
Parameters	1	Job name	
	2	Batch command associated with the job process (command may be truncated)	
	3	System process identifier	
	4	Signal	
Action	Analyze the message for job type and signal. If the system error messages does not help resolve the problem, contact Axway Support.		
062	JCT W CMDEXIT [%1] Batch command [%2] pid %3 exited with status %4 -- %5		
Type	Warning		
Meaning			

Parameters	1	Job name
	2	Batch command associated with the job process (command may be truncated)
	3	System process identifier
	4	Status
	5	Status
Action	The final two lines of the Standard Error file output are displayed as Status %4-%5. If this error message does not help resolve the problem, contact Axway Support.	
063	063 JCT W CMDMESS [%1] Batch command [%2] pid %3 messed up with wait status %4	
Type	Warning	
Meaning	Job process defaulted and no diagnostic information is available.	
Parameters	1	Job name
	2	Batch command associated with the job process (command may be truncated)
	3	System process identifier
	4	Status
Action	Contact Axway Support.	

Axway Gateway: Messages and codes

LU 6.2 SNA network (SNAC)

This topic lists LU 6.2 SNA network log messages.

003	SNAC I CONN_RP- allocation request refused: conversation in progress
<i>Type</i>	Information

Meaning	CPI-C is a single conversation process. Another conversation cannot begin until the current one is ended.		
Parameters	None		
Action	None		
004	SNAC I ALLOC_RQ allocation request: dest=%1 %2 %3 %4		
Type	Information		
Meaning	Beginning of an LU 6.2 conversation. The "dest" parameter indicates the CPI-C symbolic destination.		
Parameters	1	Conversation ID assigned	
	2	Remote LU name	
	3	Transaction program name	
	4	Name of the conversation mode	
Action	None		
005	SNAC I ALLOC_AK allocation accepted: RLU=%1 MODE=%2 CID=%3		
Type	Information		
Meaning	The requested conversation is accepted.		
Parameters	1	Remote LU name	
	2	Name of the conversation mode	
	3	Conversation ID assigned	
Action	None		
006	SNAC I ALLOC_NK allocation refused: RLU=%1 FONC=%2, ORIGINE=%3, RCODE=%4, ERRNO=%5		

<i>Type</i>	<i>Information</i>	
<i>Meaning</i>	An unexpected error occurred during conversation allocation. The name of the CPI-C function concerned and its return code are displayed. ORIGIN indicates whether the refusal is from the Local or Remote Site.	
<i>Parameters</i>	1	Remote LU name
	2	CPI-C function concerned
	3	Origin of the error
	4	CPI-C function return code
	5	System-specific return code
<i>Action</i>	If the function is cminit , check that the symbolic destination is defined in the SNA setup. If not, check the appropriate CPI-C manual delivered with the product to determine why the function failed.	

**007 SNAC I ALLOC_ID allocation requested: RLU=%1 MODE=%2
CID=%3**

<i>Type</i>	<i>Information</i>	
<i>Meaning</i>	The remote system has requested a conversation allocation.	
<i>Parameters</i>	1	Remote LU name
	2	Name of the conversation mode
	3	Conversation ID assigned
<i>Action</i>	None	

008 SNAC I DEALL_RQ deallocation request: RLU=%1 OPTION=%2

<i>Type</i>	<i>Information</i>	
<i>Meaning</i>	The local system has requested termination of the conversation.	
<i>Parameters</i>	1	Remote LU name
	2	De-allocation option
<i>Action</i>	None	

009	SNAC I DEALL_ID deallocation request: RLU=%1 OPTION=%2	
Type	Information	
Meaning	The local system has requested termination of the conversation.	
Parameters	1	Remote LU name
	2	De-allocation option
Action	None	
010	SNAC I CONV_END end of conversation: RLU=%1 RCODE=%2, ERRNO=%3	
Type	Information	
Meaning	Reason for the termination of the conversation (normal or abnormal). If an error has occurred, refer to SNA message number 20 for details.	
Parameters	1	Remote LU name
	2	CPI-C error
	3	System specific error
Action	None	
020	SNAC E ERROR RLU=%1 FONC=%2 ORIGIN=%3, RCODE=%4, ERRNO=%5	
Type	Error	
Meaning	An unexpected error occurred during conversation. The name of the CPI-C function concerned and its return code are displayed. ORIGIN indicates whether the refusal is from the Local or Remote Site.	

<i>Parameters</i>	1	Remote LU name
	2	CPI-C function concerned
	3	Origin of the error
	4	CPI-C function return code
	5	System-specific return code
<i>Action</i>	If the function is cminit , check that the symbolic destination is defined in the SNA setup. If not, check the CPI-C manual delivered with the product to determine why the function failed.	

Axway Gateway: Messages and codes

Axway Messaging connector routing (XMSC)

This topic lists the log messages for the Axway Messaging connector routing. The general format for these messages is a text message.

001	XMSC E PROP list properties selected, name:%1 not found	
<i>Type</i>	Error	
<i>Meaning</i>	Property List was not found in the Rule Table.	
<i>Parameters</i>	1	Property List name
<i>Action</i>	Check that the Property List name entered in the Rule Table exists.	
002	XMSC E CONN error of connection to XMS:%1 %2	
<i>Type</i>	Error	
<i>Meaning</i>	Error connecting to Axway Messaging.	
<i>Parameters</i>	1	Error code
	2	Error message
<i>Action</i>	Check that Axway Messaging is started.	

003	XMSC I CONN connection to XMS established	
Type	Information	
Meaning	A connection to Axway Messaging was established.	
Parameters	None	
004	XMSC E CONN error of disconnection to XMS:%1 %2	
Type	Error	
Meaning	Error disconnecting from Axway Messaging.	
Parameters	1	Error code
	2	Error message
Action	Check that the Axway Messaging has not stopped.	
005	XMSC I CONN connection to XMS finished	
Type	Information	
Meaning	Connection to Axway Messaging is successfully completed.	
Parameters	None	
006	XMSC E MSG error to get XMS message:%1 %2	
Type	Error	
Meaning	Error retrieving an Axway Messaging message.	
Parameters	1	Code error
	2	Error message
Action	Check the following error message sent by Axway Messaging.	

007	XMSC E MSG error: the type of XMS message (%1) is not XMS_DATA_TYPE	
Type	Error	
Meaning	The type of message recuperated is not the XMS_DATA_TYPE	
Parameters	1	Message ID
Action	Check to see why a message of a different type arrived in the queue to be processed by the Axway Messaging connector.	
008	XMSC E ALLOC allocation error for: %1	
Type	Error	
Meaning	Memory allocation error occurred	
Parameters	1	Allocation field name
009	XMSC E FILE file already exist: %1	
Type	Error	
Meaning	File already exists	
Parameters	1	File name
Action	Delete or move the file in the receiving directory.	
010	XMSC E FILE can't create file: %1	
Type	Error	
Meaning	Impossible to create a file	
Parameters	1	File name
Action	Check the write rights or check for adequate disc space.	

011	XMSC E PROP error property "%1" not found in XMS message(%2 %3)	
Type	Error	
Meaning	Message property not found	
Parameters	1	Name of property
	2	Code error
	3	Error message
Action	Check that the application that is sending the message is informed of the missing property.	
012	XMSC E PROP error type of property "%1"	
Type	Error	
Meaning	Property type is incorrect	
Parameters	1	Property type
Action	You must either modify the property type in the Property List declaration, or modify the type in the sent message.	
013	XMSC E FUNCT error in function: %1	
Type	Error	
Meaning	A function error occurred	
Parameters	1	Name of the function
014	XMSC E CONN can't connect to Gateway	
Type	Error	
Meaning	Impossible to connect to Gateway	
Parameters	None	
Action	Check that Gateway is started.	

015	XMSC I CONN connected to Gateway	
Type	Information	
Meaning	The connection to Gateway was successful	
Parameters	None	
Action	None	
016	XMSC E TRANS error to put a transfer: %1 %2	
Type	Error	
Meaning	Error occurred while attempting to complete a transfer	
Parameters	1	Error code
	2	Message error
Action	Action in accordance with the error message.	
017	XMSC I TRANS deposit of transfer %1	
Type	Information	
Meaning	Transfer send was successful	
Parameters	1	Local identifier for transfer
018	XMSC E MSG error to commit XMS message %1: %2 %3	
Type	Error	
Meaning	Impossible to commit the Axway Messaging session.	
Parameters	1	Message ID
	2	Error code
	3	Error message

<i>Action</i>	Action in accordance with the error message.	

019	XMSC E TRANS XferGet(%1) error: %2[%3]	
<i>Type</i>	Error	
<i>Meaning</i>	An error occurred recuperating a transfer	
<i>Parameters</i>	1	Transfer ID
	2	Error code
	3	Error message
<i>Action</i>	Action in accordance with the error message.	

020	XMSC E TRANS XferUpdate failed: id(%1), state(%2), rc(%3) [%4]	
<i>Type</i>	Error	
<i>Meaning</i>	Error updating a transfer	
<i>Parameters</i>	1	Transfer ID
	2	Transfer status
	3	Error code
	4	Error message
<i>Action</i>	Check that the transfer with this ID is still in the Gateway Mailbox.	

021	XMSC I TRANS XferUpdate: id(%1), state(%2)	
<i>Type</i>	Information	
<i>Meaning</i>	Transfer update successful	
<i>Parameters</i>	1	Transfer ID
	2	Transfer status
<i>Action</i>	None	

022	XMSC E MSG send message %1 failed: %2 %3	
Type	Error	
Meaning	Error sending an Axway Messaging message.	
Parameters	1	Message ID
	2	Error code
	3	Error message
Action	Action in accordance with the error message.	
023	XMSC E RULE RuleTable with XmsCtor name "%1" not found	
Type	Error	
Meaning	Rule Table with this connector name was not found.	
Parameters	1	Axway Messaging connector name
Action	Check that a Rule Table with this connector name exists.	
024	XMSC E RULE they are too many RuleTables with XmsCtor name "%1"	
Type	Error	
Meaning	There are several (more than one) Rule Tables with this connector name.	
Parameters	1	Axway Messaging connector name
Action	Delete or modify the Rule Table so that there is only one with this name.	
025	XMSC E MSG error to put a message error : %1 %2	
Type	Error	
Meaning	Error depositing message	

Parameters	1	Error code
	2	Message code
Action	Action in accordance with the error message.	
026	XMSC E MSG error, Xms messageID null	
Type	Error	
027	XMSC E MSG put the message %1 in backup file "%2"	
Type	Error	
Meaning	Put the message in queue	
Parameters	1	Message ID
	2	Queue name
028	XMSC E MSG error to put the backup message %1 in "%2": %3 %4	
Type	Error	
Meaning	Error occurred putting message in queue.	
Parameters	1	Message ID
	2	Queue name
	3	Error code
	4	Message code
Action	Action in accordance with the error message.	
029	XMSC I CONN connect to queue :%1	
Type	Information	

Meaning	Gateway has connected to Axway Messaging queue.	
Parameters	1	Axway Messaging queue name
Action	None	

030	XMSC E MSG can't sent an error message because the replyToQmgr field in the header is empty	
Type	Error	
Meaning	The replyToQmgr field in the header of the message to be processed is empty.	
Action	Configure the message headers.	

031	XMSC E MSG can't sent an error message because the replyToQueue field in the header is empty	
Type	Error	
Meaning	The replyToQmgr field in the header of the message to be processed is empty.	
Action	Configure the message headers.	

Axway Gateway: Messages and codes

MGET and MPUT messages (MXF)

This topic lists MGET and MPUT log messages.

104	MXF I LOGIN Login failed, username %1	
<i>Type</i>	Information	
<i>Meaning</i>	Login has failed.	
<i>Parameters</i>	1	User name
<i>Action</i>	Check user name and password.	

105	MXF I MGET Processing MGET for master ident %1	
Type	Information	
Meaning	MGET being processed.	
Parameters	1	Master transfer identifier
Action	None	
106	MXF I MPUT Processing MPUT for master ident %1	
Type	Information	
Meaning	MPUT being processed.	
Parameters	1	Master transfer identifier
Action	None	
107	MXF E MGET MGET processing error for master ident %1, reason=%2, errcod = %3	
Type	Error	
Meaning	An MGET processing error has occurred.	
Parameters	1	Master transfer identifier
	2	Error reason
	3	Error code
Action	Contact Axway Support.	
108	MXF E MPUT MPUT processing error for master ident %1, reason=%2, errcod = %3	
Type	Error	
Meaning	An MPUT processing error has occurred.	

Parameters	1	Master transfer identifier
	2	Error reason
	3	Error code
Action	Contact Axway Support.	
109	MXF E MGET MGET error for master ident %1, can not open list file %2	
Type	Error	
Meaning	Cannot open list of files to transfer.	
Parameters	1	Master transfer identifier
	2	Name of list file
Action	Contact Axway Support.	
110	MXF E MGET MGET fseek error for master ident %1, file %2, errno=%3	
Type	Error	
Meaning	System error	
Parameters	1	Master transfer identifier
	2	File name
	3	System error code
Action	Contact Axway Support.	
111	MXF W MGET MGET for master ident %1, line ignored : %2	
Type	Warning	
Meaning	This line has been ignored as it does not contain a valid filename.	

Parameters	1	Master transfer identifier
	2	Contents of ignored line
Action	Check the file name.	
112	MXF I MGET MGET for master ident %1, skipped file with too long name %2	
Type	Information	
Meaning	File name too long.	
Parameters	1	Master transfer identifier
	2	File name
Action	Check the file name. Max length = 127 characters.	
113	MXF E MGET MGET ftell error for master ident %1, file %2, errno=%3	
Type	Error	
Meaning	System error	
Parameters	1	Master transfer identifier
	2	File name
	3	System error code
Action	Contact Axway Support.	
114	MXF I MGET MGET for master ident %1 ended, %2 transfer requests submitted	
Type	Information	
Meaning	The MGET has been processed.	

<i>Parameters</i>	1	Master transfer identifier
	2	Number of "child" Transfer Requests
<i>Action</i>	None	

115	MXF I MGET MGET slice for master ident %1, %2 more transfer requests

<i>Type</i>	Information	
<i>Meaning</i>	Part of MGET has been processed. More Transfer Requests remain.	
<i>Parameters</i>	1	Master transfer identifier
	2	Number of "child" Transfer Requests
<i>Action</i>	None	

116	MXF E MGET MGET for master ident %1, request failed for file %2

<i>Type</i>	Error	
<i>Meaning</i>	Transfer could not be created.	
<i>Parameters</i>	1	Master transfer identifier
	2	File name
<i>Action</i>	Check that file exists in file directory.	

117	MXF I MGET MGET for master ident %1, request success for file %2, transfer ident=%3

<i>Type</i>	Information	
<i>Meaning</i>	Transfer has been created.	

Parameters	1	Master transfer identifier
	2	File name
	3	"Child" transfer identifier
Action	None	
118	MXF E MPUT MPUT error for master ident %1, can not open list file %2	
Type	Error	
Meaning	Cannot open list of files to transfer.	
Parameters	1	Master transfer identifier
	2	Name of list file
Action	Contact Axway Support.	
119	MXF E MPUT MPUT fseek error for master ident %1, file %2, errno=%3	
Type	Error	
Meaning	System error	
Parameters	1	Master transfer identifier
	2	File name
	3	System error code
Action	Contact Axway Support.	
120	MXF E MGET MPUT ftell error for master ident %1, file %2, errno=%3	
Type	Error	
Meaning	System error	

<i>Parameters</i>	1	Master transfer identifier
	2	File name
	3	System error code
<i>Action</i>	Contact Axway Support.	

121	MXF I MPUT MPUT for master ident %1 ended, %2 transfer requests submitted	
<i>Type</i>	Information	
<i>Meaning</i>	The MPUT has been processed.	
<i>Parameters</i>	1	Master transfer identifier
	2	Number of "child" Transfer Requests
<i>Action</i>	None	

122	MXF I MPUT MPUT slice for master ident %1, %2 more transfer requests	
<i>Type</i>	Information	
<i>Meaning</i>	Part of MPUT has been processed. More Transfer Requests remain.	
<i>Parameters</i>	1	Master transfer identifier
	2	Number of "child" Transfer Requests
<i>Action</i>	None	

123	MXF E MPUT MPUT for master ident %1, request failed for file %2	
<i>Type</i>	Error	
<i>Meaning</i>	Transfer could not be created.	

Parameters	1	Master transfer identifier
	2	File name
Action	Check that file exists in file directory.	
124	MXF I MPUT MPUT for master ident %1, request success for file %2, transfer ident=%3	
Type	Information	
Meaning	Transfer has been created.	
Parameters	1	Master transfer identifier
	2	File name
	3	"Child" transfer identifier
Action	None	
125	MXF I MPUT MPUT for master ident %1, local list done for directory %2, contains %3 files to send	
Type	Information	
Meaning	MPUT list created in local directory.	
Parameters	1	Master transfer identifier
	2	Directory name
	3	Number of files to send
Action	None	
126	MXF I MPUT MPUT opendir error for master ident %1, directory %2, errno=%3	
Type	Information	
Meaning	System error	

Parameters	1	Master transfer identifier
	2	Directory name
	3	System error code
Action	Contact Axway Support.	
127	MXF W MPUT MPUT for master ident %1, one file skipped because name length=%2 exceeds maximum=%3	
Type	Warning	
Meaning	File skipped because the file name is too long.	
Parameters	1	Master transfer identifier
	2	Length of file name
	3	Maximum allowed length for file name
Action	Check that file names are valid. Max length = 127 characters.	
128	MXF W MPUT MPUT for master ident %1, skipped file is %2	
Type	Warning	
Meaning	Name of the skipped file.	
Parameters	1	Master transfer identifier
	2	Name of skipped file
Action	Check that file names are valid. Max length = 127 characters.	
129	MXF E MGET MGET for master ident %1, filter %2 parsing error, filtering is disabled	
Type	Error	
Meaning	Filter error	

Parameters	1	Master transfer identifier
	2	Filter
Action	Check filter syntax.	
130	MXF E MPUT MPUT for master ident %1, filter %2 parsing error, filtering is disabled	
Type	Error	
Meaning	Filter error	
Parameters	1	Master transfer identifier
	2	Filter
Action	Check filter syntax.	
131	MXF E MGET MGET for master ident %1, filter syntax error : %2	
Type	Error	
Meaning	Filter syntax error	
Parameters	1	Master transfer identifier
	2	Error message
Action	Check filter syntax.	
132	MXF E MPUT MPUT for master ident %1, filter syntax error : %2	
Type	Error	
Meaning	Filter syntax error	
Parameters	1	Master transfer identifier
	2	Error message
Action	Check filter syntax.	

133	MXF E MPUT MGET for master ident %1, no sender alias available
<i>Type</i>	Error
<i>Meaning</i>	Sender alias missing.
<i>Parameters</i>	1 Master transfer identifier
<i>Action</i>	Check contents of Transfer Request.
134	MXF E MPUT MGET for master ident %1, no receiver alias available
<i>Type</i>	Error
<i>Meaning</i>	Receiver alias missing.
<i>Parameters</i>	1 Master transfer identifier
<i>Action</i>	Check contents of Transfer Request.
135	MXF E MPUT MPUT for master ident %1, no sender alias available
<i>Type</i>	Error
<i>Meaning</i>	Sender alias missing.
<i>Parameters</i>	1 Master transfer identifier
<i>Action</i>	Check contents of Transfer Request.
136	MXF E MPUT MPUT for master ident %1, no receiver alias available
<i>Type</i>	Error
<i>Meaning</i>	Receiver alias missing.
<i>Parameters</i>	1 Master transfer identifier
<i>Action</i>	Check contents of Transfer Request.

Axway Gateway: Messages and codes

Monitor checking (CHK)

This topic lists Monitor checking log messages.

001	CHK S CHKOK Monitor check ended successfully
<i>Type</i>	Success
<i>Meaning</i>	Monitor checked all resources successfully before starting.
<i>Parameters</i>	None
<i>Action</i>	None
002	CHK I CHKBEGIN Start Monitor checking
<i>Type</i>	Information
<i>Meaning</i>	Monitor starts resource checking procedure.
<i>Parameters</i>	None
<i>Action</i>	None
050	CHK S RCVXFOK Recovered mailbox successfully
<i>Type</i>	Success
<i>Meaning</i>	The recovery of master Mailbox was successful.
<i>Parameters</i>	None
<i>Action</i>	None
051	CHK S RCVBCKOK Recovered backup successfully
<i>Type</i>	Success
<i>Meaning</i>	The recovery of mirror Mailbox was successful.
<i>Parameters</i>	None

<i>Action</i>	None

052	CHK W E_RCVXF Mailbox recovery failed: %1
<i>Type</i>	Warning
<i>Meaning</i>	The recovery of master Mailbox has failed.
<i>Parameters</i>	1 System error in string format
<i>Action</i>	Check system error to correct it. Try manual recovery. If recovery still fails, initialize Mailbox.

053	CHK W E_RCVBCK Backup recovery failed: %1
<i>Type</i>	Warning
<i>Meaning</i>	The recovery of mirror Mailbox has failed.
<i>Parameters</i>	1 System error in string format
<i>Action</i>	Check system error to correct it. Try manual recovery. If recovery still fails, initialize Mailbox or disable the Mirror option .

100	CHK E XFERINIT Opening mailbox failed: %1
<i>Type</i>	Error
<i>Meaning</i>	Checking procedure cannot open Mailbox file.
<i>Parameters</i>	1 System error in string format
<i>Action</i>	Check system error to correct it.

101	CHK E XFERCREAT Creating mailbox failed: %1
<i>Type</i>	Error
<i>Meaning</i>	Monitor was unable to create Mailbox file.

Parameters	1	System error is in string format
Action	Check system error to correct it.	
102	CHK E XFERSCAN Reading mailbox failed: %1	
Type	Error	
Meaning	An error occurred when checking Mailbox file structure.	
Parameters	1	System error is in string format or monitor error
Action	Check system error to correct it. Try manual recovery. If recovery still fails, initialize Mailbox or disable the Mirror option .	
103	CHK E XFEREND Closing mailbox failed: %1	
Type	Error	
Meaning	An error occurred while closing Mailbox file.	
Parameters	1	System error is in string format
Action	Check system error to correct it. Try recovery. If recovery still fails, initialize Mailbox or disable the Mirror option .	
104	CHK E SUPCTXR Error while loading context (%1): %2	
Type	Error	
Meaning	Checking procedure could not load context file.	
Parameters	1	System error is in string format
Action	Normally, context file is recreated with a dirty status if loading it failed. So a call to database checking should correct the problem. If you still cannot load context file, check system error to correct it.	
105	CHK E SUPCTXW Error while saving context (%1): %2	

<i>Type</i>	Error
<i>Meaning</i>	Checking procedure could not save context file.
<i>Parameters</i>	1 System error is in string format
<i>Action</i>	Check system error. Try manual recovery. If recovery still fails, initialize Mailbox or disable the Mirror option .

Network module (NET and NDSP)

This topic lists Network module log messages.

001	NET E COMMER %1
<i>Type</i>	Error
<i>Meaning</i>	An error condition has been reported by the underlying network subsystem. This software, which is supported by the operating system and/or the communication card, is platform dependent.
<i>Parameters</i>	1 Name of the communication service (API routine or system service) and an error number with additional text that usually describes the problem
<i>Action</i>	Refer to the documentation supplied with the communication interface for information on the error reported. You may also need to check the documentation for the OS.
002	NET I INITDONE comm_type=%1 - sap="%2" - incoming_max=%3 - device_name=%4 local_address=%5
<i>Type</i>	Information
<i>Meaning</i>	Successful initialization of the network interface. The parameters are those used for the initialization. They reflect the configuration selected during Gateway installation.

Parameters	1	Type of network support. Possible values are: TCP, DSA, LU 62, PAD.
	2	Selection parameters for incoming calls whose formats depend on the underlying network type and OS platform: • TCP/IP: port number for listening for incoming calls • PAD: protocol-id or an entry point in the routing table for incoming calls
	3	Maximum number of incoming connection calls. This parameter is used to limit the number of simultaneous incoming calls.
	4	Name referencing the communication hardware or software used by the network interface, depending on the type of network used. Used for some platforms only. It is typically used to select an attachment point or a communication card.
	5	References an access point to reach the local host from the network area
Action	None	

011	NDSP I REQ outgoing connection request [prot="%1"/comm="%2"/src="%3"/dest="%4"/udata="%5"]	
Type	Information	
Meaning	The network connection requested by a protocol module has been routed to the corresponding network subsystem.	
Parameters	1	File transfer protocol module that requested the network connection
	2	Network subsystem: TCP, DSA, LU 6.2, PAD
	3	Network address of the originator Site
	4	Network address of the destination Site
	5	TCP/IP port number
Action	None	

012	NDSP I IND incoming connection indication [net="%1"/comm="%3"/src="%4"/ dest="%5"/udata="%5"%6]	
Type	Information	
Meaning	An incoming network connection request has been intercepted and has been routed to the corresponding local protocol module.	
Parameters	1	File transfer protocol module that requested the network connection
	2	Network subsystem: TCP, DSA, LU 6.2, PAD
	3	Calling network address
	4	Called network address
	5	TCP/IP port number for the local destination
	6	Route name of the network description in the network security files
Action	None	
013	NDSP I ROUT connection request routed to %1	
Type	Information	
Meaning	A network connection request has been routed to the selected network procedure.	
Parameters	1	The network procedure to which the connection is forwarded
Action	None	
014	NDSP W REFUSER connection request refused by the user	
Type	Warning	
Meaning	A network connection request has been rejected by a user exit.	
Parameters	None	
Action	Verify the related exit.	

015	NDSP E BADTP connection request rejected: TP procedure unknown
<i>Type</i>	Error
<i>Meaning</i>	An outgoing connection request has been rejected locally by Gateway. The Gateway configuration file is probably incorrect.
<i>Parameters</i>	None
<i>Action</i>	Possible causes: • Gateway was not correctly started and initialized • The name of the file transfer module referenced in the request does not exist, or is different from that set up in the Gateway configuration file Correct the error. Contact Axway Support if the problem persists.
016	NDSP E BADNET connection request rejected: NET procedure unknown
<i>Type</i>	Error
<i>Meaning</i>	An incoming network connection request has been rejected by the monitor. The Gateway configuration file is probably incorrect.
<i>Parameters</i>	None
<i>Action</i>	Possible causes: • Gateway was not correctly started and initialized • The name of the Gateway network module referenced in the request does not exist, or is different from that set up in the Gateway configuration file Correct the error. Contact Axway Support if the problem persists.
017	NDSP E FAIL connection request rejected: selection invalid
<i>Type</i>	Error
<i>Meaning</i>	An incoming network connection request has been rejected by the monitor. The selected protocol module is not reachable.
<i>Parameters</i>	None

Action	Check that Gateway was correctly started with all processes running. If the problem persists, contact Axway Support.		
018	NDSP E BADADDR connection request rejected: unknown calling address		
Type	Error		
Meaning	An incoming network connection request has been rejected by the monitor. The Gateway network security files are probably incorrect.		
Parameters	None		
Action	Possible causes: • Connection is not allowed • Security files are not correctly set up Modify the security files and make sure that transfer is allowed. If the problem persists, contact Axway Support.		
019	NDSP E BADSITE connection request rejected: unknown physical site %1 name %2		
Type	Error		
Meaning	An incoming network connection request has been rejected by the monitor. The Gateway network security files or database are probably incorrect.		
Parameters	1	Type of Site name ("alias" or "proto ident")	
	2	Site name	
Action	Possible causes: • Security files are not correctly set up • Gateway database is not correctly set up Modify the security files and check the Gateway database. If the problem persists, contact Axway Support.		
020	NDSP E SECUSITE %1(%2) security error: returned site doesn't match expected site ["%3"/"%4"]		

<i>Type</i>	Error
<i>Meaning</i>	An incoming transfer request has been rejected by the monitor. The protocol identifier returned by the remote node does not match the protocol identifier of the Site found in the security file.
<i>Parameters</i>	1 Protocol identifier (sent by remote)
	2 Connection identifier
	3 Protocol identifier (link to the Site alias name found in local network security file)
	4 Route name (network security file record name)
<i>Action</i>	Possible causes: • Connection is not allowed • Gateway database is not correctly set up Modify the security files or Gateway database. If the problem persists, contact Axway Support.
021	NDSP E SECUX25 %1(%2) X25 security error: returned site doesn't match calling address [call ad="%3"/"%4"/"%5"] [wait ad="%6"/"%7"/"%8"]
<i>Type</i>	Error
<i>Meaning</i>	An incoming Transfer Request has been rejected by the monitor. When the incoming connection arrives, Gateway finds the related record in the security files. If the flag parameter does not contain the CTRL value, Gateway does not compare the protocol identifier returned by the remote node. However, it checks if there is a record in the network security file that corresponds to that protocol identifier. If it finds one, this message is logged.

<i>Parameters</i>	1	Remote Site name (alias)
	2	Connection identifier
	3	Address related to the protocol identifier returned
	4	User data related to the protocol identifier returned
	5	Route name related to the protocol identifier returned
	6	Address found in the incoming connection
	7	User data found in the incoming connection
	8	Route name related to the incoming connection (network security file record name)
<i>Action</i>	Possible causes: • The connection is not allowed • The security files are not correctly set up • The Gateway database is not correctly set up Modify the security files or the Gateway database. Try to find out if that transfer is allowed. Contact Axway Support if some other cause is involved.	
022	NDSP E SECUIP %1(%2) IP security error: returned site doesn't match calling address call ad="%3"/"%4"/"%5" [wait ad="%6"/"%7"/"%8"]	
<i>Type</i>	Error	
<i>Meaning</i>	An incoming transfer request has been rejected by the monitor. When the incoming connection arrives, Gateway finds the related record in the security files. If the flag parameter does not contain CTRL value, Gateway does not compare the protocol identifier returned by the remote node. However, it checks if there is a record in the network security file that corresponds to that protocol identifier. If it finds one, this message is logged.	

<i>Parameters</i>	1	Remote Site name (alias)
	2	Connection identifier
	3	Address related to the protocol identifier returned
	4	User data related to the protocol identifier returned
	5	Route name related to the protocol identifier returned
	6	Address found in the incoming connection
	7	User data found in the incoming connection
	8	Route name related to the incoming connection
<i>Action</i>	Possible causes: • The connection is not allowed • The security files are not correctly set up • The Gateway database is not correctly set up Modify the security files or Gateway database. Try to find out if that transfer is allowed. Contact Axway Support if some other cause is involved.	
023	NDSP E BADPROTO connection request rejected: physical site name %1 doesn't exist for protocol %2	
<i>Type</i>	Error	
<i>Meaning</i>	The Site found in the network security files contains a file transfer protocol that does not match the configuration found in the Gateway configuration file.	
<i>Parameters</i>	1	Remote Site name (alias)
	2	Type of file transfer protocol found in the Gateway configuration file
<i>Action</i>	Modify the port number (IP) in the Remote Site definition to match those found in the Gateway configuration file.	
024	NDSP E BADTABLE loading %1 secu file problem: %2 (line %3)	
<i>Type</i>	Error	
<i>Meaning</i>	A syntax error was found while loading a network security file.	

Parameters	1	Type of network security file (IP)
	2	Problem identification message
	3	Line number where the problem was encountered
Action	Modify the security files to respect the correct syntax.	
025	NDSP E REFMON connection request refused by the monitor	
Type	Error	
Meaning	A network connection request has been rejected by the Gateway monitor. This message is logged when a memory problem or a time-out occurs.	
Parameters	None	
Action	Contact Axway Support.	
026	NDSP I REFRESH refresh %1 net secu file by %2: %3 records --> %4 records	
Type	Information	
Meaning	Reload of the network security files was made by using <code>pe1ct1</code> command line to take into account the changes.	
Parameters	1	Type of network security file (IP)
	2	User name
	3	Previous number of records
	4	New number of records
Action	None	
027	NDSP I LOADTAB loading %1 net secu file: %2 records	
Type	Information	
Meaning	First load of the network security files when Gateway is starting.	

Parameters	1	Type of network security file (IP)
	2	Number of records
Action	None	
028	NDSP I ROUT connection indication routed to %1	
Type	Information	
Meaning	A network connection indication has been routed to the selected file transfer procedure.	
Parameters	1	The file transfer procedure to which the connection is forwarded
Action	None	
050	NET I TLSOK (%1) handshake success, version %2, cipher suite %3, secure renegotiation %4	
Type	Information	
Meaning	Indicates that a TLS handshake was achieved. The connection is now secured. The TLS parameters are negotiated according to the remote connection end requirements and the security profiles defined in SECS.	
Parameters	1	Connection identifier
	2	TLS protocol version: 3.0 or 3.1
	3	Local cipher suite in use
	4	Y or N
Action	None	
051	NET I TLSOKCAC (%1) cache success, cipher suite %2, version %3, secure renegotiation %4	
Type	Information	
Meaning	The element was found in the cache.	

<i>Parameters</i>	1	Connection identifier
	2	Cipher suite used
	3	TLS protocol version
	4	Y or N
<i>Action</i>	None	

052 NET I TLSCLNTF (%1) close notify exchanged

<i>Type</i>	Information	
<i>Meaning</i>	Indicates that a TLS session closed correctly by exchanging close notify alert messages. This message is a guarantee that both partners successfully closed the session. This is the last exchanged message, and the connection is closed immediately.	
<i>Parameters</i>	1	Connection identifier
<i>Action</i>	None	

053 NET W TLSALSND (%1) alert %2 sent (%3)

<i>Type</i>	Warning	
<i>Meaning</i>	Indicates that a TLS alert was sent. An alert is sent when an error occurs. All these alerts (except "no renegotiation") are considered fatal and are followed by a disconnection.	

<i>Parameters</i>	1	Connection identifier

2

Alert code:

- **0:** Close notify
- **10:** Unexpected message
- **20:** Bad record mac
- **21:** Decryption failed
- **22:** Record overflow
- **30:** Decompression failure
- **40:** Handshake failure
- **41:** No certificate
- **42:** Bad certificate
- **43:** Unsupported certificate
- **44:** Certificate revoked
- **45:** Certificate expired
- **46:** Certificate unknown
- **47:** Illegal parameter
- **48:** Unknown ca
- **49:** Access denied
- **50:** Decode error
- **51:** Decrypt error
- **60:** Export restriction
- **70:** Protocol version
- **71:** Insufficient security (may occur if no cipher suite matches between client and server)
- **80:** Internal error
- **90:** User canceled
- **100:** No renegotiation
- **110:** Unsupported extension
- **111:** Certificate unobtainable
- **112:** Unrecognized name
- **113:** Bad certificate status response
- **114:** Bad certificate hash value
- **115:** Unknown psk identity

3

Alert label

*Action*Check the alert code for more information.

054	NET W TLSALRCV (%1) alert %2 received (%3)	
Type	Warning	
Meaning	Indicates that a TLS alert was received.	
Parameters	1	Connection identifier
	2	Alert code (as for message 053 above)
	3	Alert label
Action	Check the alert code for more information.	
055	NET E TLSERROR (%1) handshake aborted, error %2/%3	
Type	Error	
Meaning	Error during handshake.	
Parameters	1	Connection identifier
	2	Error code
	3	Error sub-code
Action	If the error is related to SECS, check the netprof and sprof definitions in the security server.	
056	NET E TLSINIT (%1) handshake can not be initiated, check configuration	
Type	Error	
Meaning	SECS definitions do not contain a sprof or netprof item suitable for initiating this TLS connection.	
Parameters	1	Connection identifier
Action	If the error is related to SECS, check the netprof and sprof definitions in the security server.	

060	NET I SSHOK (%1) session success
<i>Type</i>	Information
<i>Meaning</i>	SSH session established.
<i>Parameters</i>	1 NET context number
<i>Action</i>	None
061	NET I SSHEND (%1) session ended correctly
<i>Type</i>	Information
<i>Meaning</i>	Session ended correctly.
<i>Parameters</i>	1 NET context number
<i>Action</i>	None
062	NET E SSHINIT (%1) session can not be initiated, check configuration
<i>Type</i>	Error
<i>Meaning</i>	SSH Profile error. The parameters defined in the SSH Profile are not compatible with those of the Remote Site.
<i>Parameters</i>	1 NET context number
<i>Action</i>	Check SSH Profile configuration.
063	NET E SSHERROR (%1) session aborted on error, %2 (error %3/%4)
<i>Type</i>	Error
<i>Meaning</i>	SSH protocol error (internal error).

<i>Parameters</i>	1	NET context number
	2	Message
	3	Error code
	4	Sub-code
<i>Action</i>	Contact Axway Support.	

064	NET E SSHEND (%1) session ended on error, %2 (error %3)	
<i>Type</i>	Error	
<i>Meaning</i>	SSH protocol error.	
<i>Parameters</i>	1	NET context number
	2	Error message
	3	Error code
<i>Action</i>	Contact Axway Support.	

065	NET I SSHINFO (%1) %2	
<i>Type</i>	Information	
<i>Meaning</i>	SSH negotiation message.	
	This message is used at the start of an SSH negotiation to identify the Initiator (remote or host, client or server).	
	The message is also used at the end of an SSH negotiation to indicate that negotiation was successful (otherwise an error message is generated).	
<i>Parameters</i>	1	SSH session number
	2	Message
<i>Action</i>	None	

066	NET W SSHWARN (%1) %2	
<i>Type</i>	Warning	

Meaning	Warning message containing disconnection reason code from client authentication phase.	
Parameters	1	NET context number
	2	Message
Action	Possible causes: • Incorrect authentication parameters (for example, username or password) • The connection is not allowed by remote machine Correct the authentication parameters or, in case of connection not permitted, check remote machine configuration and log.	
069	NET W SSHWARN (%1) %2 Login deadline expired	
Type	Warning	
Meaning	Gateway has cut a connection from a partner, that was not followed by a completed login within a reasonable timeout. This mechanism avoids DOS-type attacks, that attempt to keep one connection slot occupied without proceeding to log in.	
Parameters	1	SSH version string
	2	Connection identifier
Possible causes	• Possible DOS attack • Network delays	
Action	• Monitor calling addresses for the "login timed out" connections. Check for repeated connection attempts not followed by login attempts. • Check network bandwidth.	
070	NET I NSRINF %1(%2) %3	
Type	Information	
Meaning	Generic Axway Secure Relay networking information display.	

Parameters	1	Mode: • Transfer phase: ◦ Client ◦ Server • Initialization phase
	2	NET context number
	3	Explanatory text message
Action	None	
071	NET W NSRWRN %1(%2) %3	
Type	Warning	
Meaning	Generic Axway Secure Relay networking warning display.	
Parameters	1	Mode: • Transfer phase: ◦ Client ◦ Server • Initialization phase
	2	NET context number
	3	Explanatory text message
Action	None	
072	NET E NSRERR %1(%2) %3	
Type	Error	
Meaning	Generic Axway Secure Relay networking error display.	

Parameters	1	Mode: • Transfer phase: ◦ Client ◦ Server • Initialization phase
	2	NET context number
	3	Explanatory text message
Action	Analyze the text message to determine action.	
073	NET E NSRXSR %1(%2) %3	
Type	Error	
Meaning	[Message not currently used]	
101	NET I CONN_REQ (%1) outgoing connection request [src_add="%2",dest_add="%3"]	
Type	Information	
Meaning	The primitive for outgoing connection requests has been called up.	
Parameters	1	Connection identifier (number)
	2	Network source address (calling)
	3	Network destination address (called)
Action	None	
102	NET I CONN_IND (%1) incoming connection indication [src_add="%2", dest_add="%3"]	
Type	Information	
Meaning	The primitive that processes incoming connection requests has been called up.	

Parameters	1	Connection identifier (number)
	2	Network source address (calling)
	3	Network destination address (called)
Action	None	
103	NET I CONN_RESP (%1) incoming connection response [resp_add="%2"]	
Type	Information	
Meaning	The primitive for responding to incoming connection requests has been called up.	
Parameters	1	Connection identifier (number)
	2	Network address of the responder
Action	None	
104	NET I CONN_CONF (%1) outgoing connection confirmation [resp_add="%2"]	
Type	Information	
Meaning	The primitive that processes the confirmation of connection requests has been called up.	
Parameters	1	Connection identifier (number)
	2	Network address of the responder
Action	None	
105	NET I DISC_REQ (%1) disconnection request [reason="%2"]	
Type	Information	
Meaning	The primitive to send a connection disconnection has been called up.	

<i>Parameters</i>	1	Connection identifier (number)
	2	Disconnection reason
<i>Action</i>	None	
106	NET I DISC_IND (%1) disconnection indication [reason="%2"] [origin="%3"]	
<i>Type</i>	Information	
<i>Meaning</i>	A disconnect indication has been received and the corresponding primitive called up.	
<i>Parameters</i>	1	Connection identifier (number)
	2	Disconnection reason
	3	Origin • 0: not set • 1: not used • 2: socket error or cannot connect to proxy
<i>Action</i>	None	
107	NET I RES_REQ (%1) reset request [reason="%2"]	
<i>Type</i>	Information	
<i>Meaning</i>	A connection reset has been initiated and the corresponding primitive called up.	
<i>Parameters</i>	1	Connection identifier (number)
	2	Reason for the connection reset
<i>Action</i>	None	
108	NET I RES_IND (%1) reset indication [reason="%2"]	
<i>Type</i>	Information	

Meaning	A connection reset has been received.	
Parameters	1	Connection identifier (number)
	2	Reason for the connection reset
Action	None	
109	NET I RES_RESP (%1) reset response	
Type	Information	
Meaning	A connection reset response has been sent and the corresponding primitive called up.	
Parameters	1	Connection identifier (number)
Action	None	
110	NET I RES_CONF (%1) reset confirmation	
Type	Information	
Meaning	The confirmation of the connection reset has been received and the corresponding primitive called up.	
Parameters	1	Connection identifier (number)
Action	None	
111	NET S DATA_REQ (%1) data transmission request [size="%2"]	
Type	Success	
Meaning	A data transmission primitive has been called up.	
Parameters	1	Connection identifier (number)
	2	Size of the buffer transmitted
Action	None	

112	NET S DATA_IND (%1) incoming data indication [size="%2"]	
Type	Success	
Meaning	A data reception primitive has been called up.	
Parameters	1	Connection identifier (number)
	2	Buffer size received
Action	None	
113	NET S EXP_REQ (%1) urgent data request [size="%2"]	
Type	Success	
Meaning	Indicates that urgent data transmission primitive has been called up.	
Parameters	1	Connection identifier (number)
	2	Buffer size transmitted
Action	None	
114	NET S EXP_IND (%1) urgent data indication [size="%2"]	
Type	Success	
Meaning	Indicates that urgent data reception primitive has been called up.	
Parameters	1	Connection identifier (number)
	2	Buffer size received
Action	None	
115	NET I CONN_ROUTE outgoing connection request routed: host=%1 address=%2	
Type	Information	
Meaning	An outgoing connection request has been routed to a destination address.	

Parameters	1	Destination host
	2	Destination address
Action	None	
116	NET I LIST_REQ (%1) listen connection request [dest_add="%2"]	
Type	Information	
Meaning	A dynamic listening connection is asked for FTP data transfer (for list or file data).	
Parameters	1	Connection identifier (number)
	2	TCP/IP waiting address without port number
Action	None	
117	NET I LIST_CONF (%1) listen connection confirmation (%4) [dest_address="%2/%3"]	
Type	Information	
Meaning	A dynamic listening connection is set for FTP data transfer (for list or file data).	
Parameters	1	Connection identifier (number)
	2	TCP/IP waiting address without port number
	3	Listen port number
	4	Listen connection identifier for data
Action	None	
118	NET I TLS_REQ (%1) TLS handshake request	
Type	Information	
Meaning	The primitive to start the underlying TLS layer is called.	

Parameters	1	Connection identifier (number)
Action	None	
119	NET I TLS_RESP (%1) TLS handshake response	
Type	Information	
Meaning	The primitive for responding to incoming TLS start indication is called.	
Parameters	1	Connection identifier (number)
Action	None	
120	NET I TLS_IND (%1) TLS handshake indication	
Type	Information	
Meaning	A TLS start indication was received and the corresponding primitive called.	
Parameters	1	Connection identifier (number)
Action	None	
121	NET I TLS_CONF (%1) TLS handshake confirmation	
Type	Information	
Meaning	The primitive that processes the confirmation of TLS start request was called.	
Parameters	1	Connection identifier (number)
Action	None	
300	NET I PX_LOAD Proxy %1 successfully loaded	
Type	Information	

Meaning	The requested proxy was found and loaded.	
Parameters	1	Proxy name
Action	None	
301	NET I PX_CNX (%1) Successfully connected to proxy %2	
Type	Information	
Meaning	The connection to the requested proxy was successful.	
Parameters	1	Connection identifier (number)
	2	Proxy name
Action	None	
310	NET E PX_ERR Could not access proxy %1: error %2	
Type	Error	
Meaning	An error occurred while loading the requested proxy.	
Parameters	1	Proxy name
	2	Error message
Action	Check the NET proxy field in the Remote Site, and the proxy database, according to the error message.	
311	NET E PX_ERR Proxy %1 is not a NET proxy	
Type	Error	
Meaning	The proxy required in the NET proxy field of the Remote Site is not a NET proxy.	
Parameters	1	Proxy name
Action	Check the NET proxy field in the Remote Site, and the proxy database. The proxy required in the field may be an FT proxy.	

312	NET E PX_ERR (%1) Protocol error while connecting to proxy %2: %3	
<i>Type</i>	Error	
<i>Meaning</i>	A protocol error occurred while negotiating the connection with the proxy.	
<i>Parameters</i>	1	Connection identifier (number)
	2	Proxy name
	3	Error message
<i>Action</i>	Check the proxy configuration according to the error message.	

Axway Gateway: Messages and codes

OFTP protocol log messages (ODT)

This topic lists OFTP log messages.

100	ODT E ERROR %1(%2) %3	
<i>Type</i>	Error	
<i>Meaning</i>	An error condition has been detected. This message is used to report system errors (Input/Output, memory, and network) or errors that cannot be explained by the protocol diagnostic code set.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local number of the transfer (0 if not available)
	4	Error text
<i>Action</i>	For system errors, refer to your platform documentation. For other errors, check protocol or application parameters.	

101	ODT I TRACE %1	
<i>Type</i>	Information	

Meaning	Use this message to trace OFTP protocol data units sent to the Remote Site. The FPDU can be traced in text mode (field name and value) and/or in dump mode (hexadecimal and ASCII code).	
Parameters	1	Protocol Data Unit (PDU)
Action	None	
200	ODT I CONN_REQ %1(%2) connection established with %3	
Type	Information	
Meaning	The protocol and physical connection on the requester side have been established.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local Site name
Action	None	
201	ODT I CONN_IND %1(%2) connection established with %3"	
Type	Information	
Meaning	The protocol and physical connection on the requester side have been established.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local Site name
Action	None	
202	ODT I SEND_REQ %1(%2) [%3] beginning of transmission, file name:%4, date:%5, time:%6	
Type	Information	

Meaning	Indicates on the sender side the beginning of the file transmission. This message comprises two lines.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
Action	None	
203	ODT I SEND_REQ %1(%2) [%3] beginning of transmission from %4, file: %5	
Type	Information	
Meaning	Indicates on the sender side the beginning of the file transmission. This message comprises two lines.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Local Site name
	6	File name in file-system
Action	None	
204	ODT I XEND_SND %1(%2) [%3] end of transmission, file name: %4, date: %5, time: %6	
Type	Information	

Meaning	Indicates on the sender side the end of the file transmission. This message comprises two lines.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
Action	None	
205	ODT I XEND_SND %1(%2) [%3] end of transmission from%4, file: %5	
Type	Information	
Meaning	Indicates on the sender side the end of the file transmission. This message comprises two lines.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
	5	File name in file-system
Action	None	
206	ODT W XERR_SND %1(%2) [%3] error of transmission, file name: %4, date: %5, time: %6	
Type	Warning	
Meaning	Indicates on the requester side an error during the file transmission. This message comprises three lines.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
Action	Check protocol or application parameters and file-system free space.	
207	ODT W XERR_SND %1(%2) [%3] error of transmission from %4, file: %5	
Type	Warning	
Meaning	Indicates on the requester side an error during the file transmission. This message comprises three lines.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
	5	File name in file-system
Action	Check protocol or application parameters and file-system free space.	
208	ODT W XERR_SND %1(%2) [%3] error of transmission [reason=%4]	
Type	Warning	
Meaning	Indicates on the requester side an error during the file transmission. This message comprises three lines.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Reason for error in text and decimal code
<i>Action</i>	Check protocol or application parameters and file-system free space.	

209	ODT I RECV_REQ %1(%2) [%3] begin of reception, file name:%4, date:%5, time:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side the beginning of the file transmission. This message comprises two lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
<i>Action</i>	None	

210	ODT I RECV_REQ %1(%2) [%3] begin of reception for %4, file: %5	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side the beginning of the file transmission. This message comprises two lines.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
	5	File name in file-system
<i>Action</i>	None	

211	ODT I XEND_RCV %1(%2) [%3] end of reception, file name:%4, date:%5, time:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side the end of the file transmission. This message comprises two lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
<i>Action</i>	None	

212	ODT I XEND_RCV %1(%2) [%3] end of reception for %4, file: %5	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side the end of the file transmission. This message comprises two lines.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
	5	File name in file-system
<i>Action</i>	None	

213	ODT W XERR_RCV %1(%2) [%3] error of reception, file name: %4, date:%5, time:%6	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates on the receiver side an error during the file transmission. This message comprises three lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
<i>Action</i>	Check protocol or application parameters and file-system free space.	

214	ODT W XERR_RCV %1(%2) [%3] error of reception for %4, file: %5	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates on the receiver side an error during the file transmission. This message comprises three lines.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
	5	File name in file-system
<i>Action</i>	Check protocol or application parameters and file-system free space.	

215	ODT W XERR_RCV %1(%2) [%3] error of reception [reason="%4"]	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates on the receiver side an error during the file transmission. This message comprises three lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Reason for error in text and decimal code
<i>Action</i>	Check protocol or application parameters and file-system free space.	

216	ODT I DISC_REQ %1(%2) disconnect request send by %3 [reason="%4"]	
<i>Type</i>	Information	
<i>Meaning</i>	The protocol and physical connection on the speaker side have been closed.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local Site name
	4	Reason for disconnection in text and decimal code
<i>Action</i>	None	

217	ODT I DISC_IND %1(%2) disconnect request received for %3 [reason="%4"]	
<i>Type</i>	Information	
<i>Meaning</i>	The protocol and physical connection on the listener side have been closed.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local Site name
	4	Reason for disconnection in text and decimal code
<i>Action</i>	None	
218	ODT I XACK_REQ %1(%2) [%3] send file acknowledge, file name:%4, date:%5, time:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the sender side that an acknowledgment for a previously-received file has been sent. This message comprises two lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
<i>Action</i>	None	
219	ODT I XACK_REQ %1(%2) [%3] send file acknowledge from %4	
<i>Type</i>	Information	

<i>Meaning</i>	Indicates on the sender side that an acknowledgment for a previously-received file has been sent. This message comprises two lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
<i>Action</i>	None	

220	ODT I XACK_IND %1(%2) [%3] receive file acknowledge, file name:%4, date:%5, time:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side that an acknowledgment for a previously sent file has been received. This message comprises two lines.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	OFTP file data set name
	5	Transfer Request creation date
	6	Transfer Request creation time
<i>Action</i>	None	

221	ODT I XACK_IND %1(%2) [%3] receive file acknowledge for %4	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side that an acknowledgment for a previously-sent file has been received. This message comprises two lines.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Local Site name
<i>Action</i>	None	

222	ODT I CHDR_REQ %1(%2) change direction send by "%3"	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the sender side that a Change Direction request has been sent.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local Site name
<i>Action</i>	None	

223	ODT I CHDR_IND %1(%2) change direction recv for %3	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates on the receiver side that a Change Direction request has been received.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local Site name
<i>Action</i>	None	

224	ODT I XRSF_REQ %1(%2) [%3] restart of transmission at record:%4	

<i>Type</i>	Information	
<i>Meaning</i>	Indicates the restart record number on the sender side.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Record number
<i>Action</i>	None	

225	ODT I XRRF_IND %1(%2) [%3] restart of reception at record:%4	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates the restart record number on the receiver side.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Record number
<i>Action</i>	None	

Payload integrity

Diagnostic codes

Diagnostic codes related to data integrity, that are set as `last_end_diag` on transfers failed due to data integrity errors:

- 0 No error
- 1 Failed to load signature information
- 2 Failed to save signature information

- 3 Failed to verify payload signature
- 4 Failed to build payload signature

Errors

Errors reported by the data integrity helper process:

100	INT E KEYERR Couldn't load key from file '%1'	
Type	Error	
Meaning	Either the private key file or the public key file could not be loaded.	
Parameters	1	Path to the file the helper attempted to load
Action	Check that the file contains a rsa key in pem format. In case the file is password protected, make sure the password file is generated an the configuration points to it.	

101	INT W NOCONF Missing configuration entry '%1'	
Type	Warning	
Meaning	Missing value for a configuration parameter	
Parameters	1	The name of the parameter that is undefined

Axway Gateway: Messages and codes

PEL protocol (PEL)

This topic lists PEL log messages.

001	PEL E SELOP_ERR %1(%2) [%3] select-open error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file selection or open operations. The local file system error is returned in parameter %6.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Transfer Request and of the Application used for the compatibility of the file characteristics with the file system attributes referenced in the transfer. Check the access security and make sure that Gateway has the open-read privilege for the file. If a system-dependent parameter is requested for file access, check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
002	PEL E CREOP_ERR %1(%2) [%3] create-open error, file %4/%5: %6	
Type	Error	
Meaning	Error on the file creation or open operations for incoming file transmission. The local file system error is returned in parameter %6.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the "write" privilege in the directory defined (in the Application) for the creation of the files in reception. If a system-dependent parameter is requested for file access, make sure that it is used. There must be enough disk space to receive the file. Refer to your platform documentation to understand the local file system error.	

003	PEL E CLOUDS_ERR %1(%2) [%3] close-deselect error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file closing or de-selection operations. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. Refer to your platform documentation to understand the local file system error.	

004	PEL E READ_ERR %1(%2) [%3] read error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Read error encountered on the file to send. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the "read" privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	

005	PEL E WRITE_ERR %1(%2) [%3] write error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Write error encountered on the received file. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the Application used by Gateway for the file reception, and make sure that the file system attributes are consistent. Gateway must have the "write" privilege in the directory defined (in the Application) for the creation of the files in reception. If a systems dependent parameter is requested for file access, you must make sure that it is actually used. There must be enough disk space to receive the file. Refer to your platform documentation to understand the local file system error.	
006	PEL E UREAD_ERR %1(%2) [%3] previous record read error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Read error encountered on the file to send. The local file system error is returned in parameter %6.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the "read" privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
007	PEL E SEEK_ERR %1(%2) [%3] position error, file %4/%5: %6	
Type	Error	
Meaning	File positioning error encountered on the file to send. The local file system error is returned in parameter %6.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. If a systems dependent parameter is requested for file access, you must check that the Transfer Request contains their description. Refer to your platform documentation to understand the local file system error. Contact Axway Support if the problem persists.	

010	PEL E TRN_FILE_XFER %1(%2) [%3] send format conv. error %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Character set conversion or formatting error on the file to send. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the attributes used by character set conversion and formatting modules in the parameters of the Application or the Transfer Request. The corresponding character set conversion table must exist. Refer to your platform documentation to understand the local file system error. Contact Axway Support if the problem persists.	
011	PEL E TRN_XFER_FILE %1(%2) [%3] receive format conv. error %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Character set conversion or formatting error on the received file. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic

<i>Action</i>	Check the attributes used by character set conversion and formatting modules in the parameters of the Application or the Transfer Request. The corresponding character set conversion table must exist. Refer to your platform documentation to understand the local file system error. Contact Axway Support if the problem persists.

020	PEL E DECOMP %1(%2) [%3] receive data not respecting compression rules, file %4/%5

<i>Type</i>	Error

<i>Meaning</i>	The Local Site cannot uncompress the received data.

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name

<i>Action</i>	Contact Axway Support if the problem persists.

021	PEL I SENT %1(%2) [%3] %4

<i>Type</i>	Information

<i>Meaning</i>	This message is used to trace PEL protocol data units sent to the Remote Site.

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Protocol data unit

<i>Action</i>	None

022	PEL I RECEIVED %1(%2) [%3] %4

<i>Type</i>	Information	
<i>Meaning</i>	This message is used to trace PEL protocol data units received from the Remote Site.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Protocol data unit
<i>Action</i>	None	

100 PEL I TRACE %1

<i>Type</i>	Information	
<i>Meaning</i>	General format for PEL protocol data unit dump.	
<i>Parameters</i>	1	Field for one line of dump
<i>Action</i>	Not applicable	

131 PEL I CONN_REQ %1(%2) [%3] connection established %4

<i>Type</i>	Information	
<i>Meaning</i>	PEL session established in Initiator mode.	
<i>Parameters</i>	1	Remote Site name
	2	Number of the internal context associated with the session
	3	Identification of the Transfer Request
	4	Local request identifier (number)
<i>Action</i>	None	

132 PEL I CONN_IND %1(%2) [%3] connection established %4

Type	Information	
Meaning	PEL session established in server mode.	
Parameters	1	Remote Site name
	2	Number of the internal context associated with the session
	3	Identification of the Transfer Request
	4	Local request identifier (number)
Action	None	
134	PEL I CNAK_REQ %1(%2) [%3] connection refused by %4. reason=%5	
Type	Information	
Meaning	PEL session by the Local Site.	
Parameters	1	Remote Site name
	2	Number of the internal context associated with the session
	3	Identification of the Transfer Request
	4	Local request identifier (number)
	5	Refusal reason
Action	Check the reason.	
135	PEL I CNAK_IND %1(%2) [%3] connection refused to %4. reason=%5	
Type	Information	
Meaning	PEL session to the Local Site.	

<i>Parameters</i>	1	Remote Site name
	2	Number of the internal context associated with the session
	3	Identification of the Transfer Request
	4	Local request identifier (number)
	5	Refusal reason
<i>Action</i>	Check the reason.	

136	PEL I SEND_REQ %1(%2) [%3] beginning of transmission from %4 to %5	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating a beginning of transmission.	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
<i>Action</i>	None	

137	PEL I SEND_REQ %1(%2) [%3] beginning of transmission, appli:%4, file: %5 %	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating beginning of transmission.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	File name
<i>Action</i>	None	

138	PEL I RECV_REQ %1(%2) [%3] beginning of reception from %4 to %5	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating beginning of transmission.	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
<i>Action</i>	None	

139	PEL I RECV_REQ %1(%2) [%3] beginning of reception, appli:%4, file: %5 %6	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating beginning of transmission.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	File name
Action	None	
141	PEL I XEND_SND %1(%2) [%3] end of transmission from %4 to %5	
Type	Information	
Meaning	Information message indicating an end of transmission.	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	None	
142	PEL I XEND_SND %1(%2) [%3] end of transmission, appli:%4, file: %5 %6	
Type	Information	
Meaning	Information message indicating an end of transmission.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	File name
<i>Action</i>	None	

143	PEL I XEND_RCV %1(%2) [%3] end of reception from %4 to %5	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an end of reception.	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
<i>Action</i>	None	

144	PEL I XEND_RCV %1(%2) [%3] end of reception, appli:%4, file: %5 %6	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an end of reception.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	File name
<i>Action</i>	None	

145	PEL E DIAG_REQ %1(%2) [%3] DIAG. sent:%4	
<i>Type</i>	Error	
<i>Meaning</i>	Error diagnostic sent to a Remote Site.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer context
	4	Error diagnostic
<i>Action</i>	Check the diagnostic value.	

146	PEL E DIAG_IND %1(%2) [%3] DIAG. received:%4	
<i>Type</i>	Error	
<i>Meaning</i>	Error diagnostic received from a Remote Site.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer context
	4	Error diagnostic
<i>Action</i>	Check the diagnostic value.	

147	PEL E DIAG_REQ %1(%2) [%3] DIAG. sent:%4	
Type	Error	
Meaning	Timeout reached while expecting a response from the Remote Site.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer context
Action	Contact Axway Support.	
148	PEL I XRDL_REQ %1(%2) [%3] restart, record:%4	
Type	Information	
Meaning	Transfer restarted by Local Site.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer context
	4	Record number to restart from
Action	None	
149	PEL I XRDL_IND %1(%2) [%3] restart, record:%4	
Type	Information	
Meaning	Transfer restarted by Remote Site.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer context
	4	Record number to restart from

<i>Action</i>	None

153	PEL I LOTS_REQ %1(%2) [%3] request for file sent from %4
<i>Type</i>	Information
<i>Meaning</i>	PEL LOTS command sent to an adjacent Site.
<i>Parameters</i>	1 Remote Site name
	2 Connection identifier (number)
	3 Identification of the transfer context
	4 Local request identifier (number)
<i>Action</i>	None

154	PEL I LOTS_IND %1(%2) [%3] request for file received on %4
<i>Type</i>	Information
<i>Meaning</i>	PEL LOTS command received from an adjacent Site.
<i>Parameters</i>	1 Remote Site name
	2 Connection identifier (number)
	3 Identification of the transfer context
	4 Local request identifier (number)
<i>Action</i>	None

155	PEL I LOTS_END %1(%2) [%3] end of file-list for %4
<i>Type</i>	Information
<i>Meaning</i>	PEL LL command successfully recorded.

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer context
	4	Local request identifier (number)
<i>Action</i>	None	

156	PEL E XERR_SND %1(%2) [%3] error of transmission from %4 to %5, reason=%6	
<i>Type</i>	Error	
<i>Meaning</i>	Transmission error detected.	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	Identification of the sender (Site name)
	5	Identification of the receiver (Site name)
	6	Reason on end of transmission (0 is successful)
<i>Action</i>	None	

157	PEL E XERR_SND %1(%2) [%3] error of transmission, appli:%4, file: %5 %6	
<i>Type</i>	Error	
<i>Meaning</i>	Transmission error detected.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Directory path of the file
	6	File name
Action	None	
158	PEL E XERR_RCV %1(%2) [%3] error of reception from %4 to %5, reason=%6	
Type	Error	
Meaning	Reception error detected.	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver (Site name)
	5	Identification of the receiver (Site name)
	6	Reason on end of transmission (0 is successful)
Action	None	
159	PEL E XERR_RCV %1(%2) [%3] error of reception, appli:%4, file: %5 %6	
Type	Error	
Meaning	Reception error detected.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	File type
	5	Directory path of the file
	6	File name
<i>Action</i>	None	

Axway Gateway: Messages and codes

PeSIT HS protocol (FPSIT)

This topic lists PeSIT HS log messages.

001	FPSIT E SEL_ERR %1(%2) [%3] select error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file selection operation	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the access security and path to make sure that Gateway can access the file. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	

002	FPSIT E SELOP_ERR %1(%2) [%3] select-open error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file selection or open operations. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the Transfer Request and of the Application used for the compatibility of the file characteristics with the file system attributes referenced by the transfer. Check the access security and path to make sure that Gateway can access the file. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
003	FPSIT E CRE_ERR %1(%2) [%3] create error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file creation operations for incoming transmission. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic

<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is used. You must have enough space on your disk to receive the file. Refer to your platform documentation to understand the local file system error.	
004	FPSIT E CREOP_ERR %1(%2) [%3] create-open error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file creation or open operations for incoming file transmission. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is used. You must have enough space on your disk to receive the file. Refer to your platform documentation to understand the local file system error.	
005	FPSIT E DESEL_ERR %1(%2) [%3] deselect error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file de-selection operation. The local file system error is returned in parameter %6	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error	

006	FPSIT E OPEN_ERR %1(%2) [%3] open error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file opening operation. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error. Check the parameters of the Transfer Request and of the Application used for the compatibility of the file characteristics with the file system attributes referenced by the transfer. Check the access security and path to make sure that Gateway can access the file. If a system dependent parameter is requested for file access, you must check that the request contains its description.	
007	FPSIT E CLOSE_ERR %1(%2) [%3] close error, file %4 %5: %6	
<i>Type</i>	Error	

<i>Meaning</i>	Error on the file closing operation. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error	

008	FPSIT E CLODS_ERR %1(%2) [%3] close-deselect error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file closing or de-selection operations. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error	

009	FPSIT E READ_ERR %1(%2) [%3] read error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Read error encountered on the file to send. The local file system error is returned in parameter %6	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>read</i> privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
010	FPSIT E WRITE_ERR %1(%2) [%3] write error, file %4 %5: %6	
Type	Error	
Meaning	Write error encountered on the received file. The local file system error is returned in parameter %6	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used by Gateway for the file reception, and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is actually used. There must be enough disk space to receive the file. Refer to your platform documentation to understand the local file system error.	

011	FPSIT E UREAD_ERR %1(%2) [%3] previous record read error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Read error encountered on the file to send. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>read</i> privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
012	FPSIT E SEEK_ERR %1(%2) [%3] position error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	File positioning error encountered on the file to send. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic

<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. If a system dependent parameter is requested for file access, you must check that the Transfer Request contains their description. Refer to your platform documentation to understand the local file system error. Contact Axway Support if the problem persists.	
013	FPSIT E FLUSH_ERR %1(%2) [%3] forced write error, file %4 %5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Forced writing failed on the received file. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error. Contact Axway Support if the problem persists.	
014	FPSIT W COMP_ERR %1(%2) [%3] received data incorrectly compressed, file %4 %5	
<i>Type</i>	Error	
<i>Meaning</i>	The Local Site cannot uncompress the received data	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
Action	Check that the data transmitted on the network by the Remote Site is compressed according to PeSIT normalized rules. The same compression mode must be used on both emission and reception sides. Contact Axway Support if the problem persists.	
015	FPSIT I TRACE %1	
Type	Information	
Meaning	This message is used to trace PeSIT protocol data units sent to the Remote Site	
Parameters	1	Variable area for editing one line of data dump
Action	Not applicable	
020	FPSIT E ABORT_IND %1(%2) [%3] ABORT received: %4 %5	
Type	Error	
Meaning	Abort received on PeSIT session	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Type of error reported
	5	Reason for the abort
Action	If the problem is a recurring one, check the Remote and Local Site definitions. If the abort occurs in data phase, check the transfer parameters used for the Transfer Request.	

021	FPSIT E ABORT_REQ %1(%2) [%3] ABORT sent : %4 %5	
Type	Error	
Meaning	Abort sent to PeSIT session	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Type of error reported
	5	Reason for the abort
Action	If the problem is a recurring one, check the Remote and Local Site definitions. If the abort occurs in data phase, check the transfer parameters used for the Transfer Request.	
022	FPSIT E IDT_IND %1(%2) [%3] IDT received D:%4 %5, R:%6, N=%7	
Type	Error	
Meaning	IDT received/sent on PeSIT session	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Type of error reported
	5	PeSIT diagnostic code
	6	Reason for the abort
	7	File name
Action	Parameters 4, 5, 6 and 7 are described in the PeSIT protocol specifications. For more information, contact Axway Support.	

023	FPSIT E IDT_IND %1(%2) [%3] IDT received D:%4 %5, R:%6, N=%7	
Type	Error	
Meaning	IDT received/sent on PeSIT session	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Type of error reported
	5	Reason for the abort
	6	File name
	7	Internal use
Action	Parameters 4, 5 and 6 are described in the PeSIT protocol specifications. For more information, contact Axway Support.	
024	FPSIT I CONN_IND %1(%2) [%3] connection request received for %4	
Type	Information	
Meaning	Information message meaning that a PeSIT connection indication has been received	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Local request identifier (number)
Action	None	
025	FPSIT I CONN_REQ %1(%2) [%3] connection request sent by %4	
Type	Information	

Meaning	Information message meaning that a PeSIT connection indication has been sent	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Local request identifier (number)
Action	None	
026	FPSIT I XFER_RCV1 %1(%2) [%3] begin receiving to %4, file:%5 %6	
Type	Information	
Meaning	Information message indicating start of reception	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	Received file name
Action	None	
027	FPSIT I XFER_RCV2 %1(%2) [%3] begin receiving, TYPE:%4, ID:%5, N:%6	
Type	Information	
Meaning	Information message indicating start of reception	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	Protocol file name
Action	None	
028	FPSIT I XFER_SND1 %1(%2) [%3] begin sending from %4, file:%5 %6	
Type	Information	
Meaning	Information message indicating a beginning of transmission	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	None	
029	FPSIT I XFER_SND2 %1(%2) [%3] begin sending from %4, file:%5 %6	
Type	Information	
Meaning	Information message indicating a beginning of transmission	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	Protocol file name
<i>Action</i>	None	

030	FPSIT I XEND_RCV1 %1(%2) [%3] end receiving to %4, file:%5 %6	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an end of reception	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
<i>Action</i>	None	

031	FPSIT I XEND_RCV2 %1(%2) [%3] end receiving, TYPE:%4, ID:%5, N:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an end of reception	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	Protocol file name
Action	None	
032	FPSIT I XEND_SND1 %1(%2) [%3] end sending from %4, file:%5 %6	
Type	Information	
Meaning	Information message indicating an end of transmission	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	None	
033	FPSIT I XEND_SND2 %1(%2) [%3] end sending, TYPE: %4, ID:%5, N:%6	
Type	Information	
Meaning	Information message indicating an end of transmission	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	Protocol file name
<i>Action</i>	None	

034	FPSIT E XERR_IND %1(%2) [%3] error receiving to %4, file: %5 %6	
<i>Type</i>	Error	
<i>Meaning</i>	Reception error	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	

035	FPSIT E XERR_IND %1(%2) [%3] error receiving, TYPE:%4, ID:%5, N:%6	
<i>Type</i>	Error	
<i>Meaning</i>	Reception error	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	Protocol file name
Action	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	

036	FPSIT E XERR_REQ %1(%2) [%3] error sending from %4, file: %5 %6	
Type	Error	
Meaning	Error sending a file	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	

037	FPSIT E XERR_REQ %1(%2) [%3] error sending, TYPE: %4, ID:%5 %6	
Type	Error	
Meaning	Error sending a file	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	File type
	5	Transfer identifier
	6	Protocol file name
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	
038	FPSIT I RSTR_IND %1(%2) [%3] resume, sync:%4, record:%5, bytes:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating that a restart indication has been received in PeSIT protocol	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Check point number
	5	Record number
	6	Byte offset
<i>Action</i>	None	
039	FPSIT I CONN_ABT %1(%2) [%3] connection interrupted with %4	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating a connection abort with no data transfer phase	

<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Remote Site name
<i>Action</i>	Check the message indicating the reason for the abort	

040	FPSIT I SELECT_E1 %1(%2) [%3] SELECT fails: TYPE:%4, ID:%5, N:%6	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating a file selection request failure	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Protocol file type
	5	Protocol connection ID
	6	Protocol file name
<i>Action</i>	See notes in the following message	

041	FPSIT I SELECT_E2 %1(%2) [%3] SELECT fails: %4	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating a file selection request failure	
<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Diagnostic code and reason of the failure

Action	If the diagnostic code is "205:file not found", no particular action is required. Otherwise, check PeSIT protocol documentation for an explanation of the protocol error encountered.	
042	FPSIT I CREATE_E1 %1(%2) [%3] CREATE fails: TYPE:%4, ID:%5, N:%6	
Type	Information	
Meaning	Information message indicating a file creation failure	
Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the Transfer Request
	4	Protocol file type
	5	Protocol connection ID
	6	Protocol file name
Action	See notes in the following message	
043	FPSIT I CREATE_E2 %1(%2) [%3] CREATE fails: %4	
Type	Information	
Meaning	Information message indicating a file creation failure	
Parameters	1	Identification of the sender
	2	Internal context number associated with the transfer
	3	Identification of the Transfer Request
	4	Diagnostic code and reason of the failure
Action	Check PeSIT protocol documentation for an explanation of the protocol error encountered	

051	FPSIT I MSG_SND2 %1(%2) [%3] begin sending %4, TYPE:%5, ID:%6, N:%7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating a file creation failure	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	MSG/REPLY
	5	File type
	6	Transfer identifier
	7	Protocol file name
<i>Action</i>	None	

052	FPSIT I MEND_SND2 %1(%2) [%3] end sending %4, TYPE:%5, ID:%6, N:%7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an end of transmission	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	MSG/REPLY
	5	File type
	6	Transfer identifier
	7	Protocol file name
<i>Action</i>	None	

053	FPSIT I MSG_RCV2 %1(%2) [%3] begin receiving %4, TYPE:%5, ID:%6, N:%7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating a start of message reception	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	MSG/REPLY
	5	File type
	6	Transfer identifier
	7	Protocol file name
<i>Action</i>	None	

054	FPSIT I MEND_RCV2 %1(%2) [%3] end receiving %4, TYPE:%5, ID:%6, N:%7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an end of reception	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	MSG/REPLY
	5	File type
	6	Transfer identifier
	7	Protocol file name
<i>Action</i>	None	

055	FPSIT I CONN_NACK %1(%2) [%3] invalid connection request: unsupported option %4	
Type	Information	
Meaning	An incoming connection request was refused because it asked for an unsupported option	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The unsupported option in question: it can either be negative acknowledgment or Change Direction for PeSIT protocol
Action	Configure these options via the Remote Site definition	
101	FPSIT I TRACE %1	
Type	Information	
Meaning	This message is used to trace PeSIT protocol data units sent to the Remote Site	
Parameters	1	Remote Site name
Action	Not applicable	
102	FPSIT I CDI_SND %1(%2) [%3] Change of direction sent. Heartbeat interval: %4	
Type	Information	
Meaning	Information message indicating that a change of direction request has been sent	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Request number
	4	Heartbeat message time interval (in seconds)
Action	Not applicable	

103	FPSIT I CDI_RCV %1(%2) [%3] Change of direction received. Heartbeat interval: %4	
Type	Information	
Meaning	Information message indicating that a change of direction request has been received	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Request number
	4	Heartbeat message time interval (in seconds)
Action	Not applicable	
104	FPSIT I ACDI_SND %1(%2) [%3] Change of direction accepted. Heartbeat interval: %4	
Type	Information	
Meaning	Information message indicating that a change of direction request has been accepted	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Request number
	4	Heartbeat message time interval (in seconds)
Action	Not applicable	
105	PSIT I ACDI_RCV %1(%2) [%3] Change of direction confirmed. Heartbeat interval: %4	
Type	Information	
Meaning	Information message confirming change of direction	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Request number
	4	Heartbeat message time interval (in seconds)
<i>Action</i>	Not applicable	

106	FPSIT E ABORT_REQ %1(%2) [%3] ABORT sent : %4 %5	
<i>Type</i>	Error	
<i>Meaning</i>	PeSIT error triggers Gateway ABORT request	
<i>Parameters</i>	1	Remote Site name
	2	Connector ID
	3	Request number
	4	PeSIT error code
	5	Internal error code
<i>Action</i>	The corrective action depends on the returned error code: • To interpret PeSIT error codes, refer to the PeSIT topics in this documentation, or alternatively, refer to PeSIT standards documentation. • To correct errors referenced by internal error codes, contact Axway Support.	

107	FPSIT E ABORT_IND %1(%2) [%3] ABORT received : %4 %5	
<i>Type</i>	Error	
<i>Meaning</i>	ABORT request received following error	

Parameters	1	Remote Site name
	2	Connector ID
	3	Request number
	4	PeSIT error code
	5	Internal error code
Action	The corrective action depends on the returned error code: • To interpret PeSIT error codes refer to the PeSIT topics in this documentation, or alternatively, refer to PeSIT standards documentation. • To correct errors referenced by internal error codes, contact Axway Support.	
110	FPSIT W HA_DUP %1(%2) Transfer is a duplicate (PI3: %3 PI4: %4 PI12: %5 PI13: %6 PI51: %7)	
Type	Warning	
Meaning	A duplicate incoming transfer was detected.	
Parameters	1	Remote Site name
	2	Connector ID
	3	PI3 (originator)
	4	PI4 (destination)
	5	PI12 (file name)
	6	PI13 (transfer identifier)
	7	PI51 (file creation date)
111	FPSIT E HA_ERROR %1 Reason: %2	
Type	Error	
Meaning	An error occurred during a PeSIT Duplicate Checking in the context of High Availability operation.	

<i>Parameters</i>	1	Error message
	2	Error reason
112	FPSIT E HA_DCERR %1(%2) Duplicate check failed (PI3: %3 PI4: %4 PI12: %5 PI13: %6 PI51: %7). Reason: %8	
<i>Type</i>	Error	
<i>Meaning</i>	The duplicate check failed for an incoming transfer. The transfer will continue.	
<i>Parameters</i>	1	Error message
	2	Error reason
	3	PI3 (originator)
	4	PI4 (destination)
	5	PI12 (file name)
	6	PI13 (transfer identifier)
	7	PI51 (file creation date)
	8	Error reason
113	FPSIT E HA_STERR %1(%2) Transfer DB store failed, duplicate check will not be available (PI3: %3 PI4: %4 PI12: %5 PI13: %6 PI51: %7). Reason: %8	
<i>Type</i>	Error	
<i>Meaning</i>	Storing the transfer parameters in the High Availability database failed. Duplicate check will not be available for this transfer.	
<i>Parameters</i>	1	Remote Site name
	2	Connector ID
	3	PI3 (originator)
	4	PI4 (destination)

5	PI12 (file name)
6	PI13 (transfer identifier)
7	PI51 (file creation date)
8	Error reason

Axway Gateway: Messages and codes

Post Office Protocol (POP3)

This topic lists POP3 log messages.

001	POP I CONN_REQ %1(%2) [%3] connection request: user=%4	
Type	Information	
Meaning	Message indicating that the user issued a connection request	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The login user ID
Action	None	
002	POP I CONN_ACK %1(%2) [%3] connection established. %4	
Type	Information	
Meaning	The POP3 server has accepted the connection request	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the transfer request
	4	The welcome message as sent by the POP3 server
Action	None	

003	POP I USER_ACK %1(%2) [%3] login accepted. %4	
Type	Information	
Meaning	The POP3 server has accepted the login request	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The reply message as sent by the POP3 server
Action	None	
004	POP E USER_NAK %1(%2) [%3] login refused. %4	
Type	Error	
Meaning	The POP3 server has refused the login request	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The reply message as sent by the POP3 server
Action	None	
010	POP I RECV_CNT %1(%2) [%3] %4 mails available	
Type	Information	
Meaning	Message indicating the number of emails available	

<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The mail count as sent by the POP3 server
<i>Action</i>	None	

011	POP I RECV_IND %1(%2) [%3] beginning of reception From:%4 To:%5	
<i>Type</i>	Information	
<i>Meaning</i>	Message indicating the beginning of a mail reception cycle	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The sender of the email
	5	The receiver of the email
<i>Action</i>	None	

012	POP I RECV_IND %1(%2) [%3] beginning of reception, mailtype=%4, appli=%5, file=%6	
<i>Type</i>	Information	
<i>Meaning</i>	Message indicating the beginning of a mail reception cycle	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The type of mail (XFER, MAIL, DATA or REPORT)
	5	The name of the Axway application used

<i>Action</i>	None

013	POP I RECV_END %1(%2) [%3] end of reception From:%4 To:%5
<i>Type</i>	Information
<i>Meaning</i>	Message indicating the end of a mail reception cycle
<i>Parameters</i>	1 Remote Site name
	2 Internal context number associated with the transfer
	3 Local identification of the Transfer Request
	4 The sender of the mail
	5 The receiver of the mail
<i>Action</i>	None

014	POP I RECV_END %1(%2) [%3] end of reception, mailtype=%4, appli=%5, file=%6
<i>Type</i>	Information
<i>Meaning</i>	Message indicating the end of a mail reception cycle.
<i>Parameters</i>	1 Remote Site name
	2 Internal context number associated with the transfer
	3 Local identification of the Transfer Request
	4 The type of mail (XFER, MAIL, DATA or REPORT)
	5 The name of the Axway application used
<i>Action</i>	None

015	POP E RECV_ERR %1(%2) [%3] error of reception From:%4 To:%5, reason=%6

Type	Error	
Meaning	Error message indicating that an error occurred during the mail reception phase.	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The sender of the mail
	5	The receiver of the mail
	6	Text describing the nature of the error
Action	None	
016	POP E RECV_ERR %1(%2) [%3] error of reception, mailtype=%4, appli=%5, file=%6	
Type	Error	
Meaning	Error message indicating that an error occurred during the mail reception phase.	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The type of mail (XFER, MAIL, DATA or REPORT)
	5	The name of the Axway application used
Action	None	
017	POP E XFER_ERR %1(%2) [%3] transfer refused. duplicate transfer. From:%4 To:%5	
Type	Error	
Meaning	A duplicated mail was detected during the mail reception phase.	

<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The sender of the mail
	5	The receiver of the mail
<i>Action</i>	None	

018	POP E XFER_ERR %1(%2) [%3] transfer refused. From:%4 To:%5 reason=%6	
<i>Type</i>	Error	
<i>Meaning</i>	Error message indicating that mail reception was aborted due to a detected error	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The sender of the mail
	5	The receiver of the mail
	6	Text describing the nature of the error
<i>Action</i>	None	

019	POP I TERM_IND %1(%2) [%3] Session terminated by remote site	
<i>Type</i>	Information	
<i>Meaning</i>	The POP3 server terminated the reception session. Message indicating that the reception session is terminated by the POP3 server.	

Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
Action	None	
020	POP I TERM_REQ %1(%2) [%3] Session terminated by local site	
Type	Information	
Meaning	The POP3 client terminated the reception session. Message indicating that the reception session is terminated by the POP3 client	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
Action	None	
021	POP I XFER_ACK %1(%2) ACK: xfer=%3 from=%4 to=%5	
Type	Information	
Meaning	Message indicating that a mail acknowledgment was received	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	The mail identification
	4	The sender of the mail
	5	The receiver of the mail
Action	None	
022	POP I XFER_ACK %1(%2) ACK: %3 (%4)	

Type	Information	
Meaning	Message indicating that a mail acknowledgment was received	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	acknowledgment type
	4	acknowledgment action
Action	None	
023	POP W XFER_REC %1(%2) [%3] Start recovery. reason=%4, diag=%5, err=%6	
Type	Warning	
Meaning	Message indicating the start of the recovery process for a previous mail reception failure	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The supervisor refusal reason
	5	The last POP3 diagnostic code
	6	The last system error code
Action	None	
024	POP W XFER_REC %1(%2) [%3] Error or duplicate. Mail left on server	
Type	Warning	
Meaning	Message indicating that an error or duplicated mail was detected and left intact on the POP3 server	

Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
Action	None	

025	POP W XFER_REC %1(%2) [%3] Error or duplicate. Mail deleted from server	
Type	Warning	
Meaning	An error or duplicated mail has been detected and then deleted from the POP3 server	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
Action	None	

030	POP E ERORR %1(%2) [%3] memory allocation error. reason=%4	
Type	Error	
Meaning	Error message indicating that a system memory allocation API error was detected	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	Text describing the failure context
Action	None	

031	POP E ERROR %1(%2) [%3] SFM error. reason=%4, diag=%5, errno=%6	
<i>Type</i>	Error	
<i>Meaning</i>	Error message indicating that a system file management API error was detected	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	Text describing the failure context
	5	Diagnostic code
	6	System error code
<i>Action</i>	None	

032	POP E ERROR %1(%2) [%3] timeout	
<i>Type</i>	Error	
<i>Meaning</i>	Error message indicating that an inactivity timeout occurred	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
<i>Action</i>	None	

033	POP E ERROR %1(%2) [%3] Transfer aborted by local site	
<i>Type</i>	Error	
<i>Meaning</i>	The mail reception process was aborted due to an unrecoverable error	

Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
Action	None	
034	POP E ERROR %1(%2) [%3] Transfer aborted by SUP: %4	
Type	Error	
Meaning	The mail reception process was aborted on Supervisor request	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	Text describing the abort reason
Action	None	
035	POP E ERROR %1(%2) [%3] internal protocol error	
Type	Error	
Meaning	The mail reception process was aborted due to an unrecoverable POP3 protocol error	
Parameters	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
Action	None	
036	POP E ERROR %1(%2) [%3] %4	
Type	Error	

<i>Meaning</i>	The mail reception process was aborted due to an unrecoverable MIME parsing error	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	Text detailing the MIME parser error context
<i>Action</i>	None	

037 POP E ERROR %1(%2) [%3] recovery fails

<i>Type</i>	Error	
<i>Meaning</i>	The attempted recovery process failed. The mail will either be deleted from or left intact on the POP3 server, depending on the Remote Site definition	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
<i>Action</i>	None	

040 POP I TRACE %1(%2) [%3] CMMD: %4

<i>Type</i>	Information	
<i>Meaning</i>	Trace message for the POP3 protocol exchange	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The command text as sent by the client
<i>Action</i>	None	

041	POP I TRACE %1(%2) [%3] RESP: %4	
<i>Type</i>	Information	
<i>Meaning</i>	Trace message for the POP3 protocol exchange	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The response text as sent by the server
<i>Action</i>	None	

042	POP I TRACE %1(%2) [%3] DATA: %4	
<i>Type</i>	Information	
<i>Meaning</i>	Trace message for the POP3 protocol exchange	
<i>Parameters</i>	1	Remote Site name
	2	Internal context number associated with the transfer
	3	Local identification of the Transfer Request
	4	The data text as sent by the server
<i>Action</i>	None	

Axway Gateway: Messages and codes

Process Monitoring messages (GC)

This topic lists Process Monitoring log messages generated by the GateController (GC) process.

001	GC S IGLOB Initialization of GateController DONE	
<i>Type</i>	Success	
<i>Meaning</i>	Indicates that the initialization of GateController was successful.	
<i>Parameters</i>	None	

<i>Action</i>	None

002	GC E INITGLERR Initialization of GateController failed
<i>Type</i>	Error
<i>Meaning</i>	Indicates that initialization of GateController has failed.
<i>Parameters</i>	None
<i>Action</i>	Activate GateController traces and check the trace output in <code>p_runtime_dir/tmp/GATECONTROLLER.out</code> .

003	GC S IENV Initialization of GateController environment DONE
<i>Type</i>	Success
<i>Meaning</i>	Indicates that the initialization of GateController configuration environment variables was successful.
<i>Parameters</i>	None
<i>Action</i>	None

004	GC E IENVERR Configuration file missing
<i>Type</i>	Error
<i>Meaning</i>	Indicates that the GateController environment initialization returned an error.
<i>Parameters</i>	None
<i>Action</i>	Verify the <code>conffile</code> (<code>p_config_file</code> environment variable).

005	GC I ICTX The context of GateController is updated (state : %1 ; pid : %2 ; number of Gateway restart : %3)
<i>Type</i>	Information
<i>Meaning</i>	GateController internal memory context has been updated.

Parameters	1	Internal state
	2	GateController pid
	3	Number of registered Gateway restarts
Action	None	
006	GC S IPMT Process monitoring tools are initialized	
Type	Success	
Meaning	Indicates that the initialization of GateController internal monitoring tools was successful.	
Parameters	None	
Action	None	
008	GC S IGWT The Gateway process tree has been initialized successfully	
Type	Success	
Meaning	Indicates that the initialization of the Gateway process tree was successful.	
Parameters	None	
Action	None	
009	GC E IGWTERR Error during initialization of Gateway process tree	
Type	Error	
Meaning	Indicates that the initialization of the Gateway process tree caused an error.	
Parameters	None	
Action	Activate GateController traces and check the trace output in p_runtime_dir/tmp/GATECONTROLLER.out. Check the [gatecontroller] information in p_runtime_dir/etc/pelsetup.ini.	

010	GC I IPT The heartbeat timer has been initialized	
Type	Information	
Meaning	Indicates the initialization of the heartbeat mechanism.	
Parameters	None	
Action	None	
011	GC I IGWS The Gateway status has been initialized	
Type	Information	
Meaning	Indicates the initialization of the internal status.	
Parameters	None	
Action	None	
012	GC S IDEAM GateController TCP Daemon is initialized on %1:%2	
Type	Success	
Meaning	Indicates the initialization of the GateController TCP daemon.	
Parameters	1	Host name
	2	Port number
Action	None	
013	GC E IDEAMERR GateController TCP Daemon initialization on %1:%2 failed	
Type	Error	
Meaning	Indicates that the initialization of the GateController TCP daemon failed.	
Parameters	1	Host name
	2	Port number
Action	Verify that the port is not already used.	

014	GC I STARTMON The monitor will start	
Type	Information	
Meaning	Indicates that the monitor will be started.	
Parameters	None	
Action	None	
015	GC W PBSTATUS Problematic Gateway status detected : %1	
Type	Warning	
Meaning	Indicates that GateController detected a problematic Gateway status.	
Parameters	1	Status
Action	Check the next message in the log.	
016	GC W POL1 GateController detected the status %2 caused by %1. GateController will restart the Gateway	
Type	Warning	
Meaning	Indicates that the policy for Broken (vital process) status will be applied: restart of Gateway.	
Parameters	1	Name of process that caused the problem
	2	Status
Action	Check the log and your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.	
017	GC W POL2 GateController detected the status %2 caused by %1. GateController will restart %1	
Type	Warning	

<i>Meaning</i>	Indicates that the policy for Broken (daemon process) status will be applied: restart of the %1 daemon.	
<i>Parameters</i>	1	Name of process that caused the problem
	2	Status
<i>Action</i>	Check the log and your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.	
018	GC W POL3 GateController detected the status %2 caused by %1 bad stop. %1 will normally be restart by the supervisor	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates that Broken (dynamic process) status has been detected.	
<i>Parameters</i>	1	Name of process that caused the problem
	2	Status
<i>Action</i>	Check the log and your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.	
019	GC W POL4 GateController detected the status %2 caused by %1 bad stop. %1 will normally be restart by the supervisor	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates that Broken (dynamic process) status has been detected.	
<i>Parameters</i>	1	Name of process that caused the problem
	2	Status
<i>Action</i>	Check the log and your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.	

020	GC W POL5 GateController detected the status %2 caused by %1 bad stop. %1 will normally be restart by the supervisor	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates that Broken (dynamic process) status has been detected.	
<i>Parameters</i>	1	Name of process that caused the problem
	2	Status
<i>Action</i>	Check the log and your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.	
021	GC I HBCHKX No heartbeat has been received from %1 since %2*%3 seconds	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates that no heartbeat has been received from a process for more than x*y seconds (y<5).	
<i>Parameters</i>	1	Process name
	2	x (heartbeat frequency)
	3	y
<i>Action</i>	None	
022	GC W HBCHKERR No heartbeat has been received from %1 since more %2*5 seconds... %1 seems dead...	
<i>Type</i>	Warning	
<i>Meaning</i>	Indicates that no heartbeat has been received from a process for more than x*5 seconds.	
<i>Parameters</i>	1	Process name
	2	x (heartbeat frequency)

<i>Action</i>	Check the log message corresponding to process name %1 and check your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.	
023	GC I REQRCV GateController receive a request (socket num. %1 / host : %2)	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates the reception of a request on a socket.	
<i>Parameters</i>	1	Socket number
	2	Host name
<i>Action</i>	None	
024	GC E REQRCVERR Error during receive of a request (socket num. %1) : %2	
<i>Type</i>	Error	
<i>Meaning</i>	Indicates that there was an error when receiving a request on a socket.	
<i>Parameters</i>	1	Socket number
	2	Error code
<i>Action</i>	Activate and check the trace for the GateController process. If the problem persists, contact Axway Support.	
025	GC I REQTRT The request of type %3 will be treated (socket num. %1, host : %2)	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates that the request was processed and will be treated.	

<i>Parameters</i>	1	Socket number
	2	Host name
	3	Request type
<i>Action</i>	None	

026	GC E REQTRTERR Error during treatment of request (socket num. %1, host : %2) : %3

<i>Type</i>	Error	
<i>Meaning</i>	Indicates that there was an error when processing a request.	
<i>Parameters</i>	1	Socket number
	2	Host name
	3	Error code
<i>Action</i>	Activate and check the trace for the GateController process. If the problem persists, contact Axway Support.	

027	GC I REQASW GateController sending answer for a command of type %3 (socket num. %1 - host : %2)

<i>Type</i>	Information	
<i>Meaning</i>	Indicates that a reply was sent for an administration command.	
<i>Parameters</i>	1	Socket number
	2	Host name
	3	Request type
<i>Action</i>	None	

028	GC E REQASWERR Error during sending answer for an administration command (socket num. %1 - host : %2) : %3

<i>Type</i>	Error	
<i>Meaning</i>	Indicates that there was an error when sending an answer for an administration command.	
<i>Parameters</i>	1	Socket number
	2	Host name
	3	Error code
<i>Action</i>	Check connection between administration client and GateController. Activate and check the trace for the GateController process. If the problem persists, contact Axway Support.	

029	GC I REQEND Work on socket %1 (host %2) ended	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates that the socket exchange has ended.	
<i>Parameters</i>	1	Socket number
	2	Host name
<i>Action</i>	None	

030	GC I CMDSTOP GateController has received a stop request. Gateway will be halted	
<i>Type</i>	Information	
<i>Meaning</i>	GateController has received a stop request and Gateway will be stopped.	
<i>Parameters</i>	None	
<i>Action</i>	None	

031	GC I CMDSTUS GateController has received a status request	
<i>Type</i>	Information	
<i>Meaning</i>	GateController has received a status request.	

<i>Parameters</i>	None
<i>Action</i>	None
032	GC I PNTSTART GateController has received a paternal notification for %1 task start
<i>Type</i>	Information
<i>Meaning</i>	Task %1 has been started.
<i>Parameters</i>	1 Task name
<i>Action</i>	None
033	GC I PNTSTOPO GateController has received a paternal notification for %1 task normal stop
<i>Type</i>	Information
<i>Meaning</i>	Task %1 has been stopped without error.
<i>Parameters</i>	1 Task name
<i>Action</i>	None
034	GC I PNTSTOPP GateController has received a paternal notification for %1 task problematic stop
<i>Type</i>	Information
<i>Meaning</i>	Task %1 has been stopped with error.
<i>Parameters</i>	1 Task name
<i>Action</i>	Check log message concerning %1. Check your configuration. Check trace file for task %1 or check your external script. If the problem persists, contact Axway Support.

035	GC I HBRCV GateController has received a heartbeat from %1	
Type	Information	
Meaning	Indicates that a heartbeat notification has been received from task %1.	
Parameters	1	Task name
Action	None	
036	GC I NEWSTUS GateController detected a modification of Gateway status caused by %1 (%2)	
Type	Information	
Meaning	Indicates that the Gateway status has changed.	
Parameters	1	Process name
	2	State of process
Action	Check the previous message and the next message in the log. Check your configuration. Check the trace file for process %1. If the problem persists, contact Axway Support.	
037	GC I NEWCON GateController accepted a new connection (socket %1)	
Type	Information	
Meaning	Indicates that a socket connection request has been accepted.	
Parameters	1	Socket number
Action	None	
038	GC I NEWCONERR Error during a new connection	
Type	Information	
Meaning	Indicates that there was an error when making a new connection.	

<i>Parameters</i>	None
<i>Action</i>	Activate GateController traces and check the trace output in <code>p_runtime_dir/tmp/GATECONTROLLER.out</code> . If the problem persists, contact Axway Support.
039	GC I END_GC GateController and Gateway are terminated
<i>Type</i>	Information
<i>Meaning</i>	Indicates the end of the GateController process and therefore of Gateway.
<i>Parameters</i>	None
<i>Action</i>	None
040	GC E NOMOREGC GateController detected an awkward Gateway restart loop. GateController will be disabled at the next Gateway restart. You should check the backup made before
<i>Type</i>	Error
<i>Meaning</i>	Indicates that an error was detected during GateController restart.
<i>Parameters</i>	None
<i>Action</i>	The backup directory will give more information about the reason of the restart. It is also possible the GateController has been deactivated during startup sequence.
041	GC W NBUDONE A new backup has been made in the backup directory to keep the reasons of the Gateway restart
<i>Type</i>	Warning
<i>Meaning</i>	A snapshot of the Gateway state has been made in the backup directory.
<i>Parameters</i>	None
<i>Action</i>	Check the backup directory to get more information about the product restart.

042	GC I PNTSTARTR GateController should received soon a paternal notification for %1 task stop	
Type	Information	
Meaning	Task %1 has been stopped and the process that launched it should notify GateController.	
Parameters	1	Task name
Action	None	
043	GC I PNTSTARTC The %1 task has been restarted after a crash. GateController will update the Gateway status	
Type	Information	
Meaning	GateController has been notified of a process restart after a crash.	
Parameters	1	Task name
Action	None	
044	GC I PNTSTOPOR GateController has received a paternal notification for %1 task normal stop. This process has already been restarted	
Type	Information	
Meaning	Task %1 has been stopped properly.	
Parameters	1	Task name
Action	None	
045	GC I PNTSTOPPR GateController has received a paternal notification for %1 task problematic stop. This process has already been restarted	
Type	Information	
Meaning	Task %1 has been stopped unexpectedly.	

Parameters	1	Task name
Action	None	
047	GC E ERETSIPE GateController pid(%1) unable to restart the server [CSTCP]. Already have reached the gc_proc_max_restart	
Type	Error	
Meaning	TCP server daemon has exceeded its maximum amount of restarts and won't be restarted.	
Parameters	1	System process identifier
Action	Check TCP server daemon process output	
048	GC E ERETSXFB GateController pid(%1) unable to restart the server [CSIUI]. Already have reached the gc_proc_max_restart	
Type	Error	
Meaning	IUI server daemon has exceeded its maximum amount of restarts and will not be restarted.	
Parameters	1	System process identifier
Action	Check IUI server daemon process output	
050	GC W JOBSIG Process %1 pid %2 terminated on signal %3	
Type	Warning	
Meaning	The process has terminated after having received a signal.	
Parameters	1	Task name
	2	System process identifier
	3	Signal
Action	The process is likely to have stopped on its own. Check the log or the output file if it is a Gateway process.	

051	GC W JOBSTOP Process %1 pid %2 stopped on signal %3	
Type	Warning	
Meaning	The process has crashed after having received a signal.	
Parameters	1	Task name
	2	System process identifier
	3	Signal
Action	The process is likely to have stopped on its own. Check the log, the output or a core file if it is a Gateway process.	
052	GC W JOBEXIT Process %1 pid %2 exited with status %3	
Type	Warning	
Meaning	The process has exited with a code.	
Parameters	1	Task name
	2	System process identifier
	3	Exit status
Action	A status that is not null is likely to indicate that the process is giving an error warning. You should check its output.	
065	GC E GCSRCONN GateController was unable to connect to SecureRelay Master Agent	
Type	Error	
Meaning	Indicates that there was an error when trying to connect to Secure Relay Master Agent.	
Parameters	None	
Action	Check Secure Relay trace file and Secure Relay Master Agent Log to find the cause of the issue.	

066	CG I POLOFF GateController detected the status %2 caused by %1. GateController restart policy is disabled.
<i>Type</i>	Information
<i>Meaning</i>	Indicates that the policy for handling broken statuses caused by vital processes is disabled. This prevents the Gatecontroller from applying the default policy for handling such statuses, which is to restart the entire Gateway application.
<i>Parameters</i>	1 Name of process that caused the problem 2 Status
<i>Action</i>	Check the log and your configuration. Activate and check the trace for process %1. If the problem persists, contact Axway Support.

Axway Gateway: Messages and codes

Protocol Connection (PCNX)

This topic lists Protocol Connection log messages.

001	PCNX I SEL (%1) CGate %2 selected
<i>Type</i>	Information
<i>Meaning</i>	A CGate object matching the incoming connection has been found and selected. Several verifications are to be processed according to this CGate definition. Note: This message is only displayed if the value of the configuration variable <code>cgate_log_level</code> is greater than or equal to 1.
<i>Parameters</i>	1 Connection identifier (number)
	2 Selected CGate object name
<i>Action</i>	None
002	PCNX E CF (%1) Connection Failure, reason: %2
<i>Type</i>	Error

<i>Meaning</i>	The protocol connection through the CGate object has failed and the connection is rejected. Note: This message is displayed whatever the value of the configuration variable <code>cgate_log_level</code>.	
<i>Parameters</i>	1	Connection identifier (number)
	2	Connection failure reason (string) which can take one of the following values: • No CGate matching the connection • Invalid FTP login password • Password expired • Password change forbidden • Password syntax invalid • New and old passwords are the same • Invalid HTTP login password • Invalid PHSE pre-connection login • Invalid PHSE pre-connection password • Invalid PHSE pre-connection client password • Invalid ODT client password • Invalid PEL client password • Invalid TLS security profile • Invalid TLS security user parameter • Insufficient TLS client authentication
<i>Action</i>	Refer to CGates: Protocol connection phase for more details about the protocol connection mechanisms and check your configuration.	
003	PCNX I CONN (%1) %2 Connection Request Received	
<i>Type</i>	Information	
<i>Meaning</i>	An incoming connection request has been received. Note: This message is only displayed if the value of the configuration variable <code>cgate_log_level</code> is greater than or equal to 2.	
<i>Parameters</i>	1	Connection identifier (number)
	2	Connection protocol type which can be either FTP, HTTP, PEL, ODT, or PHSE

<i>Action</i>	None

004	PCNX I CONN (%1) %2
<i>Type</i>	Information
<i>Meaning</i>	An incoming connection request has been received. This message gives the details of what parameters have been received from the network. This message can be displayed on several lines. Its content is protocol dependent. Note: This message is only displayed if the value of the configuration variable <code>cgate_log_level</code> is greater than or equal to 2.
<i>Parameters</i>	1 Connection identifier (number) 2 Protocol dependent message content: • Common (FTP, HTTP and PHSE) ◦ calling address, calling SAP, called address, called SAP ◦ login user, login password • HTTP specific parameters: HTTP host name • PHSE specific parameters: client ident, server ident • ODT and PEL: client ident
<i>Action</i>	None

005	PCNX I CONN (%1) %2
<i>Type</i>	Information
<i>Meaning</i>	The protocol connection through the CGate object is successful. This message gives information about the output parameters obtained from the selected CGate hierarchy. This message can be displayed on several lines. Note: This message is only displayed if the value of the configuration variable <code>cgate_log_level</code> is greater than or equal to 2.

<i>Parameters</i>	1	Connection identifier (number)
	2	Protocol dependent message content: • Common (FTP, HTTP, ODT, PEL, and PHSE) • Template Site • Route remote agent, route local agent, route destination ident, route originator ident • HTTP and FTP shared parameters • VFD root directory • VFD home directory • HTTP specific parameters • Home page • List template • PHSE specific parameters • Server password
<i>Action</i>	None	

020	020 PCNX I TPM (%1) Entry found in trading partner database for user %2	
<i>Type</i>	Information	
<i>Meaning</i>	Entry found in trading partner database for user	
<i>Parameters</i>	1	Connection identifier (number)
	2	User name found in database
<i>Action</i>	None	

021	021 PCNX E TPM_CF (%1) Connection Failure for user %2 reason : %3	
<i>Type</i>	Error	
<i>Meaning</i>	Connection refused for user specified in parameter 2	

<i>Parameters</i>	1	Connection identifier (number)
	2	User name found in database
	3	Reason for connection failure
<i>Action</i>	Check the reason displayed in parameter 3 and correct the error.	

Axway Gateway: Messages and codes

Reason codes and messages

This topic lists reason codes and messages.

000	No error
<i>Meaning</i>	No error
<i>Action</i>	None

005	No space left for an initiator connection
<i>Meaning</i>	A connection attempt in initiator mode is refused due to max number of connections reached.
<i>Action</i>	Increase max number of connections or wait until connections free.

006	No space left for an responder connection
<i>Meaning</i>	A connection attempt in responder mode is refused due to max number of connections reached.
<i>Action</i>	Increase max number of connections or wait until connections free.

007	Initiator mode not allowed
<i>Meaning</i>	Site is not configured for initiator mode capabilities.
<i>Action</i>	Change Site configuration.

008	Responder mode not allowed
<i>Meaning</i>	Site is not configured for responder mode capabilities.
<i>Action</i>	Change Site configuration.
009	Initiator/sender mode not allowed
<i>Meaning</i>	Site is not configured for initiator/sender mode capabilities.
<i>Action</i>	Change Site configuration.
010	Responder/sender mode not allowed
<i>Meaning</i>	Site is not configured for responder/sender mode capabilities.
<i>Action</i>	Change Site configuration.
011	Initiator/receiver mode not allowed
<i>Meaning</i>	Site is not configured for initiator/receiver mode capabilities.
<i>Action</i>	Change Site configuration.
012	Responder/receiver mode not allowed
<i>Meaning</i>	Site is not configured for responder/receiver mode capabilities.
<i>Action</i>	Change Site configuration.
013	Invalid local identifier
<i>Meaning</i>	The destination Site declared in the transfer is not accepted by Gateway.
<i>Action</i>	Check configuration (local Sites, CGates, exits).

014	Invalid remote identifier	
<i>Meaning</i>	The origin Site declared in the transfer is not accepted by Gateway.	
<i>Action</i>	Check configuration (Sites, CGates, exits).	
017	Data base access error	
<i>Meaning</i>	An access to a Gateway object has failed.	
<i>Action</i>	Retry operation. Check that object has not been removed from the database.	
165	Transfer failure	
<i>Meaning</i>	A failure occurs during the transfer phase.	
<i>Action</i>	Check log for more information.	
166	Connection failure	
<i>Meaning</i>	A connection failure has occurred.	
<i>Action</i>	Check log for more information.	
190	EDI processing error %1	
<i>Meaning</i>	An error in EDI processing occurs.	
<i>Parameter</i>	1	Text message describing the EDI processing error
<i>Action</i>	Action depends on displayed text message.	
191	EDI process abnormal exit	
<i>Meaning</i>	Process terminated in error.	
<i>Action</i>	Check log for more information.	

195	TPM connector processing error %1	
<i>Meaning</i>	An error in PassPort connector processing has occurred.	
<i>Parameter</i>	1	Text message describing the PassPort processing error
<i>Action</i>	Action depends on displayed text message.	

196	Integrity processing error	
<i>Meaning</i>	The transfer was canceled by the data or payload integrity helper	
<i>Action</i>	Check log and last_end_diag parameter of the transfer for more information.	

Axway Gateway: Messages and codes

Routing (NOTIF)

This topic lists routing (NOTIF) log messages.

402	NOT I XEND_CF [%1] Final destination = %2 undefined	
<i>Type</i>	Information	
<i>Meaning</i>	The final destination does not exist.	
<i>Parameters</i>	1	Local transfer identifier
	2	Final destination name
<i>Action</i>	Create the final destination corresponding object.	

403	NOT I XEND_CF [%1] %2 = %3 selected for routing	
<i>Type</i>	Information	
<i>Meaning</i>	A destination, towards which the received file will be transferred, has been selected.	

Parameters	1	Local transfer identifier
	2	Routing according to the selected object. The partner used for the routed transfer is chosen according to one of the following parameters: • <code>dest_alias</code> associated with a Model • <code>route_remote_agent</code> associated with a CGATE • <code>route_remote_agent</code> associated with a directory file (VFD) • <code>route_remote_agent</code> associated with a Site • transfer destination
	3	Selected Remote Site alias
Action	None	

404	NOT E XEND_CF [%1] Routing failure: type = %2, protocol = %3	
Type	Error	
Meaning	Routing failure: the transfer type and the protocol are not compatible.	
Parameters	1	Local transfer identifier
	2	Transfer type
	3	Protocol
Action	None	

405	NOT E XEND_CF [%1] Routing failure: model = %2 not found	
Type	Error	
Meaning	Routing failure: the referenced Model was not found.	
Parameters	1	Local transfer identifier
	2	Model name
Action	Create the referenced Model.	

406	NOT E XEND_CF [%1] Routing failure: no remote_agent selected	
Type	Error	
Meaning	Routing failure: no Remote Site has been selected.	
Parameters	1	Local transfer identifier
Action	Complete the routing definitions.	
407	NOT E XEND_CF [%1] Routing failure: % 2 is not defined	
Type	Error	
Meaning	Routing failure: the selected destination has not been defined.	
Parameters	1	Local transfer identifier
	2	Selected Site alias
Action	Create the corresponding Site.	
408	NOT E XEND_CF [%1] Routing failure: %2 is a local site	
Type	Error	
Meaning	Routing failure: the selected destination is a Local Site.	
Parameters	1	Local transfer identifier
	2	Selected Site alias
Action	Select a Remote Site to route to.	
409	NOT E XEND_CF routing failure: XferRequest error, ErrCod = %1, errno = %2	
Type	Error	
Meaning	Routing failure: the monitor failed during the new transfer request deposit.	

Parameters	1	Local transfer identifier
	2	Corresponding local error (if any)
	3	System specific return code
Action	Analyze why monitor failed during new transfer request deposit.	
410	NOT S XEND_CF [%1] Routing towards %2, ident %3, model %4, route_from_xfer %5	
Type	Success	
Meaning	Routing was successful towards the Site %2	
Parameters	1	Local transfer identifier
	2	Routing Site name
	3	New transfer identifier (number)
	4	Name of the Model used
	5	Transfer identifier that is linked to the new transfer
Action	None	
420	NOT I XEND_CF [%1] Decision rule [%2], condition (%3 %4 %5) = FALSE	
Type	Information	
Meaning	Condition %2 of the Decision Rule is not in agreement with the incoming transfer. Therefore this rule will not be selected.	
Parameters	1	Local transfer identifier
	2	Condition that the incoming transfer does not agree with
	3	Operator
	4	Value
Action	Analyze why the Decision Rule was not selected.	

421	NOT I XEND_CF [%1] Decision rule [%2] selected, execution type = %3	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates the name and execution type of the Decision Rule selected.	
<i>Parameters</i>	1	Local transfer identifier
	2	Decision Rule name
	3	Type of processing that will be applied. One of the following execution types:
		• Application of a Model • Execution of a Perl script or command line • Redirection to AMTriX/ Axway Integrator • None
<i>Action</i>	None	

422	NOT I XEND_CF [%1] Default Model %2 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates the name of the Model referenced at the table level that will be applied if the incoming transfer does not agree with any rule.	
<i>Parameters</i>	1	Local transfer identifier
	2	Name of the Model, defined at the table level, that will be applied
<i>Action</i>	None	

423	NOT I XEND_CF [%1] Model %2 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	Indicates the name of the Model that will be applied as referenced in the selected Decision Rule.	
<i>Parameters</i>	1	Local transfer identifier
	2	Model name

<i>Action</i>	None

424	NOT E XEND_CF [%1] No model is referenced
<i>Type</i>	Error
<i>Meaning</i>	No Model name is referenced at the Decision Rule level or the table level.
<i>Parameters</i>	1 Local transfer identifier
<i>Action</i>	Analyze why the Model cannot be applied.

425	NOT I XEND_NOT I XEND_CF [%1] Decision rule [%2]: after substituting symbolic variable in the condition (%3 %4 %5), incorrect type for the value
<i>Type</i>	Information
<i>Meaning</i>	The Decision Rule does not return the correct type of variable after substituting the symbolic variable.
<i>Parameters</i>	1 Local transfer identifier 2 Decision Rule name 3 Argument 4 Operator 5 Value
<i>Action</i>	Analyze why the Decision Rule will not be selected.

426	NOT I XEND_CF [%1] For Model %2, value %3 is not valid for the parameter %4
<i>Type</i>	Information
<i>Meaning</i>	At the level of the Model %2 that will be applied, the parameter and corresponding value are not compatible.

<i>Parameters</i>	1	Local transfer identifier
	2	Model name
	3	Invalid value
	4	Parameter (the Model parameter that does not allow the defined value)
<i>Action</i>	Analyze why the Model cannot be applied.	

427	NOT E XEND_CF [%1] Not routed, %2 : %3	
<i>Type</i>	Error	
<i>Meaning</i>	Routing was not successful because a condition was not met.	
<i>Parameters</i>	1	Local transfer identifier
	2	Parameter whose condition was not met
	3	Value of the parameter
<i>Action</i>	Analyze why the routing was not successful.	

428	NOT E XEND_CF [%1] Not routed, local_ident and remote_agent are the same	
<i>Type</i>	Error	
<i>Meaning</i>	The origin (source) and destination are the same.	
<i>Parameters</i>	1	Local transfer identifier
<i>Action</i>	Modify either the <code>local_ident</code> or the <code>remote_agent</code> .	

429	NOT E XEND_CF [%1] Not routed, %2 : %3 is a local site	
<i>Type</i>	Error	
<i>Meaning</i>	Routing failure: the selected destination is a Local Site.	

Parameters	1	Local transfer identifier
	2	Object and parameters: • Model remote_agent • Model org_alias • Model dest_alias • CGate route_remote_agent • VFD route_remote_agent • Site route_remote_agent • Destination
	3	Selected Site alias
Action	Select Remote Site to route to.	

430	NOT E XEND_CF [%1] Not routed, %2 : site %3 is not defined	
Type	Error	
Meaning	Routing failure: the selected destination was not defined.	
Parameters	1	Local transfer identifier
	2	Object and parameters: • Model remote_agent • Model org_alias • Model dest_alias • CGate route_remote_agent • VFD route_remote_agent • Site route_remote_agent • Destination
	3	Selected Site alias
Action	Create the corresponding Site.	

431	NOT I XEND_CF [%1] Perl script %2 will be executed	

Type	Information	
Meaning	Indicates the Perl script that will be executed.	
Parameters	1	Local transfer identifier
	2	Absolute path of the Perl script that will be executed
Action	None	
432	NOT E XEND_CF [%1] No perl script is referenced	
Type	Error	
Meaning	No Perl script is referenced at the table level or the level of the Decision Rule selected.	
Parameters	1	Local transfer identifier
Action	Analyze why the Perl script cannot be executed.	
433	NOT I XEND_CF [%1] Default perl script %2 will be executed	
Type	Information	
Meaning	Indicates the default Perl script, referenced at the table level that will be executed.	
Parameters	1	Local transfer identifier
	2	Absolute path of the Perl script, referenced at the table level, that will be executed if no rules satisfy the incoming transfer
Action	None	
434	NOT E XEND_CF [%1] For General Model %2, %3 %4 is not defined	
Type	Error	
Meaning	Indicates that one of the parameters of a General Model is not valid.	

Parameters	1	Local transfer identifier
	2	Name of the General Model
	3	Model parameter that is not defined: • remote_agent • org_alias • dest_alias
	4	Selected Site alias
Action	Analyze why the General Model is not correct.	

435	NOT E XEND_CF [%1] Perl script %2 doesn't exist	
Type	Error	
Meaning	Indicates that the Perl script does not exist.	
Parameters	1	Local transfer identifier
	2	Absolute path of the referenced Perl script
Action	Analyze why the Perl script cannot be executed.	

436	NOT I XEND_CF [%1] The command line %2 will be executed	
Type	Information	
Meaning	Indicates that the specified command will be executed.	
Parameters	1	Local transfer identifier
	2	Command line that will be executed. This command includes an executable program as well as its arguments.
Action	None	

437	NOT E XEND_CF [%1] No command line is referenced	
Type	Error	

Meaning	No command line is referenced at the table level or in the Decision Rule selected.	
Parameters	1	Local transfer identifier
Action	Analyze why the execution of the command line cannot be executed.	
438	NOT I XEND_CF [%1] The default command line %2 will be executed	
Type	Information	
Meaning	Indicates the default command line, referenced at the table level that will be executed if no rules satisfy the incoming transfer.	
Parameters	1	Local transfer identifier
	2	Command line (referenced at the table level) that will be executed. This command includes an executable program as well as these arguments.
Action	None	
439	NOT I XEND_CF [%1] Default Execution type = %2	
Type	Information	
Meaning	Indicates that the default execution type will be activated.	
Parameters	1	Local transfer identifier
	2	Type of processing that will be applied. One of the following execution types: • Application of a Model • Execution of a Perl script or shell • Redirection to AMTriX/Axway Integrator
Action	None	
440	NOT I XEND_CF [%1] Resulting %2 parameter = "%3" in the Model %4	

<i>Type</i>	Information	
<i>Meaning</i>	Provides information on the resolved Model.	
<i>Parameters</i>	1	Local transfer identifier
	2	Parameter name
	3	Parameter value after substitution of the symbolic variable
	4	Name of the referenced Model in the selected Decision Rule
<i>Action</i>	None	

442 NOT I XEND_CF Examining rule table [%1]

<i>Type</i>	Information	
<i>Meaning</i>	Indicates that the Rule Table is being examined.	
<i>Parameters</i>	1	Name of Rule Table
<i>Action</i>	None	

443 [%1] Rule table [%2] , condition (%3 %4 %5) = FALSE

<i>Type</i>	Information	
<i>Meaning</i>	Displays the evaluations done when searching for a decision rule to match.	
<i>Parameters</i>	1	Identifier of the evaluated transfer
	2	The evaluated rule table
	3	Selection criteria
	4	Selection value
	5	Evaluation result
<i>Action</i>	None	

444	Trading partner %1 not found when sending automatic receipt for transfer [%2]	
Type	Error	
Meaning	The trading partner needed for sending the receipt was not found	
Parameters	1	Originating trading partner for incoming transfer
	2	Identifier of the incoming transfer
Action	Check trading partner configuration	
445	[%1] Routing to XIB not performed, no payload file	
Type	Warning	
Meaning	A decision rule is trying to route to XIB a transfer that does not have payload	
Parameters	1	Identifier of the transfer
Action	Check decision rules and do not route to XIB transfers with no payload. They have no meaning for XIB	
690	NOTE E TRINEX Transfer to route %1 does not exist	
Type	Error	
Meaning	Transfer cannot be routed since routing encountered too many errors. The transfer has been purged or deleted before being routed.	
Parameters	1	Local transfer identifier
Action	Avoid purging or deleting transfers before they are routed.	
691	NOTE E TRLOCKED Transfer to route %1 is currently locked	
Type	Error	
Meaning	Transfer cannot be routed since routing encountered too many errors. The transfer is currently in use.	

Parameters	1	Local transfer identifier
Action	Check the transfer state to avoid this problem.	
692	NOTE E TRACCERR Transfer to route %1 access error: %2	
Type	Error	
Meaning	Transfer cannot be routed since routing encountered too many errors.	
Parameters	1	Local transfer identifier
	2	Access error message
Action	The error message should give a hint regarding the routing problem.	
693	NOTE E TRPRMERR Too many routing errors (%2), ignoring transfer %1	
Type	Error	
Meaning	Transfer cannot be routed since routing encountered too many errors.	
Parameters	1	Local transfer identifier
	2	Number of trials
Action	Check the routing rules.	
694	NOTE E XEND_CF [%1] Error while executing the command line %2	
Type	Error	
Meaning	The command line referenced at the table level or in the selected Decision Rule cannot be executed properly.	
Parameters	1	Local transfer identifier
	2	Command line to be executed (this command includes an executable program as well as its arguments)
Action	Check the command line spelling, rights or content.	

Routing Axway Messaging (RXMS)

This topic lists the Axway Messaging routing log messages.

001	RXMS S FUNCT error in function: %1	
Type	Error	
Meaning	Error in a function.	
Parameters	1	Name of the function
Action	Check the log for associated messages.	
002	RXMS E CONN error of connection to XMS:%1 %2	
Type	Error	
Meaning	Error in connecting to Axway Messaging.	
Parameters	1	Axway Messaging error code
	2	Axway Messaging Error message
Action	Read error code and message and consult Axway Messaging documentation for meaning.	
003	RXMS I CONN connection to XMS established	
Type	Information	
Meaning	Successful connection to Axway Messaging.	
Action	None	
004	RXMS E CONN error of disconnection to XMS:%1 %2	
Type	Error	

Meaning	Error in disconnecting from Axway Messaging.	
Parameters	1	Axway Messaging error code
	2	Axway Messaging error message text
Action	Read error code and message and consult Axway Messaging documentation for meaning.	
005	RXMS I CONN connection to XMS finished	
Type	Information	
Meaning	Successful disconnection from Axway Messaging.	
Action	None	
006	RXMS E MODEL model:%1 not found	
Type	Error	
Meaning	The indicated Model was not found.	
Parameters	1	Name of the Model
Action	Verify the Model name in the Decision Rule.	
007	RXMS E PROP property:%1 not found	
Type	Error	
Meaning	The indicated property was not found.	
Parameters	1	Name of the property
Action	Verify that the property exists in the Property List you are using for the transfer.	
008	RXMS E PROP error in field value "%1": %2	

<i>Type</i>	Error
<i>Meaning</i>	Error in evaluating a property field.
<i>Parameters</i>	1 Name of the property
	2 Value
<i>Action</i>	Verify the property definition for the specified field.

009 RXMS E MSG error creating XMS message : %1 %2

<i>Type</i>	Error
<i>Meaning</i>	Error in creating an Axway Messaging message.
<i>Parameters</i>	1 Axway Messaging error code
	2 Axway Messaging error message
<i>Action</i>	Read error code and message and consult Axway Messaging documentation for meaning.

010 RXMS E MSG error creating XMS notification : %1 %2

<i>Type</i>	Error
<i>Meaning</i>	Error in creating an Axway Messaging notification.
<i>Parameters</i>	1 Axway Messaging error code
	2 Axway Messaging error message
<i>Action</i>	Read error code and message and consult Axway Messaging documentation for meaning.

011 RXMS E MSG error depositing XMS message : %1 %2

<i>Type</i>	Error
<i>Meaning</i>	Error depositing an Axway Messaging message.

Parameters	1	Axway Messaging error code
	2	Axway Messaging error message
Action	Read error code and message and consult Axway Messaging documentation for meaning.	
012	RXMS I MSG XMS message sent, id: %1	
Type	Information	
Meaning	Message successfully sent.	
Parameters	1	Message ID
Action	None	
013	RXMS E MSG error committing XMS message : %1 %2	
Type	Error	
Meaning	Error in committing an Axway Messaging message.	
Parameters	1	Axway Messaging error code
	2	Axway Messaging error message
Action	Read error code and message and consult Axway Messaging documentation for meaning.	
014	RXMS E MSG file "%1" is absent	
Type	Error	
Meaning	The file to transfer is absent.	
Parameters	1	File name and path
Action	Verify and correct name and path of file to transfer.	

015	RXMS E MSG file "%1" is too big	
Type	Error	
Meaning	The file to transfer is too big.	
Parameters	1	File name and path
Action	Verify Axway Messaging configuration parameter for the configured file size limit. Maximum possible configurable file size is 128 MB.	
016	RXMS I MSG xms option not activated	
Type	Information	
Meaning	The XMS (Axway Messaging) option is not activated in the Gateway configuration.	
Action	Activate the XMS option.	
017	RXMS I MSG xms option activated	
Type	Information	
Meaning	The XMS (Axway Messaging) option is activated.	
Action	None	
018	RXMS I MSG xms connector "%1" not activated	
Type	Information	
Meaning	The Axway Messaging connector is not activated in the Gateway configuration menu.	
Parameters	1	Connector name
Action	Activate the Axway Messaging connector.	
019	RXMS I MSG xms connector "%1" activated	

<i>Type</i>	Information	
<i>Meaning</i>	The Axway Messaging connector is activated in the Gateway configuration menu.	
<i>Parameters</i>	1	Connector name
<i>Action</i>	None	

Axway Gateway: Messages and codes

Rule Table evaluation

The type of process is listed in the error message and varies depending on the message originator. In the following tables the process is represented by **[process]**.

500	[%1] E SYMBVAR_E [%2] decision rule [%3], symbolic variable (%4) in the condition (%5 %6 %7) is not recognized	
<i>Type</i>	Error	
<i>Meaning</i>	Unknown symbolic variable in condition.	
<i>Parameters</i>	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Symbolic variable
	5	Condition contents - operand
	6	Condition contents - operator
	7	Condition contents - operand
<i>Action</i>	Check that the symbolic variables used are correct for the given Rule Table type.	
501	[%1] E SYMBVAR_E [%2] decision rule [%3], symbolic variable (%4) in the condition (%5 %6 %7) does not have a value	
<i>Type</i>	Error	

Meaning	The symbolic variable in the condition has a void value.	
Parameters	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Symbolic variable
	5	Condition contents - operand
	6	Condition contents - operator
	7	Condition contents - operand
Action	None	
502	[%1] E SYMBVAR_E [%2] decision rule [%3], symbolic variable (%4) in the condition (%5 %6 %7) has an incorrect value, the rule table routing has ended	
Type	Error	
Meaning	There is a symbolic variable with an invalid value in the condition.	
Parameters	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Symbolic variable
	5	Condition contents - operand
	6	Condition contents - operator
	7	Condition contents - operand
Action	None	
503	[%1] E SYNTAX_E [%2] decision rule [%3], the condition (%4,%5,%6) contains a syntax error	

Type	Error	
Meaning	There is a syntax error in the condition.	
Parameters	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Condition contents - operand
	5	Condition contents - operator
	6	Condition contents - operand
Action	Check the condition syntax.	
504		
[%1] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the field %5 for the model %6 is unknown		
Type	Error	
Meaning	There is an unknown symbolic variable in the transfer Model	
Parameters	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Symbolic variable
	5	Model field
	6	Model name
Action	Check that the symbolic variables used are correct for the given the Rule Table type	
505		
[%1] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the field %5 for the model %6 does not have a value		
Type	Error	

Meaning	There is a symbolic variable with a void value in the transfer Model.	
Parameters	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Symbolic variable
	5	Model field
	6	Model name
Action	None	
506	[%1] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the field %5 for the model %6 contains an incorrect value	
Type	Error	
Meaning	There is a symbolic variable with an invalid value in the transfer model	
Parameters	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Symbolic variable
	5	Model field
	6	Model name
Action	None	
507	[%1] E SYNTAX_E [%2] decision rule [%3], the field %4 for the model %5 contains a syntax error	
Type	Error	
Meaning	There is a syntax error in the transfer model	

<i>Parameters</i>	1	Process name
	2	Rule Table name
	3	Decision Rule
	4	Field name
	5	Model name
<i>Action</i>	Check the field syntax	

508	[%1] E SYMBVAR_E [%2] The symbolic variable (%3) used in the field %4 in the default transfer model %5 is unknown	
<i>Type</i>	Error	
<i>Meaning</i>	There is an unknown symbolic variable in the default transfer model	
<i>Parameters</i>	1	Process name
	2	Rule Table name
	3	Symbolic variable
	4	Default Model field
	5	Default Model name
<i>Action</i>	Check that the symbolic variables used are correct for the given the Rule Table type	

509	[%1] E SYMBVAR_E [%2] The symbolic variable (%3) used in the field %4 in the model %5 did not receive a value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with a void value in the default transfer model	

<i>Parameters</i>	1	Rule Table name
	2	Transfer Model (default)
	3	Symbolic variable
	4	Model field name
	5	Model name
<i>Action</i>	None	

510	[%1] E SYMBVAR_E [%2] The symbolic variable (%3) used in the field %4 in the default transfer model %5 received an incorrect value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with an invalid value in the default transfer model	
<i>Parameters</i>	1	Rule Table name
	2	Transfer Model (default)
	3	Symbolic variable
	4	Model field name
	5	Model name
<i>Action</i>	None	

511	[%1] E SYNTAX_E [%2] The field %3 in the default transfer model %4 contains a syntax error	
<i>Type</i>	Error	
<i>Meaning</i>	There is a syntax error in the default transfer model	

Parameters	1	Rule Table name
	2	Transfer Model (default)
	3	Model field name
	4	Model name
Action	Check the syntax of the default transfer model field	

512	[%1] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the command line to execute (%5) is unknown	
Type	Error	
Meaning	There is an unknown symbolic variable in the command line	
Parameters	1	Rule Table name
	2	Decision Rule
	3	Symbolic variable
	4	Command line
Action	Check that the symbolic variables used are correct for the given the Rule Table type	

513	[%1] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the command line to execute (%5) did not receive a value	
Type	Error	
Meaning	There is a symbolic variable with a void value in the command line.	

<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Symbolic variable
	4	Command line
	5	Value
<i>Action</i>	None	

514	[%1] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the command line to execute (%5) received an incorrect value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with an invalid value in the command line.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Symbolic variable
	4	Command line
	5	Value
<i>Action</i>	None	

515	[%1] E SYNTAX_E [%2] decision rule [%3], the command line to execute (%4) contains a syntax error	
<i>Type</i>	Error	
<i>Meaning</i>	There is a syntax error in the command line.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Command line
<i>Action</i>	Check the syntax for the line to execute	

516	[%1] E SYMBVAR_E [%2] the symbolic variable (%3) used in the default command line to execute (%4) is unknown	
<i>Type</i>	Error	
<i>Meaning</i>	There is an unknown symbolic variable in the default command line.	
<i>Parameters</i>	1	Rule Table name
	2	Symbolic variable
	3	Command line
<i>Action</i>	Check the used symbolic variables is correct for the given the Rule Table type.	
517	[%1] E SYMBVAR_E [%2] the symbolic variable (%3) used in the default command line to execute (%4) did not receive a value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with a void value in the default command line.	
<i>Parameters</i>	1	Rule Table name
	2	Symbolic variable
	3	Command line
<i>Action</i>	None	
518	[%1] E SYMBVAR_E [%2] the symbolic variable (%3) used in the default command line to execute (%4) received an incorrect value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with an invalid value in the default command line.	
<i>Parameters</i>	1	Rule Table name
	2	Symbolic variable
	3	Command line
<i>Action</i>	None	

519	[process] E SYNTAX_E [%2] decision rule [%3], the default command line (%4) contains a syntax error	
<i>Type</i>	Error	
<i>Meaning</i>	There is a syntax error in the default command line.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Command line
<i>Action</i>	Check the syntax of the default command line.	
520	[process] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the field %5 in the purge model %6 is unknown	
<i>Type</i>	Error	
<i>Meaning</i>	There is an unknown symbolic variable in the purge Model.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Symbolic variable
	4	Purge Model field name
	5	Model name
<i>Action</i>	Check the used symbolic variables is correct for the given the Rule Table type.	
521	[process] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the field %5 in the purge model %6 did not receive a value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with a void value in the purge Model.	

<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Symbolic variable
	4	Purge Model field name
	5	Model name
<i>Action</i>	None	

522	[process] E SYMBVAR_E [%2] decision rule [%3], the symbolic variable (%4) used in the field %5 in the purge model %6 received an incorrect value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with an invalid value in the purge Model.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Symbolic variable
	4	Purge Model field name
	5	Model
<i>Action</i>	None	

523	[process] E SYNTAX_E [%2] decision rule [%3], the field %4 in the purge model %5 contains a syntax error	
<i>Type</i>	Error	
<i>Meaning</i>	There is a syntax error in the purge Model.	

<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Purge Model field name
	4	Model
<i>Action</i>	Check the syntax of the purge Model field.	

524	[process] E SYMBVAR_E [%2] The symbolic variable (%3) used in the field %4 in the default purge model %5 is unknown	
<i>Type</i>	Error	
<i>Meaning</i>	There is an unknown symbolic variable in the default purge Model.	
<i>Parameters</i>	1	Rule Table name
	2	Symbolic variable
	3	Default purge Model field name
	4	Model name
<i>Action</i>	Check the used symbolic variables is correct for the given Rule Table type.	

525	[process] E SYMBVAR_E [%2] The symbolic variable (%3) used in the field %4 in the default purge model %5 did not receive a value	
<i>Type</i>	Error	
<i>Meaning</i>	There is a symbolic variable with a void value in the default purge Model.	
<i>Parameters</i>	1	Rule Table name
	2	Symbolic variable
	3	Default purge Model field name
	4	Default purge Model name
<i>Action</i>	None	

526	[process] E SYMBVAR_E [%2] The symbolic variable (%3) used in the field %4 in the default purge model %5 received an incorrect value		
Type	Error		
Meaning	Symbolic variable with invalid value in the default purge Model		
Parameters	1	Rule Table name	
	2	Symbolic variable	
	3	Default purge Model field name	
	4	Default purge Model name	
Action	None		
527	[process] E SYNTAX_E [%2] The field %3 in the default purge model %4 contains a syntax error		
Type	Error		
Meaning	Syntax error in the default purge Model		
Parameters	1	Rule Table name	
	2	Default purge Model field name	
	3	Default purge Model name	
Action	Check the syntax of the default purge Model field		
528	[process] E OP_E [%2] In the decision rule [%3] the operator (%4) in the condition (%5 %6 %7) is not correct for this type of operand		
Type	Error		
Meaning	Invalid operator for the type of operands		

Parameters	1	Rule Table name
	2	Decision Rule
	3	Operator
	4	Operand
	5	Operator
	6	Operand
Action	Check that the operator is correct for the type of the operands	
529	[process] E TYPES_E [%2] in decision rule [%3], the left operand and the right operand in the condition (%4 %5 %6) cannot be compared because they are not the same type	
Type	Error	
Meaning	Incompatible type for operands	
Parameters	1	Rule Table name
	2	Decision Rule
	3	Operand
	4	Operator
	5	Operand
Action	Check that the type of operand is correct.	
530	[process] E OPERAND_E [%2] decision rule [%3], the number of arguments to the right of the operator (%4) in the condition (%5 %6 %7) is incorrect	
Type	Error	
Meaning	Incorrect number of right operands.	

<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Operator
	4	Operand
	5	Operator
	6	Operand
<i>Action</i>	Check and change the condition.	

531	[process] E EX_TYPE_E [%2] decision rule [%3], the task to execute is %4 type while the rule table is %5 type. This configuration is incorrect.	
<i>Type</i>	Error	
<i>Meaning</i>	Incompatible execution type in Decision Rule task.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Task type
	4	Rule Table type
<i>Action</i>	Check and change the Decision Rule.	

532	[process] E EX_TYPE_E [%2] The task to execute (default task) is %3 type while the rule table is %4 type. This configuration is incorrect.	
<i>Type</i>	Error	
<i>Meaning</i>	Incompatible task type for the Rule Table.	
<i>Parameters</i>	1	Rule Table name
	2	Task type
	3	Rule Table type

Action	Check and change the task type or Rule Table.		
533	[process] I TEMP_EVT The scheduled event planned for %2 occurs; the rule table %3 is going to be scanned		
Type	Information		
Meaning	The scheduled event is started		
Parameters	1	Date/time	
	2	Rule Table name	
534	[process] I FS_EVT The file %2 was put in the watch directory %3, the rule table %4 is going to be scanned		
Type	Information		
Meaning	Scheduled Directory Scanning is started.		
Parameters	1	The deposited file name	
	2	Watch directory name	
	3	Rule Table name	
535	[process] I XMS_EVT The XMS message %2 was recuperated in the queue %3, the rule table %4 is going to be scanned		
Type	Information		
Meaning	Axway Messaging is started.		
Parameters	1	Axway Messaging message	
	2	Axway Messaging queue name	
	3	Rule Table name	
536	[process] I DR_SELECT [%2] the decision rule %3 having the execution type %4 was selected		

<i>Type</i>	Information	
<i>Meaning</i>	The Decision Rule was selected.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Execution type
537	[process] I DR_SELECT [%2] the decision rule %3 was refused because the condition (%4 %5 %6) is invalid	
<i>Type</i>	Information	
<i>Meaning</i>	The Decision Rule was not selected.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Operand
	4	Operator
	5	Operand
538	[process] I MDL_EX [%2] decision rule [%3], the model %4 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	The selected Decision Rule model will be applied.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Model name
539	[process] I D_MDL_EX [%2] the default model %3 will be applied	

Type	Information		
Meaning	The default Model will be applied.		
Parameters	1	Rule Table name	
	2	Default Model name	
540	[process] E MDL_E [%2] decision rule [%3], the model %4 does not exist		
Type	Error		
Meaning	The Model was not found		
Parameters	1	Rule Table name	
	2	Decision Rule	
	3	Model name	
Action	Check that the model name is correct. If model does not exist, create new model.		
541	[process] E D_MDL_E [%2] the default model %3 does not exist		
Type	Error		
Meaning	The default Model was not found.		
Parameters	1	Rule Table name	
	2	Default Model name	
Action	Check that the Model name is correct. If Model does not exist, create new Model.		
542	[process] I SS_EX [%2] decision rule [%3], the batch script %4 will be applied		
Type	Information		

Meaning	The batch script will be applied.	
Parameters	1	Rule Table name
	2	Decision Rule
	3	Batch script
543	[process] I D_SS_EX [%2] the default batch script %3 will be applied	
Type	Information	
Meaning	The default batch script will be applied.	
Parameters	1	Rule Table name
	2	Default batch script
544	[process] E SS_E [%2] decision rule [%3], the batch script is not specified	
Type	Error	
Meaning	The batch script is not referenced.	
Parameters	1	Rule Table name
	2	Decision Rule
Action	Specify the batch script in the Decision Rule action.	
545	[process] E D_SS_E [%2] the default batch script is not specified	
Type	Error	
Meaning	The default script is not referenced	
Parameters	1	Rule Table name
Action	Specify the batch script in the Rule Table default action	

546	[process] I PS_EX [%2] decision rule [%3], the PERL script %4 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	The Perl script will be applied	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Perl script
547	[process] I D_PS_EX [%2] the default Perl script %3 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	The default Perl script will be applied	
<i>Parameters</i>	1	Rule Table name
	2	Default Perl script
548	[process] E PS_E [%2] decision rule [%3], the Perl script is not specified	
<i>Type</i>	Error	
<i>Meaning</i>	The Perl script is not referenced.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
<i>Action</i>	Specify the Perl script in the Decision Rule action.	
549	[process] E D_PS_E [%2] the default Perl script is not specified	
<i>Type</i>	Error	
<i>Meaning</i>	The default Perl script is not referenced.	

<i>Parameters</i>	1	Rule Table name
<i>Action</i>	Specify the Perl script in the Rule Table default action.	
550	[process] E PS_E [%2] decision rule [%3], the specified Perl script can not be found	
<i>Type</i>	Error	
<i>Meaning</i>	The Perl script does not exist	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
<i>Action</i>	Check the name of the Perl script, or create a new script.	
551	[process] E D_PS_E [%2] the default Perl script specified in the rule table cannot be found	
<i>Type</i>	Error	
<i>Meaning</i>	The default Perl script does not exist	
<i>Parameters</i>	1	Rule Table name
<i>Action</i>	Check the name of the Perl script, or create a new script.	
552	[process] I NOP [%2] decision rule [%3], no task will be launched	
<i>Type</i>	Information	
<i>Meaning</i>	The task cannot be applied.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
553	[process] I D_NOP [%2] Default task, no task will be launched	

<i>Type</i>	Information	
<i>Meaning</i>	Default task. None will be applied.	
<i>Parameters</i>	1	Rule Table name

554	[process] I PUR_EX [%2] decision rule [%3], the purge model %4 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	Mailbox cleaning will be applied.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Purge Model name

555	[process] I D_PUR_EX [%2] the default purge model %3 will be applied	
<i>Type</i>	Information	
<i>Meaning</i>	The default Mailbox cleaning will be applied.	
<i>Parameters</i>	1	Rule Table name
	2	Default purge Model

556	[process] E PUR_E [%2] decision rule [%3], the purge model %4 does not exist	
<i>Type</i>	Error	
<i>Meaning</i>	The purge Model was not found.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule
	3	Purge Model name
<i>Action</i>	Check the name of the purge Model or create a new one if it does not exist.	

557	[process] E PUR_E [%2] the default purge model %3 doesn't exist	
Type	Error	
Meaning	The default purge Model was not found.	
Parameters	1	Rule Table name
	2	Purge Model name
Action	Check the name of the purge Model or create a new one if it does not exist.	
558	[process] I LPUR_EX [%2] decision rule [%3], the log will be purged	
Type	Information	
Meaning	Log cleaning will be applied.	
Parameters	1	Rule Table name
	2	Decision Rule
559	[process] I DLPUR_EX [%2] Default task, the log will be purged	
Type	Information	
Meaning	Default: log cleaning will be applied	
Parameters	1	Rule Table name
560	[process] I SPUR_EX [%2] decision rule [%3], the stats will be purged	
Type	Information	
Meaning	The stats cleaning will be applied	
Parameters	1	Rule Table name
	2	Decision Rule

561	[process] I DSPUR_EX [%2] Default task, the stats will be purged	
<i>Type</i>	Information	
<i>Meaning</i>	The default stats cleaning will be applied	
<i>Parameters</i>	1	Rule Table name
562	[process] I PS_E The Perl script %2 could not be executed	
<i>Type</i>	Information	
<i>Meaning</i>	The Perl script was not launched.	
<i>Parameters</i>	1	Perl script
563	[process] I PS_EX The Perl script %2 was executed	
<i>Type</i>	Information	
<i>Meaning</i>	The Perl script was launched.	
<i>Parameters</i>	1	Perl script
564	[process] I XFER_EX [%2] transfer launched according to the model %3	
<i>Type</i>	Information	
<i>Meaning</i>	The transfer was started.	
<i>Parameters</i>	1	Rule Table name
	2	Model name
565	[process] W XFER_E The transfer according to the model %2 could not be launched. The reason for the failure is: %3	

Type	Warning	
Meaning	The transfer was not started.	
Parameters	1	Rule Table name
	2	Model name
	3	Error returned
Action	Check that the Model parameters are set correctly.	
566	[process] I SS_E The batch script %2 could not be executed	
Type	Information	
Meaning	The script was not started	
Parameters	1	Batch script
567	[%1] I SS_EX The batch script %2 was executed	
Type	Information	
Meaning	The script was launched.	
Parameters	1	Process name
	2	Batch script
568	[NORMALLY NOT DISPLAYED]	
Type		
Meaning		
Parameters		
569	%1 I MB_CLEA The mailbox was successfully cleaned according to the purge model %2	

Type	Information	
Meaning	The Mailbox cleaning task is complete.	
Parameters	1	Process name
	2	Purge Model
570	%1 E MB_CLEA_E The mailbox could not be cleaned according to the purge model %2	
Type	Error	
Meaning	The Mailbox could not be cleaned.	
Parameters	1	Process name
	2	Purge Model
Action	None	
571	[process] I LG_CLEA The log was successfully cleared.	
Type	Information	
Meaning	The log cleaning task is complete.	
572	[process] I LG_CLEA_E The log could not be cleared	
Type	Information	
Meaning	Log cleaning was not performed.	
573	[process] I ST_CLEA The stats were successfully cleaned.	
Type	Information	
Meaning	The stats cleaning was performed.	

574	[%1] I ST_CLEA_E The stats could not be cleaned.	
<i>Type</i>	Information	
<i>Meaning</i>	The stats cleaning was not performed	
575	[process] I EXIT_YES [%2] The user exit %3 returned the value 1. The rule table scanning can begin.	
<i>Type</i>	Information	
<i>Meaning</i>	The user exit was successful	
<i>Parameters</i>	1	Rule Table name
	2	User exit name
576	[process] I EXIT_NO [%2] The user exit %3 returned the value 0. The rule table scanning is cancelled.	
<i>Type</i>	Information	
<i>Meaning</i>	The user exit was not successful	
<i>Parameters</i>	1	Rule Table name
	2	User exit name
577	[process] I EXIT_ERR [%2] The user exit %3 returned the value - 1. The rule table scanning is cancelled.	
<i>Type</i>	Information	
<i>Meaning</i>	The exit returned an error.	
<i>Parameters</i>	1	Rule Table name
	2	User exit name

578	[%1] I NO_CONDS [%2] decision rule %3 does not contain conditions. It is therefore automatically selected.	
<i>Type</i>	Information	
<i>Meaning</i>	There are no conditions in the Decision Rule.	
<i>Parameters</i>	1	Rule Table name
	2	Decision Rule

579	[%1] I WORK_DONE [%2] The rule table task is completed.	
<i>Type</i>	Information	
<i>Meaning</i>	Task complete.	
<i>Parameters</i>	1	Rule Table name

Axway Gateway: Messages and codes

SECS messages

This topic lists Security server log messages.

010	SECS E SES_INIT (%1) No Netprof matching [src_addr=%2] [dst_addr=%3]	
<i>Type</i>	Error	
<i>Meaning</i>	Error while trying to retrieve Network Profile object from network parameters (incoming connection).	
<i>Parameters</i>	1	Connection identifier
	2	Source address
	3	Destination address
<i>Action</i>	Check the Network Profile database configuration and make sure that there is one existing record that matches these network parameters.	

011	SECS I SES_INIT (%1) Netprof : %2 selected [src_addr=%3] [dst_addr=%4]	
Type	Information	
Meaning	Information indicating the Network Profile selected for the current connection (incoming connection).	
Parameters	1	Connection identifier
	2	Netprof object name
	3	Source address
	4	Destination address
Action	None	
012	SECS E SES_INIT (%1) %2 Security Profile Error : %3 not found	
Type	Error	
Meaning	Error while trying to retrieve the Security Profile for the current connection. The name of the profile is taken from the Site for outgoing connections or from the Network Profile object found for incoming connections.	
Parameters	1	Connection identifier
	2	Server or client mode
	3	Security profile name
Action	Check if the referenced Security Profile exists in the Security Profile database. If it does not exist, create it or modify the Site or the Network Profile object attributes.	
013	SECS I SES_INIT (%1) %2 Security Profile : %3 selected	
Type	Information	
Meaning	Information indicating the Security Profile name selected for the current connection. The name of the profile is taken from the Site for outgoing connections or from the Network Profile object found for incoming connections.	

Parameters	1	Connection identifier
	2	Server or client mode
	3	Security profile name
Action	None	
014	SECS I CA_BUILD (%1) %2 Sending Accepted DN : %3	
Type	Information	
Meaning	<i>Server mode only</i> Information indicating the local name of the certification trusted authority subject name sent to the client. Typically, several messages are sent consecutively, since more than one authority is accepted.	
Parameters	1	Connection identifier
	2	Server or client mode
	3	Security profile name
Action	None	
015	SECS E CA_BUILD (%1) %2 Accepted DN : %3 not found	
Type	Error	
Meaning	<i>Server mode only</i> Error while trying to retrieve the certificate referenced by its name in the accepted authorities field of the current profile.	
Parameters	1	Connection identifier
	2	Server or client mode
	3	Certificate name
Action	Check that the desired authority is correctly referenced in the current profile, or that the certificate is still present in the local database.	

016	SECS I CA_BUILD (%1) %2 Sending Accepted DN : %3	
<i>Type</i>	Information	
<i>Meaning</i>	<i>Server mode only</i> Information indicating the distinguished name of the certification trusted authority subject name sent to the client. Typically, several messages are sent consecutively, since more than one authority is accepted.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Trusted authority distinguished name
<i>Action</i>	None	
017	SECS E CA_BUILD (%1) %2 Accepted DN : %3 not found	
<i>Type</i>	Error	
<i>Meaning</i>	<i>Server mode only</i> Error while trying to retrieve the certificate referenced by its name in the accepted authorities field of the current profile.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Distinguished name of the trusted authority
<i>Action</i>	Check that the desired authority is correctly referenced in the current profile, or that the certificate is still present in the local database.	
018	SECS I CA_RECV (%1) %2 Dn received : %3 found in base	
<i>Type</i>	Information	

<i>Meaning</i>	<i>Client mode only</i> Information indicating the local name of the certification trusted authority subject name received from the server. This Distinguished Name corresponds to a certificate present in the local database. Typically, several messages are sent consecutively, since more than one authority is received.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Distinguished Name of the trusted authority
<i>Action</i>	None	

019	SECS E CA_RECV (%1) %2 Dn received : %3 not found in base	
<i>Type</i>	Information	
<i>Meaning</i>	<i>Client mode only</i> Information indicating that no certificate corresponding to the trusted authority distinguished name received from the server was found in the local database. Therefore, it is impossible to send a certificate that matches such an authority.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Distinguished name of the trusted authority
<i>Action</i>	None	

020	SECS I CA_RECV (%1) %2 Dn received : %3 found in base	
<i>Type</i>	Information	
<i>Meaning</i>	<i>Client mode only</i> Information indicating the distinguished name of the certification trusted authority subject name received from the server. This distinguished name corresponds to a certificate present in the local database. Typically, several messages are sent consecutively, since more than one authority is received.	

<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Distinguished name of the trusted authority
<i>Action</i>	None	

030	SECS I C_BUILD (%1) %2 Certificate sent: %3	
<i>Type</i>	Information	
<i>Meaning</i>	Information indicating the local name of one of the certification path elements sent to the remote partner. Typically, several messages are sent consecutively, since the certification path sent contains more than one element.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Local name of the sent certificate
<i>Action</i>	None	

031	SECS E C_BUILD (%1) %2 Certificate : %3 not found	
<i>Type</i>	Error	
<i>Meaning</i>	Error while trying to send the corresponding certificate.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Local name of the certificate to send
<i>Action</i>	Check that the certificate is present in the local database, or change the current Security Profile attributes.	

032	SECS I C_BUILD (%1) %2 Certificate sent: %3	

Type	Information	
Meaning	Information indicating the distinguished name of one of the certification path elements sent to the remote partner. Typically, several messages are sent consecutively since the certification path sent contains more than one element.	
Parameters	1	Connection identifier
	2	Server or client mode
	3	Local name of the sent certificate
Action	None	
033	SECS E C_BUILD (%1) %2 Certificate : %3 not found	
Type	Error	
Meaning	Error while trying to send the corresponding certificate.	
Parameters	1	Connection identifier
	2	Server or client mode
	3	Local name of the certificate to send
Action	Check that the certificate is present in the local database, or change the current Security Profile attributes.	
034	SECS E C_BUILD (%1) %2 Certificate Sending Error [reason= %3] :%4	
Type	Error	
Meaning	Error while trying to send the corresponding certificate.	

Parameters	1	Connection identifier
	2	Server or client mode
	3	Reason for the error, one of: • User certificate key error: no private key is associated with user certificate • Certificate not found
	4	Local name of the certificate to send
Action	Check for the reason code and modify the Security Profile settings if required.	

035	SECS E SES_INIT (%1) %3 TLS Security Profile %2 has no XPP entity name, check configuration	
Type	Error	
Meaning	PassPort PS entity name missing from TLS Security Profile. This message can only occur if the PassPort connector is activated in the configuration menu.	
Parameters	1	Connection identifier
	2	TLS Security profile identity
	3	Client or server
Action	Check TLS Security Profile configuration.	

036	SECS E SES_INIT (%1) %3 TLS Security Profile %2 has no XPP entity password, check configuration	
Type	Error	
Meaning	PassPort PS entity password missing from TLS Security Profile.	
Parameters	1	Connection identifier
	2	TLS Security profile identity
	3	Client or server
Action	Check TLS Security Profile configuration.	

040	SECS I C_RECV (%1) %2 Certificate : %3 received		
Type	Information		
Meaning	Information indicating the local name of one of the certification path elements received from the remote partner. Typically, several messages are sent consecutively since the certification path sent contains more than one element.		
Parameters	1	Connection identifier	
	2	Server or client mode	
	3	Local name of the sent certificate	
Action	None		

041	SECS E C_RECV (%1) %2 Certificate check error [reason=%3] : %4		
Type	Error		
Meaning	Information indicating that an error occurred while verifying the certification path received from the remote partner. This indicates that the remote partner authentication has failed.		

Parameters	1	Connection identifier
	2	Server or client mode
	3	Reason code, one of: • <i>Decoding error</i>: unable to decode the certificate content • <i>Template matching</i>: no template matching this certificate was found in the current profile • <i>Validity error</i>: the certificate validity is not correct (too early or too late) • <i>Signature error</i>: the certificate signature is not correct • <i>Root not found</i>: the root authority certificate was not sent within the certification path and was not found in the local database • <i>Invalid Root</i>: the certification path does not lead to a trusted authority • <i>Key usage not granted</i>: the certificate does not authorize the user to carry out this action
	4	Distinguished name of the received certificate
Action	Check the reason code for more information.	

042	SECS I C_RECV (%1) %2 Certification Path Received Correct	
Type	Information	
Meaning	The correct Certification Path has been received.	
Parameters	1	Connection identifier
	2	Server or client mode
Action	None	

043	SECS E SECS (%1) %2 RSA computation error [operation=%3]	
Type	Error	
Meaning	Information indicating that an error occurred during the RSA encryption computation.	

<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Name of failed operation
<i>Action</i>	Contact Axway Support.	

050	SECS E SP_IN_CHK (%1) %2 Security Profile Check Error: %3 not found	
<i>Type</i>	Error	
<i>Meaning</i>	Server mode only Error while checking Site input Security Profile.	
<i>Parameters</i>	1	SUP identifier
	2	Server or client mode
	3	Security profile name
<i>Action</i>	In the Security Profile object, verify the Remote Site/NET security tab settings for the incoming connection	

051	SECS E SP_IN_CHK (%1) %2 Security Profile %3 Check Error: parameter %4 doesn't match [reason=%5]	
<i>Type</i>	Error	
<i>Meaning</i>	Server mode only A security parameter of the established connection does not match the input Site Security Profile.	

<i>Parameters</i>	1	SUP identifier
	2	Server or client mode
	3	Security profile name
	4	Parameter name or value
	5	Error code: • 10: Incompatible type between the Site Security Profile type and the actual mode • 11: Client authentication is mandatory in the Security Profile for the Site, but the client failed to identify itself during the handshake phase • 12: Client authentication is imposed as anonymous in the Security Profile for the Site but the client identified itself during the handshake phase • 13: Server authentication is mandatory in the Security Profile for the Site but the server did not identify itself during the handshake phase • 14: Server authentication imposed as anonymous in the Security Profile for the Site but the server identified itself during the handshake phase • 17: The Security Profile for the Site has an incompatible exit scheduling value with the Security Profile used • 18: The Security Profile for the Site does not allow the cipher suite used • 19: The Security Profile for the Site does not allow the TLS version used • 20: Section "UserLocalCertificates" of the Security Profile for the Site does not contain the sent certificate (%4)
<i>Action</i>	Check the error code for more information.	
052	SECS E SP_IN_CHK (%1) %2 Security Profile %3 Check Error [%4]: DN = %5	
<i>Type</i>	Error	
<i>Meaning</i>	<i>Server mode only</i> The input Security Profile for the Site does not match the received certification chain.	

Parameters	1	SUP identifier
	2	Server or client mode
	3	Security profile name
	4	Reason code: • <i>Decoding error</i>: unable to decode the certificate content • <i>Template matching</i>: no template matching this certificate was found in the current profile • <i>Validity error</i>: the certificate validity is not correct (too early or too late) • <i>Signature error</i>: the certificate signature is not correct • <i>Root not found</i>: the root authority certificate was not sent within the certification path and was not found in the local database • <i>Invalid Root</i>: the certification path does not lead to a trusted authority
	5	Distinguished Name of the certificate
Action	Check the reason code for more information.	

055	SECS I SP_IN_CHK (%1) %2 Security Profile Check Success: %3 matches %4	
Type	Information	
Meaning	<i>Server mode only</i> The Security Profile for the Site matches the Security Profile already used. This message only appears if <code>secs_log_level</code> is higher than 1 and if the input Security Profile for the Site is different from the one selected during the connection phase.	
Parameters	1	SUP identifier
	2	Server or client mode
	3	Security Profile name
	4	Name of the first Security Profile used
Action	None	

060	SECS E THC (%1) %2 Unable to decode certificate	
<i>Type</i>	Error	
<i>Meaning</i>	Not possible to decode a certificate.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
<i>Action</i>	Contact your exchange partner.	
061	SECS E THC (%1) %2 Certificate is not enabled (%3)	
<i>Type</i>	Error	
<i>Meaning</i>	The certificate is not enabled.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Certificate name (alias)
<i>Action</i>	Enable the certificate.	
062	SECS I THC (%1) %2 Certificate is present and enabled (%3)	
<i>Type</i>	Information	
<i>Meaning</i>	The certificate is present and enabled	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Certificate name (alias)
<i>Action</i>	None	
063	SECS E THC (%1) %2 Certificate is not present in the base (%3)	

<i>Type</i>	Error	
<i>Meaning</i>	The certificate is not present in the database	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Certificate subject name
<i>Action</i>	In the Sprof (security profile), activate the option Automatic import of partner certificates . See <i>Security Guide</i> > Managing SSH Security profiles for further information (requires login).	

064	SECS E RCI (%1) %2 Unable to decode certificate	
<i>Type</i>	Information	
<i>Meaning</i>	Gateway cannot decode the certificate.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
<i>Action</i>	Contact your exchange partner.	

065	SECS E RCI (%1) %2 Unable to put certificate in the base (%3)	
<i>Type</i>	Error	
<i>Meaning</i>	Gateway cannot write the certificate to the database.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Certificate name (alias)
<i>Action</i>	Contact Axway Support.	

066	SECS I RCI (%1) %2 Certificate successfully put in the base (%3)	

<i>Type</i>	Information	
<i>Meaning</i>	Gateway has written the certificate to the database. A newly registered certificate has the name format: Imported_Certificate_xxxxxx.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Certificate name (alias)
<i>Action</i>	None	

067 SECS I RCI (%1) %2 Certificate already in the base (%3)

<i>Type</i>	Information	
<i>Meaning</i>	The certificate already exists in the database.	
<i>Parameters</i>	1	Connection identifier
	2	Server or client mode
	3	Certificate name (alias)
<i>Action</i>	None	

090 SECS E SECS (%1) Security Exit Severe Error: [status=%2] [reason=%3]

<i>Type</i>	Information	
<i>Meaning</i>	A severe security exit error has occurred.	
<i>Parameters</i>	1	Connection identifier
	2	Status
	3	Reason
<i>Action</i>	Contact Axway Support.	

099	SECS E SECS unknown message (%1)	
Type	Information	
Meaning	Non identifiable message	
Parameters	1	Connection identifier
Action	Contact Axway Support.	
100	SECS E %1(%2) Security exit severe error [status=%3] [reason=%4]	
Type	Error	
Meaning	A serious error has occurred during execution of a security exit.	
Parameters	1	SUP identifier
	2	Server or client mode
	3	Local transfer identifier
	4	Corresponding local error (if any)
Action	Action depends on the reason code and the exit implementation.	
101	SECS S %1(%2) Session established with site %3, %4	
Type	Success	
Meaning	A session has been established successfully with the named Site.	
Parameters	1	SUP identifier
	2	Server or client mode
	3	Name of the SFTP Site you are connected to
	4	SSH parameters for the session: key exchange algorithm, public key algorithm, cipher algorithm, Mac algorithm.
Action	None	

102	SECS S %1(%2) Session established for user %3, %4	
Type	Success	
Meaning	A session has been established successfully with the named user.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the user connected to Gateway
	4	SSH parameters for the session: key exchange algorithm, public key algorithm, cipher algorithm, Mac algorithm.
Action	None	
103	SECS I %1(%2) Session ended with site %3	
Type	Information	
Meaning	The session with the named Site has been ended.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the SFTP Site for which the session is ended.
Action	None	
104	SECS I %1(%2) Session ended for user %3	
Type	Information	
Meaning	The session with the named user has been ended.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the user for which the session is ended.
Action	None	

105	SECS W %1(%2) No net profile matching [src_add="%3"] [dest_add="%4"]	
Type	Warning	
Meaning	There is no Network profile matching the incoming connection.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Source address of the incoming connection
	4	Destination address of the incoming connection
Action	To accept connections with these source and destination addresses, you must modify the net profile object accordingly.	
106	SECS I %1(%2) Net profile %3 selected [src_add="%4"] [dest_add="%5"]	
Type	Information	
Meaning	A Network profile has been selected for the incoming connection.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the Network profile selected
	4	Source address of the incoming connection
	5	Destination address of the incoming connection
Action	None	
107	SECS W %1(%2) Net profile %3 does not supply any security option	
Type	Warning	
Meaning	No security option is associated with the selected Network profile: the Network profile selected has its security option set to NONE.	

Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the net profile selected
Action	If an SSH connection is required, set the security option to SSH.	
108	SECS E %1(%2) SSH profile %3 not found, check configuration	
Type	Error	
Meaning	The Network profile selected is pointing to a SSH profile which does not exist in the database.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the SSH profile referenced in the net profile
Action	Check your configuration.	
109	SECS E %1(%2) Selected SSH profile %3 is not usable in client mode, check configuration	
Type	Error	
Meaning	The SSH profile selected is of SERVER type and cannot be used to initiate a connection in client mode.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the selected SSH profile
Action	Check your configuration.	
110	SECS E %1(%2) Selected SSH profile %3 is not usable in server mode, check configuration	

<i>Type</i>	Error	
<i>Meaning</i>	The SSH profile selected is of CLIENT type and cannot be used to initiate a connection in server mode.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the SSH profile selected
<i>Action</i>	Check your configuration.	

111	SECS I %1(%2) SSH profile %3 selected	
<i>Type</i>	Information	
<i>Meaning</i>	The named SSH profile has been selected to initiate a new connection.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the SSH profile selected
<i>Action</i>	None	

112	SECS E %1(%2) Signature request with an invalid algorithm identifier (%3)	
<i>Type</i>	Error	
<i>Meaning</i>	The algorithm identifier is displayed as a string if it is recognized (for example ssh-dss), otherwise it is in the form "(id) unknown" (for example "17 (unknown)").	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Requested algorithm identifier
<i>Action</i>	This message is attached to a safety test and should not normally occur. Contact Axway Support.	

113	SECS E %1(%2) Signature verification request with an invalid algorithm identifier (%3)	
<i>Type</i>	Error	
<i>Meaning</i>	The algorithm identifier is displayed as a string if it is recognized (for example ssh-dss), otherwise it is in the form "(id) unknown" (for example "17 (unknown)").	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Requested algorithm identifier
<i>Action</i>	This message is attached to a safety test and should not normally occur. Contact Axway Support.	
114	SECS I %1(%2) DSS signature achieved with private key %3	
<i>Type</i>	Information	
<i>Meaning</i>	A DSS signature is achieved with the named private key.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the DSS private key
<i>Action</i>	None	
115	SECS I %1(%2) RSA signature achieved with private key %3	
<i>Type</i>	Information	
<i>Meaning</i>	A RSA signature is achieved with the named private key.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the RSA private key
<i>Action</i>	None	

116	SECS E %1(%2) DSS signature failure with private key %3	
Type	Error	
Meaning	DSS signature has failed with the named private key	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the DSS private key
Action	Contact Axway Support.	
117	SECS E %1(%2) RSA signature failure with private key %3	
Type	Error	
Meaning	RSA signature has failed with the named private key	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the RSA private key
Action	Contact Axway Support.	
118	SECS I %1(%2) DSS signature verification achieved with public key %3	
Type	Information	
Meaning	The DSS signature verification was successful with the named public key.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the DSS public key
Action	None	

119	SECS I %1(%2) RSA signature verification achieved with public key %3	
Type	Information	
Meaning	The RSA signature verification was successful with the named public key.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the RSA public key
Action	None	
120	SECS E %1(%2) DSS signature verification failure with public key %3	
Type	Error	
Meaning	This error occurs when the partner Mac hash obtained by signature verification is not the same as the Mac hash expected.	
Result	The connection is closed immediately.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the DSS public key
Action	Your partner is probably using a corrupted key or is does not hold the private key corresponding to the public key supplied. Contact your partner and compare the keys.	
121	SECS E %1(%2) RSA signature verification failure with public key %3	
Type	Error	
Meaning	This error occurs when the partner Mac hash obtained by signature verification is not the same as the Mac hash expected.	
Result	The connection is closed immediately.	

<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the RSA public key
<i>Action</i>	Your partner is probably using a corrupted key or is does not hold the private key corresponding to the public key supplied. Contact your partner and compare the keys.	

122	SECS E %1(%2) Public parameters request with an algorithm identifier (%3) that is not set in the SSH profile (algorithm identifier (%3) is not set...)	
<i>Type</i>	Error	
<i>Meaning</i>	Algorithm identifier not set in SSH Profile	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the algorithm identifier
<i>Action</i>	This message is attached to a safety test and should not occur. Contact Axway Support.	

123	SECS E %1(%2) Public parameters supplied with an algorithm identifier (%3) that is not set in the SSH profile	
<i>Type</i>	Error	
<i>Meaning</i>	Algorithm identifier not set in SSH Profile	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the algorithm identifier
<i>Action</i>	This message is attached to a safety test and should not occur. Contact Axway Support.	

124	SECS E %1(%2) No private key alias is defined in the site or the SSH profile	
<i>Type</i>	Error	
<i>Meaning</i>	The name of the private key to use in the SSH connection is not supplied in the SFTP Site or the SSH profile configuration.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
<i>Action</i>	On the client side the private key name may be omitted if your configuration specifies PASSWORD authentication only. Check your configuration.	
125	SECS E %1(%2) No private key alias defined in the SSH profile	
<i>Type</i>	Error	
<i>Meaning</i>	The SSH profile configuration does not supply the name of the private key to be used in SSH connection setup.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
<i>Action</i>	On the server side a private key name is mandatory in the SSH profile definition. Check your configuration.	
126	SECS E %1(%2) Private key %3 not found, check configuration	
<i>Type</i>	Error	
<i>Meaning</i>	The SFTP Site or the SSH profile refers to a private key that does not exist in the database.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of the private key
<i>Action</i>	Check your configuration.	

127	SECS E %1(%2) Selected private key %3 has an algorithm identifier (%4) that is not compatible with the one requested (*moved) (%5), check configuration	
Type	Error	
Meaning	The private key selected has an algorithm identifier that does not match the negotiated algorithm identifier. The negotiated algorithm identifier is specified in the public key algorithm list of the SSH profile used.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the private key
	4	Algorithm identifier of the private key
	5	Algorithm identifier requested by SSH
Action	Check your configuration.	

128	SECS E %1(%2) User %3 login refused (password method), [reason %4]	
Type	Error	
Meaning	A user attempting to login using the password method was rejected by Gateway.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	User name
	4	Reason for login refusal
Action	The user is probably unknown or has supplied an invalid password. If this user should have access permission check your configuration.	

129	SECS E %1(%2) User %3 login refused (public key method), [reason %4]	

<i>Type</i>	Error
<i>Meaning</i>	A user attempting to login using public key method was rejected by Gateway
<i>Parameters</i>	1 SUP identifier
	2 Client or server mode
	3 User name
	4 Reason for login refusal
<i>Action</i>	The user is probably unknown. If this user should have access permission check your configuration.

130 SECS E %1(%2) Authentication control failed for site %3

<i>Type</i>	Error
<i>Meaning</i>	A SSH connection attempt with a partner server was unsuccessful. This is a general error message, which will be preceded in the log by a more specialized one.
<i>Parameters</i>	1 SUP identifier
	2 Client or server mode
	3 Site name
<i>Action</i>	Check your configuration and contact your partner.

131 SECS E %1(%2) Authentication control failed for user %3

<i>Type</i>	Error
<i>Meaning</i>	A SSH connection attempt from a user partner was unsuccessful. This is general error message, which will be preceded in the log by a more specialized one.
<i>Parameters</i>	1 SUP identifier
	2 Client or server mode
	3 User name
<i>Action</i>	Check your configuration and contact your partner.

132	SECS E %1(%2) Authentication control failed for user %3		
Type	Error		
Meaning	A SSH connection attempt from a user partner was unsuccessful. This is general error message that may be related to implementation.		
Parameters	1	SUP identifier	
	2	Client or server mode	
	3	User name	
Action	Check your configuration and contact Axway Support if necessary.		
133	SECS E %1(%2) Authentication control failed for user %3		
Type	Error		
Meaning	A SSH connection attempt from a user partner was unsuccessful.		
Parameters	1	SUP identifier	
	2	Client or server mode	
	3	User name	
Action	This is general error message that may be related to implementation. Check your configuration and contact Axway Support if necessary.		
134	SECS W %1(%2) Key %3 (group %4) imported from site %5, waiting for validation		
Type	Warning		
Meaning	A key has been imported from a server Site and has the state TO_CHECK.		

Parameters	1	SUP identifier
	2	Client or server mode
	3	Key name generated by Gateway
	4	Key group name
	5	Site that submitted the key
Action	Check the key fingerprint with your partner. You may then change its name, enable or disable it, change its group or comments, etc.	
135	SECS W %1(%2) Key %3 (group %4) imported from user %5, waiting for validation	
Type	Warning	
Meaning	A key has been imported from a client Site and has the state TO_CHECK.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Key name generated by Gateway
	4	Key group name
	5	User that submitted the key
Action	Check the key fingerprint with your partner. You may then change its name, enable or disable it, change its group or comments, and so on.	
136	SECS W %1(%2) Key import cancelled, key received from site %3, already exists as %4 (group %5) in state %6	
Type	Warning	
Meaning	A key has been submitted from a server Site and Gateway aborted the import operation because the key already exists in the product database.	

Parameters	1	SUP identifier
	2	Client or server mode
	3	Site that submitted the key
	4	Existing key name
	5	Existing key group
	6	Existing key state
Action	The key may have not been validated yet or disabled for some reason. Check the existing key.	
137	SECS W %1(%2) Key import cancelled, key received from site %3, already exist as %4 (group %5) in state %6	
Type	Warning	
Meaning	A key has been submitted from a client Site and Gateway aborted the import operation because the key already exists in the product database.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	User that submitted the key
	4	Existing key name
	5	Existing key group
	6	Existing key state
Action	The key may have not been validated yet or disabled for some reason. Check the existing key or contact your partner.	
138	SECS E %1(%2) Key access error	
Type	Error	
Meaning		

Parameters	1	SUP identifier
	2	Client or server mode
Action	This is an internal error. Contact Axway Support.	

139	SECS W %1(%2) Public keys (%3 and %4) with the same fingerprint were found, these keys may be duplicated	
Type	Warning	
Meaning	Your database contains two public keys with the same fingerprint. This means the same key has been imported twice with two different names.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	First public key
	4	Second public key
Action	Delete one of the keys. This message can appear if the trustHostedPublicKeys option is used. (See the <i>Security guide</i> > SSH authentication for further information).	

140	SECS E %1(%2) Specified public key (%3) and received key fingerprint do not match	
Type	Error	
Meaning	The key received from the partner does not match the public key specified for authentication control in the Site or the SSH profile.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Public key name specified for authentication control
Action	The connection has been aborted. Check the key fingerprint with your partner.	

141	SECS E %1(%2) Specified public key (%3) is not enabled	
Type	Error	
Meaning	The public key specified for authentication control in the Site or the SSH profile is not in the ENABLED state.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Public key name specified for authentication control
Action	The key may have been intentionally disabled temporarily. Check your configuration.	
142	SECS E %1(%2) Specified public key (%3) not found, check configuration	
Type	Error	
Meaning	The public key specified for authentication control in the Site or the SSH profile cannot be found.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Public key name specified for authentication control
Action	Check your configuration.	
143	SECS E %1(%2) Public key name (%3) specified in the site invalidates the key name in the SSH profile (%4), check configuration	
Type	Error	
Meaning	There is a conflict between the public key name specified in the Site and the public key name specified in the SSH profile.	

<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Public key name in the Site
	4	Public key name in the SSH profile
<i>Action</i>	Check your configuration.	

144	SECS E %1(%2) Received public key matches a hosted key (%3) in an other group (%4)	
<i>Type</i>	Error	
<i>Meaning</i>	A hosted key matches the key received from the partner but the authentication control has a public key group different from the received key group. This message can only appear if the trustHostedPublicKeys option is used. (See the <i>Security guide</i> > SSH authentication for further information).	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Matching hosted public key
	4	Group of hosted public key
<i>Action</i>	Check your configuration. Compare the key group in the message and the "public_key_group" parameter in the authentication control structure.	

145	SECS E %1(%2) Received public key matches a hosted key (%3) that is not enabled	
<i>Type</i>	Error	
<i>Meaning</i>	A hosted key matches the key received from the partner but is not ENABLED. This message can only appear if the trustHostedPublicKeys option is used. The connection is aborted. (See the <i>Security guide</i> > SSH authentication for further information).	

Parameters	1	SUP identifier
	2	Client or server mode
	3	Matching hosted public key
Action	Check the key with your partner, and modify its state if necessary.	
146	SECS E %1(%2) Public key received from site %3 does not match a hosted key	
Type	Error	
Meaning	The server Site has submitted a key that does not match any public key in the database. The connection has been aborted. The key may have been imported according to Gateway configuration.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Site name
Action	This message can appear if the trustHostedPublicKeys option is used. (See the <i>Security guide</i> > SSH authentication for further information).	
147	SECS E %1(%2) Public key received from user %3 rejected	
Type	Error	
Meaning	The client has submitted a key that does not match any public key in the database. The connection has been aborted. The key may have been imported according to the configuration.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	User name
Action	This message can appear if the trustHostedPublicKeys option is used. (See the <i>Security guide</i> > SSH authentication for further information).	

148	SECS E %1(%2) Public key received from site %3 rejected	
<i>Type</i>	Error	
<i>Meaning</i>	The server Site submitted a key that does not match any public key in the database. The connection has been aborted. The key may have been imported according to the configuration.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Site name
<i>Action</i>	This message can appear if the trustHostedPublicKeys option is not used. (See the <i>Security guide</i> > SSH authentication for further information).	
149	SECS E %1(%2) Public key received from user %3 does not match a hosted key	
<i>Type</i>	Error	
<i>Meaning</i>	The client Site submitted a key that does not match any public key in the database. The connection has been aborted. The key may have been imported according to the configuration.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	User name
<i>Action</i>	This message can appear if the trustHostedPublicKeys option is not used. (See the <i>Security guide</i> > SSH authentication for further information).	
150	SECS E %1(%2) RSA signature verification gets a different hash length	
<i>Type</i>	Error	
<i>Meaning</i>	This error occurs when the partner Mac hash obtained by the signature verification does not have the same length as the Mac hash expected.	

Parameters	1	SUP identifier
	2	Client or server mode
Action	Contact your partner and compare the keys.	
151	SECS E %1(%2) RSA signature verification gets a different hash value	
Type	Error	
Meaning	This error occurs when the partner Mac hash obtained by signature verification is not the same as the Mac hash expected. The obvious reason for this is that your partner is using a corrupted key or does not own the private key corresponding to the public key supplied.	
Parameters	1	SUP identifier
	2	Client or server mode
Action	The connection is closed immediately. Contact your partner and compare the keys.	
152	SECS E KEY_ERR %1(%2) DH public key generation failure, session aborted	
Type	Error	
Meaning	Internal error occurred.	
Parameters	1	SUP identifier
	2	Client or server mode
Action	Contact Axway Support.	
153	SECS E CRT_ERR %1(%2) No certificate alias defined in the site or the SSH profile	
Type	Error	
Meaning	No certificate alias has been defined in the Site or in the SSH profile.	

Parameters	1	SUP identifier
	2	Client or server mode
Action	Define a certificate alias.	
154	SECS E CRT_ERR %1(%2) No certificate alias defined in the SSH profile	
Type	Error	
Meaning	No certificate alias has been defined in the SSH profile.	
Parameters	1	SUP identifier
	2	Client or server mode
Action	Define a certificate alias.	
155	SECS E CRT_ERR %1(%2) Certificate %3 not found, check configuration	
Type	Error	
Meaning	No certificate alias has been defined in the SSH profile.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of certificate
Action	Check configuration for certificate name and location.	
156	SECS E CRT_ERR %1(%2) Selected certificate %3 has an algorithm identifier (%4) that is not compatible with the requested one (%5), check configuration	
Type	Error	
Meaning	The selected certificate has an algorithm identifier that is not compatible with the requested one.	

<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of certificate
	4	Certificate algorithm identifier
	5	Required algorithm identifier
<i>Action</i>	Check configuration	

157	SECS E CRT_ERR %1(%2) Specified certificate (%3) and received certificate do not match	
<i>Type</i>	Error	
<i>Meaning</i>	The selected certificate has an algorithm identifier that is not compatible with the requested algorithm.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Specified certificate
<i>Action</i>	Contact your exchange partner.	

158	SECS E CRT_ERR %1(%2) Specified certificate (%3) is not enabled	
<i>Type</i>	Error	
<i>Meaning</i>	The specified certificate (%3) is not enabled	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Specified certificate
<i>Action</i>	Enable the certificate or specify an enabled certificate	

159	SECS E CRT_ERR %1(%2) Specified certificate (%3) not found, check configuration	
Type	Error	
Meaning	The specified certificate was not found	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Specified certificate
Action	Check configuration for available certificates	
160	SECS E CRT_ERR %1(%2) Received certificate matches a hosted certificate (%3) that is not enabled	
Type	Error	
Meaning	The received certificate matches an unenabled hosted certificate	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Unenabled hosted certificate
Action	Enable the certificate.	
161	SECS E CRT_ERR %1(%2) Certificate received from site %3 does not match a hosted certificate	
Type	Error	
Meaning	The certificate received from the Site does not match a hosted certificate	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Partner Site
Action	Check your security certificate configuration.	

162	SECS E CRT_ERR %1(%2) Certificate received from user %3 does not match a hosted certificate	
Type	Error	
Meaning	The certificate received from user does not match a hosted certificate	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Certificate received from user
Action	Check your security certificate configuration.	
163	SECS E CRT_ERR %1(%2) Certificate received from site %3 rejected	
Type	Error	
Meaning	The certificate received from the site was rejected.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Certificate received from Site
Action	Check your security certificate configuration.	
164	SECS E CRT_ERR %1(%2) Certificate received from user %3 rejected	
Type	Error	
Meaning	Certificate received from the user was rejected.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Certificate received from user
Action	Check your security certificate configuration	

165	SECS E CRT_IMP %1(%2) Certificate access error	
Type	Error	
Meaning	Certificate access error	
Parameters	1	SUP identifier
	2	Client or server mode
Action	Check your security certificate configuration.	
166	SECS W CRT_IMP %1(%2) Certificates (%3 and %4) with the same subject name were found, these certificates may be duplicated	
Type	Warning	
Meaning	Certificate access error.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Certificate sharing subject name (duplicate)
	4	Certificate sharing subject name (duplicate)
Action	Check your security certificate configuration.	
167	SECS I SIGN_OK %1(%2) RSA signature verification achieved with certificate %3	
Type	Information	
Meaning	Confirmation of signature verification with specified certificate.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Certificate used for signature verification
Action	None	

168	SECS E SIGN_ERR %1(%2) RSA signature verification failure with certificate %3	
Type	Error	
Meaning	Verification failure with specified certificate.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Certificate used for signature verification
Action	Contact your exchange partner.	
169	SECS W CRT_IMP %1(%2) Certificate import cancelled, certificate received from site %3, already exist as %4 in state %5	
Type	Warning	
Meaning	Import canceled because certificate already exists.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of Site from which certificate was being imported
	4	Name of existing certificate
	5	State of existing certificate
Action	None	
170	SECS W CRT_IMP %1(%2) Certificate import cancelled, certificate received from user %3, already exists as %4 in state %5	
Type	Warning	
Meaning	Import canceled because certificate already exists.	

<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of user from which certificate was being imported
	4	Name of existing certificate
	5	State of existing certificate
<i>Action</i>	None	

171	SECS W CRT_IMP %1(%2) Certificate %3 imported from site %4, waiting for validation	
<i>Type</i>	Warning	
<i>Meaning</i>	Imported certificate is waiting for validation.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of certificate imported
	4	Name of Site from which certificate was imported
<i>Action</i>	None	

172	SECS W CRT_IMP %1(%2) Certificate %3 imported from user %4, waiting for validation	
<i>Type</i>	Warning	
<i>Meaning</i>	Imported certificate is waiting for validation.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of certificate imported
	4	Name of user from which certificate was imported
<i>Action</i>	None	

173	SECS I SIGN_OK %1(%2) RSA signature achieved with certificate %3		
Type	Information		
Meaning	Verification of success of signature process.		
Parameters	1	SUP identifier	
	2	Client or server mode	
	3	Name of certificate used for signature	
Action	None		
174	SECS E SIGN_ERR %1(%2) RSA signature failure with certificate %3		
Type	Error		
Meaning	Failure of signature process with specified certificate.		
Parameters	1	SUP identifier	
	2	Client or server mode	
	3	Name of certificate used in signature failure	
Action	Contact Axway Support.		
175	SECS E CRT_ERR %1(%2) Certificate received from site %3 can not be parsed		
Type	Error		
Meaning	Parsing failure of certificate from specified Site.		
Parameters	1	SUP identifier	
	2	Client or server mode	
	3	Name of originating Site of certificate that cannot be parsed	
Action	Contact your exchange partner.		

176	SECS E CRT_ERR %1(%2) Certificate received from user %3 can not be parsed	
<i>Type</i>	Error	
<i>Meaning</i>	Parsing failure of certificate from specified user.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of originating user of certificate that cannot be parsed
<i>Action</i>	Contact your exchange partner.	
177	SECS E CRT_ERR %1(%2) Certificate received from site %3 is expired	
<i>Type</i>	Error	
<i>Meaning</i>	Expiration of certificate from the specified Site.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of Site from which the expired certificate was received
<i>Action</i>	Contact your exchange partner.	
178	SECS E CRT_ERR %1(%2) Certificate received from user %3 is expired	
<i>Type</i>	Error	
<i>Meaning</i>	Expiration of certificate from the specified user.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of user from which the expired certificate was received
<i>Action</i>	Contact your exchange partner.	

179	SECS E CRT_ERR %1(%2) DER parsing error for certificate %3	
Type	Error	
Meaning	DER parsing error of the specified certificate	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the certificate object of DER parsing error
Action	Contact your exchange partner.	
180	SECS E CRT_ERR %1(%2) DER decoding error for certificate %3	
Type	Error	
Meaning	DER coding error of the specified certificate	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the certificate object of DER coding error
Action	Contact your exchange partner.	
181	SECS E CRT_ERR %1(%2) Record to DER conversion error for certificate %3	
Type	Error	
Meaning	Record to DER conversion error for the specified certificate	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the certificate object of the record to DER conversion error
Action	Contact Axway Support.	

182	SECS E CRT_ERR %1(%2) Certificate received from site %3 is self signed	
Type	Error	
Meaning	Certificate received from specified Site is self signed.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the originating Site of the self-signed certificate
Action	The certificate is non-trusted. Contact your exchange partner.	
183	SECS E CRT_ERR %1(%2) Certificate received from user %3 is self signed	
Type	Error	
Meaning	Certificate received from specified user is self signed.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name (alias) of the originating user of the self-signed certificate
Action	The certificate is non-trusted. Contact your exchange partner.	
184	SECS E CRT_ERR %1(%2) Could not find an issuer certificate with name %3	
Type	Error	
Meaning	Issuer certificate with specified name was not found.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Subject name of the unfound issuer certificate
Action	Contact your exchange partner.	

185	SECS E CRT_ERR %1(%2) Issuer certificate %3 is not enabled	
Type	Error	
Meaning	Issuer certificate with specified name is not enabled.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the un-enabled issuer certificate
Action	Enable the certificate.	
186	SECS E CRT_ERR %1(%2) Issuer certificate %3 is expired	
Type	Error	
Meaning	Issuer certificate with specified name has expired.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name (alias) of the expired issuer certificate
Action	Acquire new valid certificate.	
187	SECS E CRT_ERR %1(%2) Intermediate certificate %3 signature invalid for issuer %4	
Type	Error	
Meaning	Signature of intermediate certificate invalid for issuer.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Intermediate certificate name (alias)
	4	Issuer certificate name (alias)
Action	Contact your exchange partner.	

188	SECS E CRT_ERR %1(%2) Certificate received from site %3 has an invalid signature for issuer %4	
Type	Error	
Meaning	Signature of certificate from specified Site has invalid signature for specified issuer.	
Parameters	1	SUP identifier
	2	Client mode
	3	Site that submitted certificate
	4	Issuer certificate name (alias)
Action	Contact your exchange partner.	
189	SECS E CRT_ERR %1(%2) Certificate received from user %3 has an invalid signature for issuer %4	
Type	Error	
Meaning	Signature of certificate from specified Site has invalid signature.	
Parameters	1	SUP identifier
	2	Server mode
	3	User of certificate with invalid signature
	4	Issuer certificate name (alias)
Action	Contact your exchange partner.	
190	SECS E CRT_ERR %1(%2) Root certificate %3 self signature is invalid	
Type	Error	
Meaning	Self signature of root certificate is not valid.	

Parameters	1	SUP identifier
	2	Client or server mode
	3	Name (alias) of root certificate with invalid signature
Action	Verify the root certificate.	
191	SECS E CRT_ERR %1(%2) Certificate %3 has an unknown type (%4)	
Type	Error	
Meaning	The last certificate of the validation chain is not a root-type certificate.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of certificate with unknown type
	4	Unknown certificate type
Action	Verify the certificates in the base.	
192	SECS E CRT_ERR %1(%2) Certificate received from site %3 is hosted as %4 with same subject name but differs	
Type	Error	
Meaning	The certificate subject name field is correct but other certificate fields do not contain expected information.	
Parameters	1	SUP identifier
	2	Client mode
	3	Originating Site of certificate
	4	Name of certificate with same subject name
Action	Contact your exchange partner.	

193	SECS E CRT_ERR %1(%2) Certificate received from user %3 is hosted as %4 with same subject name but differs		
Type	Error		
Meaning	The certificate subject name field is correct but other certificate fields do not contain expected information		
Parameters	1	SUP identifier	
	2	Server mode	
	3	Originating user of certificate	
	4	Name of certificate with same subject name	
Action	Contact your exchange partner.		
194	SECS E CRT_ERR %1(%2) Specified certificate (%3) has an invalid type (%4), USER type is expected		
Type	Error		
Meaning	Received certificate of another type when expecting USER.		
Parameters	1	SUP identifier	
	2	Client or server mode	
	3	Name of certificate with unexpected type	
	4	Unexpected type	
Action	Verify the certificates in the base.		
195	SECS E CRT_ERR %1(%2) Specified certificate (%3) has an invalid type (%4), OTHER type is expected		
Type	Error		
Meaning	Received certificate of another type when expecting OTHER type.		

<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Name of certificate with unexpected type
	4	Unexpected type
<i>Action</i>	Contact your exchange partner.	

196	SECS E CRT_ERR %1(%2) Certificate received from user %3 does not match the subject name pattern in ssh profile	
<i>Type</i>	Error	
<i>Meaning</i>	Received certificate from user with mismatched SSH profile subject name.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Originating user of certificate with mismatched subject name
<i>Action</i>	If this is an error, check your SSH Security Profile.	

197	SECS E CRT_ERR %1(%2) Certificate received from user %3 does not match the issuer name pattern in ssh profile	
<i>Type</i>	Error	
<i>Meaning</i>	Received certificate from user with mismatched SSH profile issuer name.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Originating user of certificate with mismatched issuer name
<i>Action</i>	If this is an error, check your SSH Security Profile.	

198	SECS E CRT_ERR %1(%2) Certificate received from site %3 does not match the subject name pattern in ssh profile	

<i>Type</i>	Error
<i>Meaning</i>	Received certificate from user with mismatched SSH profile subject name pattern.
<i>Parameters</i>	1 SUP identifier
	2 Client mode
	3 Originating user of certificate with mismatched subject name pattern
<i>Action</i>	If this is an error, check your SSH Security Profile.

199 SECS E CRT_ERR %1(%2) Certificate received from site %3 does not match the issuer name pattern in ssh profile

<i>Type</i>	Error
<i>Meaning</i>	Received certificate from user with mismatched SSH profile issuer name pattern.
<i>Parameters</i>	1 SUP identifier
	2 Server mode
	3 Originating user of certificate with mismatched issuer name pattern
<i>Action</i>	If this is an error, check your SSH Security Profile.

200 Certificate received from site %3 is signed by an issuer (%4) different from the expected one

<i>Type</i>	Error
<i>Meaning</i>	Received certificate from specified Site is signed by unexpected issuer.
<i>Parameters</i>	1 SUP identifier
	2 Client or server mode
	3 Originating Site of certificate with unexpected issuer signature
	4 Unexpected issuer
<i>Action</i>	Check your SSH Security Profile.

201	SECS E CRT_ERR %1(%2) Certificate received from user %3 is signed by an issuer (%4) different from the expected one	
<i>Type</i>	Error	
<i>Meaning</i>	Received certificate from specified User is signed by unexpected issuer.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Originating user of certificate with unexpected issuer signature
	4	Unexpected issuer
<i>Action</i>	Check your SSH Security Profile.	
202	SECS I SIGN_OK %1(%2) RSA signature verification with a certificate issued by %3	
<i>Type</i>	Information	
<i>Meaning</i>	RSA signature verification with a certificate from specified issuer.	
<i>Parameters</i>	1	SUP identifier
	2	Client or server mode
	3	Issuer of certificate with RSA signature
<i>Action</i>	None	
203	SECS E AUTHERR %1(%2) Invalid new password supplied by user %3, password unchanged	
<i>Type</i>	Error	
<i>Meaning</i>	Specified user submitted invalid new password.	

Parameters	1	SUP identifier
	2	Client or server mode
	3	User who supplied invalid new password
Action	Ask your exchange partner to retry transfer.	
204	SECS I AUTH_OK %1(%2) Password changed successfully by user %3	
Type	Information	
Meaning	Change of password by specified user.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	User who changed password
Action	None	
205	SECS E SES_INIT %1(%2) Selected SSH profile %3 has no XPP entity name, check configuration	
Type	Error	
Meaning	Missing PassPort PS entity name in SSH profile.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Selected SSH profile
Action	Check SSH profile object configuration.	
206	SECS E SES_INIT %1(%2) Selected SSH profile %3 has no XPP entity password, check configuration	
Type	Error	

Meaning	Missing PassPort PS password in SSH profile.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Selected SSH profile
Action	Check SSH profile object configuration.	
207	SECS E CRT_ERR %1(%2) Certificate received from site %3 has insufficient key usage	
Type	Error	
Meaning	Insufficient key usage on received Site certificate.	
Parameters	1	SUP identifier
	2	Client mode
	3	Site originating message
Action	Contact your exchange partner	
208	SECS E CRT_ERR %1(%2) Certificate received from user %3 has insufficient key usage	
Type	Error	
Meaning	Insufficient key usage on received user certificate.	
Parameters	1	SUP identifier
	2	Server mode
	3	Certificate alias
Action	Contact your exchange partner	
209	SECS E CRT_ERR %1(%2) Certificate received from site %3 has insufficient extended key usage	

<i>Type</i>	Error
<i>Meaning</i>	Insufficient extended key usage on received Site certificate.
<i>Parameters</i>	1 SUP identifier
	2 Client mode
	3 Site that originated the certificate
<i>Action</i>	Contact your exchange partner

210 SECS E CRT_ERR %1(%2) Certificate received from user %3 has insufficient extended key usage

<i>Type</i>	Error
<i>Meaning</i>	Insufficient extended key usage on received user certificate.
<i>Parameters</i>	1 SUP identifier
	2 Server mode
	3 User who originated the certificate
<i>Action</i>	Contact your exchange partner.

211 SECS E LOGNERR %1(%2) User %3 login refused (password method, public key %5 expected), [reason %4]

<i>Type</i>	Error
<i>Meaning</i>	User login or password refused.
<i>Parameters</i>	1 Security Server ID
	2 Client or server mode
	3 User whose login is refused
	4 Reason for login refusal
	5 Expected public key
<i>Action</i>	Verify the CGate configuration.

212	SECS E LOGNERR %1(%2) User %3 login refused (password method, certificate %5 expected), [reason %4]	
Type	Error	
Meaning	User login refused.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	User whose login is refused
	4	Reason for login refusal
	5	Expected certificate
Action	Verify that the CGate has a certificate listed in the Security tab.	
213	SECS E SES_INIT %1(%2) Specified SSH profile (%3) is not enabled	
Type	Error	
Meaning	The SSH profile specified in the net profile or in the Site is not in the ENABLED state.	
Parameters	1	SUP identifier
	2	Client or server mode
	3	Name of the SSH profile referenced in the net profile or in the Site
Action	The SSH profile may have been intentionally disabled temporarily. Check your configuration.	
300	SECS E UEX %1	
Type	Error	
Meaning	Error occurred on SECS PassPort PS exit.	
Parameters	1	PassPort PS generated message

Action	Contact Axway Support.	
301	SECS W UEX %1	
Type	Warning	
Meaning	Error occurred on SECS PassPort PS exit.	
Parameters	1	PassPort PS generated message
Action	Contact Axway Support	
302	SECS I UEX %1	
Type	Information	
Meaning	Error occurred on SECS PassPort PS exit	
Parameters	1	PassPort PS generated message
Action	None	
303	SECS E UEX %1	
Type	Error	
Meaning	SSH error occurred on SECS PassPort PS exit	
Parameters	1	PassPort PS generated message
Action	Contact Axway Support.	
304	SECS W UEX %1	
Type	Warning	
Meaning	SSH error occurred on SECS PassPort PS exit	
Parameters	1	PassPort PS generated message
Action	Contact Axway Support.	

305	SECS I UEX %1
<i>Type</i>	Information
<i>Meaning</i>	SSH error occurred on SECS PassPort PS exit
<i>Parameters</i>	1 PassPort PS generated message
<i>Action</i>	Contact Axway Support.
310	SECS E SECT %1
<i>Type</i>	Error
<i>Meaning</i>	Generic ASx-related error message
<i>Parameters</i>	1 Message text generated by the Security Server
<i>Action</i>	None
311	SECS W SECT %1
<i>Type</i>	Warning
<i>Meaning</i>	Generic ASx-related warning message
<i>Parameters</i>	1 Message text generated by the Security Server
<i>Action</i>	None
312	SECS I SECT %1
<i>Type</i>	Information
<i>Meaning</i>	Generic ASx-related information message
<i>Parameters</i>	1 Message text generated by the Security Server
<i>Action</i>	None

313	SECS E SECT Certificate alias expected	
<i>Type</i>	Error	
<i>Meaning</i>	Trading partner does not have certificate alias	
<i>Action</i>	Verify the configuration of the Trading Partner.	
314	SECS E SECT Certificate list build failed	
<i>Type</i>	Error	
<i>Meaning</i>	Certificate list build failed	
<i>Action</i>	Contact Axway Support.	
315	SECS E SECT User certificate %1 not found	
<i>Type</i>	Error	
<i>Meaning</i>	Received certificate alias does not exist in database.	
<i>Parameters</i>	1	Name of user certificate
<i>Action</i>	Verify configuration of Trading Partner.	
316	SECS E SECT Authority certificate %1 not found	
<i>Type</i>	Error	
<i>Meaning</i>	The specified certificate does not exist in the certificate list	
<i>Parameters</i>	1	Name of authority certificate
<i>Action</i>	None	
317	SECS E SECT Authority certificate error: %2 for subject name %1	
<i>Type</i>	Error	

Meaning	An authority certificate with this name could not be found.	
Parameters	1	Subject name
	2	Error message
Action	Check the certificate database or contact Axway Support.	
318	SECS E SECT Certificate %1 is not enabled	
Type	Error	
Meaning	SECT certificate not enabled.	
Parameters	1	Certificate name
Action	Enable the certificate.	
319	SECS E SECT Certificate list length (%1) exceed maximum length (%3)	
Type	Error	
Meaning	The length of the certificate list exceeds the maximum limit (max = 16 000)	
Parameters	1	Certificate list length
Action	Contact Axway Support.	
320	SECS E SECT User certificate %1 does not have a private key	
Type	Error	
Meaning	The specified user certificate does not have a private key.	
Parameters	1	User certificate
Action	Verify the certificate.	
321	SECS I SECT Certificate %1 sent	

<i>Type</i>	Information
<i>Meaning</i>	The specified certificate has been sent.
<i>Parameters</i>	1 Certificate name
<i>Action</i>	None

322 SECS I SECT Certificate with subject name %1 sent

<i>Type</i>	Information
<i>Meaning</i>	The specified certificate has been sent.
<i>Parameters</i>	1 Subject name of sent certificate
<i>Action</i>	None

323 SECS I SECT Partner certificate with subject name %1 received

<i>Type</i>	Information
<i>Meaning</i>	The partner certificate with the specified subject name has been received.
<i>Parameters</i>	1 Subject name of received partner certificate
<i>Action</i>	None

324 SECS E SECT Partner certificate list check failed

<i>Type</i>	Error
<i>Meaning</i>	Checking of the partner certificate list failed.
<i>Parameters</i>	None
<i>Action</i>	Contact your exchange partner.

325 SECS E SECT Partner certificate list length is too small (%1)

<i>Type</i>	Error
<i>Meaning</i>	The length of the partner certificate list is too short.
<i>Parameters</i>	1 Length of the partner certificate list
<i>Action</i>	Contact your exchange partner.
326	SECS E SECT Partner certificate operation error (%1)
<i>Type</i>	Error
<i>Meaning</i>	Error in partner certificate operation.
<i>Parameters</i>	1 Operation where error took place
<i>Action</i>	Contact your exchange partner.
327	SECS E SECT Partner certificate decoding error (subject name %1)
<i>Type</i>	Error
<i>Meaning</i>	Error in partner certificate decoding.
<i>Parameters</i>	1 Certificate subject name
<i>Action</i>	Contact your exchange partner.
328	SECS E SECT Partner certificate expired (subject name %1)
<i>Type</i>	Error
<i>Meaning</i>	Partner certificate expired.
<i>Parameters</i>	1 Certificate subject name
<i>Action</i>	Contact your exchange partner.
329	SECS E SECT Partner certificate signature invalid (subject name %1 signed by issuer name %2)

<i>Type</i>	Error
<i>Meaning</i>	Invalid partner certificate signature.
<i>Parameters</i>	1 Certificate subject name
	2 User name
<i>Action</i>	Contact your exchange partner.

330	SECS E SECT Partner self signed certificate signature invalid (subject name %1)
<i>Type</i>	Error
<i>Meaning</i>	Invalid partner self signed certificate signature.
<i>Parameters</i>	1 Certificate subject name
<i>Action</i>	Contact your exchange partner.

331	SECS E SECT Partner certificate key decoding error
<i>Type</i>	Error
<i>Meaning</i>	Error in partner certificate key decoding.
<i>Parameters</i>	None
<i>Action</i>	Contact your exchange partner.

332	SECS E SECT Root certificate not found in product store (subject name %1)
<i>Type</i>	Error
<i>Meaning</i>	Root certificate not found in product store.
<i>Parameters</i>	1 Certificate subject name
<i>Action</i>	Import the certificate to the product store.

333	SECS E SECT RSA computation error (%1)	
Type	Error	
Meaning	RSA computation error.	
Parameters	1	Type of operation
Action	None	
334	SECS E SECT RSA computation aborted (no certificate alias supplied)	
Type	Error	
Meaning	Trading Partner object encryption certificate field is empty.	
Parameters	None	
Action	Verify Trading Partner object configuration.	
335	SECS E SECT RSA computation aborted, no key for certificate alias %1	
Type	Error	
Meaning	Computation log was expected.	
Parameters	1	Certificate alias
Action	Contact Axway Support.	
400	SECS E %1 (%2) Received public key provides insufficient security strength in FIPS mode.	
Type	Error	
Meaning	FIPS validation error while attempting to import a public key received from the partner (the received key is not FIPS compliant).	
Parameters	1	Client or server mode

	2	SUP identifier
<i>Action</i>	Check the partner agreement regarding the key algorithms and lengths. Update the agreement in order to enforce the usage of FIPS compliant keys/certificates.	
401	SECS E %1 (%2) Selected key [name=%3/algo=%4] provides insufficient security in FIPS mode.	
<i>Type</i>	Error	
<i>Meaning</i>	FIPS validation error while attempting to use a private key defined in Gateway objects (the key is not FIPS compliant).	
<i>Parameters</i>	1	Client or server mode
	2	SUP identifier
	3	Name of the key
	4	Algorithm of the key
<i>Action</i>	Check the usage of the FIPS restricted key in the different configuration objects and update the configuration to remove the references to this key.	

Axway Gateway: Messages and codes

Simple Mail Transfer Protocol (SMTP)

This topic lists SMTP log messages.

101	SMTP I TRACE (%1) [%2] %3	
<i>Type</i>	Information	
<i>Meaning</i>	This message is used to log unexpected events	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Message
<i>Action</i>	None	

102	SMTP I TRACE (%1) [%2] %3: %4	
<i>Type</i>	Information	
<i>Meaning</i>	This message is used to trace received and sent data	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Direction: R for received, S for sent
	4	Data
<i>Action</i>	None	
103	SMTP E ERROR (%1) [%2] Received an Invalid SMTP response [%3']	
<i>Type</i>	Error	
<i>Meaning</i>	An invalid SMTP response was received from the server	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Server response received
<i>Action</i>	Make sure you are connecting to an SMTP server	
104	SMTP E ERROR (%1) [%2] Received negative answer %3 '%4' for command '%5'	
<i>Type</i>	Error	
<i>Meaning</i>	SMTP server returned an error indication.	

<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	SMTP code returned by server
	4	Explanation string returned by server
	5	Last SMTP command sent
<i>Action</i>	Server probably does not support a required option, or a parameter was incorrect (invalid email address...)	

105	SMTP E ERROR (%1) [%2] Invalid auth. method '%3' (%4)	
<i>Type</i>	Error	
<i>Meaning</i>	This message is displayed when the SASL mechanism set in the Remote Site configuration is invalid.	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	SASL mechanism used
	4	Value of the SMTP SASL mechanism used (number)
<i>Action</i>	Check Remote Site configuration.	

106	SMTP E ERROR (%1) [%2] Authentication error (%3): %4	
<i>Type</i>	Error	
<i>Meaning</i>	This message is displayed when the SASL authentication process fails.	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	SASL mechanism used
	4	Error message
<i>Action</i>	Check the SMTP server configuration and adapt the Remote Site settings.	

107	SMTP E ERROR (%1) [%2] local site '%3' or transfer '%4': Invalid Mail From Address: '%5'	
Type	Error	
Meaning	Invalid originator email address is being used (From).	
Parameters	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Local Site Ident
	4	Transfer local identifier
	5	Invalid address used (may be empty if address was not defined)
Action	Explicitly add a From email address when creating the Transfer Request or configure the Local Site with an email address.	
108	SMTP E ERROR (%1) [%2] dest. site '%3' or transfer '%4': Invalid Mail Destination Address: '%5'	
Type	Error	
Meaning	Invalid or empty destination email address is being used (To).	
Parameters	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Local Site Ident
	4	Transfer local identifier (number)
	5	Invalid address used (may be empty if address was not defined)
Action	Explicitly add a To email address when creating the Transfer Request, or configure the Remote Site with an email address. Correct the invalid address used.	
110	SMTP I SEND_BEG (%1) [%2] %3 Start sending file: %4	

<i>Type</i>	Information	
<i>Meaning</i>	File transmission has started.	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Transfer local identifier (number)
	4	Physical path of the file being sent
<i>Action</i>	None	

111 SMTP I SEND_END (%1) [%2] %3 End sending file: %4

<i>Type</i>	Information	
<i>Meaning</i>	The complete file is sent.	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Transfer local identifier (number)
	4	Physical path of the file being sent
<i>Action</i>	None	

112 SMTP E SEND_ERR (%1) [%2] %3 Error sending file: %4

<i>Type</i>	Error	
<i>Meaning</i>	An error occurred while sending the file.	
<i>Parameters</i>	1	Local connection identifier (number)
	2	Connected Site Ident
	3	Transfer local identifier (number)
	4	Physical path of the file being sent
<i>Action</i>	Check reason, error and protocol diagnostic (if available), subsequent log message or in transfer details.	

SSH File Transfer Protocol (SFTP)

This topic lists SFTP log messages.

001	SFTP E SELOPERR %1(%2) [%3] select-open error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file selection or open operations. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the parameters of the transfer request and of the Application used for the compatibility of the file characteristics with the file system attributes referenced by the transfer. Check the access security and path to make sure that Gateway can access the file. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
002	SFTP E CREOPERR %1(%2) [%3] create-open error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file creation or open operations for incoming file transmission. The local file system error is returned in parameter %6	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is used. You must have enough space on your disk to receive the file. Refer to your platform documentation to understand the local file system error.	
003	SFTP E OPENDERR %1(%2) opendir error, directory %3 : /%4	
Type	Error	
Meaning	Error on the directory opening operation. The local file system error is returned in parameter %4	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Directory name
	4	Local file system diagnostic
Action	Refer to your platform documentation to understand the local file system error. Check the parameters of the transfer request and of the Application used for the compatibility of the directory characteristics with the file system attributes referenced by the transfer. Check the access security and path to make sure that Gateway can access the directory. If a system dependent parameter is requested for directory access, you must check that the request contains its description.	
004	SFTP E CLOSDERR %1(%2) closedir error, directory %3 : %4	

<i>Type</i>	Error	
<i>Meaning</i>	Error on the file closing operation. The local file system error is returned in parameter %4	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Directory name
	4	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error.	

005 SFTP E CLODSERR %1(%2) [%3] close-deselect error, file %4/%5: %6

<i>Type</i>	Error	
<i>Meaning</i>	Error on the file closing or de-selection operations. The local file system error is returned in parameter %6	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Refer to your platform documentation to understand the local file system error.	

006 SFTP E READ_ERR %1(%2) [%3] read error, file %4/%5: %6

<i>Type</i>	Error	
<i>Meaning</i>	Read error encountered on the file to send. The local file system error is returned in parameter %6	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>read</i> privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
007	SFTP E READDERR %1(%2) [%3] read error, file %4/%5: %6	
Type	Error	
Meaning	Read error encountered on the directory list to send. The local file system error is returned in parameter %6	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>read</i> privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
008	SFTP E WRITEERR %1(%2) [%3] write error, file %4/%5: %6	

Type	Error	
Meaning	Write error encountered on the received file. The local file system error is returned in parameter %6	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier (number)
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the parameters of the Application used by Gateway for the file reception, and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is actually used. There must be enough disk space to receive the file. Refer to your platform documentation to understand the local file system error.	

009	SFTP E STAT_ERR %1(%2) [%3] stat error, file %4/%5: %6	
Type	Error	
Meaning	Error reading the info in the indicated file. This message indicates the local file system error.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
Action	Check the message contents for information about the cause of the problem.	

010	SFTP E ABORTIND %1(%2) ABORT received: %3 %4		
Type	Error		
Meaning	Abort received on SFTP session		
Parameters	1	Remote Site name	
	2	Connection identifier (number)	
	3	Type of the error reported	
	4	Reason for the abort	
Action	If the problem is a recurring one, check the remote and Local Site definitions. If the abort occurs in data phase, check the transfer parameters used for the Transfer Request.		
011	SFTP E ABORTREQ %1(%2) ABORT sent : %3 %4		
Type	Error		
Meaning	Abort sent to SFTP session		
Parameters	1	Remote Site name	
	2	Connection identifier (number)	
	3	Type of the error reported	
	4	Reason for the abort	
Action	If the problem is a recurring one, check the Remote and Local Site definitions. If the abort occurs in data phase, check the transfer parameters used for the Transfer Request.		
016	SFTP I XFERRCV1 %1(%2) [%3] begin receiving to %4, file: %5 %6		
Type	Information		
Meaning	Information message indicating a start of reception		

Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	Identification of the receiver
	5	Directory path of the file
	6	Received file name
Action	None	

017	SFTP I XFERRCV2 %1(%2) [%3] begin receiving to %4, LIST: %5	
Type	Information	
Meaning	Information message indicating a start of reception	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the transfer request
	4	File type
	5	Directory name
Action	None	

018	SFTP I XFERSND1 %1(%2) [%3] begin sending from %4, file: %5 %6	
Type	Information	
Meaning	Information message indicating a start of transmission	

Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Local identification of the transfer request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	None	

019	SFTP I XFERSND2 %1(%2) [%3] begin sending from %4, LIST: %5	
Type	Information	
Meaning	Information message indicating a start of transmission	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the transfer request
	4	File type
	5	Directory name
Action	None	

020	SFTP I XENDRCV1 %1(%2) [%3] end receiving to %4, file:%5 %6	
Type	Information	
Meaning	Information message indicating an end of reception	

Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	None	

021	SFTP I XENDRCV2 %1(%2) [%3] end receiving, %4, LIST:%5	
Type	Information	
Meaning	Information message indicating an end of reception	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the transfer request
	4	File type
	5	Directory name
Action	None	

022	SFTP I XENDSND1 %1(%2) [%3] end sending from %4, file:%5 %6	
Type	Information	
Meaning	Information message indicating an end of transmission	

Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	None	

023	SFTP I XENDSND2 %1(%2) [%3] end sending from %4, LIST:%5	
Type	Information	
Meaning	Information message indicating an end of transmission	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the transfer request
	4	File type
	5	Directory name
Action	None	

024	SFTP E XERRRCV1 %1(%2) [%3] error receiving to %4, file: %5 %6	
Type	Error	
Meaning	Reception error	

Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	
025	SFTP E XERRRCV2 %1(%2) [%3] error receiving to %4, LIST: %5	
Type	Error	
Meaning	Reception error	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local identification of the transfer request
	4	File type
	5	Directory name
Action	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	
026	SFTP E XENDSND1 %1(%2) [%3] error sending from %4, file:%5 %6	
Type	Error	
Meaning	Transmission error	

Parameters	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	Identification of the receiver
	5	Directory path of the file
	6	File name
Action	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	
027	SFTP E XENDREQ %1(%2) [%3] error sending from %4, LIST:%5	
Type	Error	
Meaning	Transmission error	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	File type
	5	Directory name
Action	Check the messages indicating the cause of the problem. Contact Axway Support if no solution is found.	
028	SFTP I CONN_ABT %1(%2) [%3] connection interrupted with %4	
Type	Information	
Meaning	Information message indicating a connection abort with no data transfer phase	

<i>Parameters</i>	1	Identification of the sender
	2	Connection identifier (number)
	3	Identification of the transfer request
	4	Remote Site name
<i>Action</i>	Check the message indicating the reason for the abort.	

029	SFTP E XERRDATA %1(%2) received data size (%3) exceeds max (%4)	
<i>Type</i>	Error	
<i>Meaning</i>	The size of SFTP requests is too big	
<i>Parameters</i>	1	Identification of the sender (Remote Site name)
	2	Connection identifier (number)
	3	Size of received packet (bytes)
	4	Maximum allowed size of packet (bytes)
<i>Action</i>	Configure the Remote Partner so that it sends smaller sized packets.	

030	SFTP E XERRDATA %1(%2) received packet size (%3) exceeds max (%4)	
<i>Type</i>	Error	
<i>Meaning</i>	The amount of data contained in a <i>write</i> request is too big	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Size of received packet (bytes)
	4	Maximum allowed size of packet (bytes)
<i>Action</i>	Configure the Remote Partner so that it sends smaller sized <i>write</i> requests.	

031	SFTP E MULTIXFR %1(%2) File : %3, monitored transfer rejected: a transfer is running in this session	
Type	Error	
Meaning	Some SFTP clients try to initiate a transfer even though a monitored transfer is already in progress in the same SSH session. Gateway does not support this.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	File name
Action	Wait until the first transfer has finished. Alternatively, deactivate the multiple transfer option (dependent on client).	
040	SFTP I VFD_REMI %1(%2) removing file: %3	
Type	Information	
Meaning	Removing the file specified in the message	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	File name
Action	None	
041	SFTP I VFD_MKDI %1(%2) creating directory: %3	
Type	Information	
Meaning	Creating the directory specified in the message	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Directory name
Action	None	

042	SFTP I VFD_RMDI %1(%2) removing directory: %3	
Type	Information	
Meaning	Removing the directory specified in the message.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Directory name
Action	None	
043	SFTP I VFD_RENI %1(%2) renaming file: %3 to %4	
Type	Information	
Meaning	Renaming the indicated file with the indicated file name.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Old file name
	4	New file name
Action	None	
044	SFTP E VFD_REME %1(%2) error removing file: %3	
Type	Error	
Meaning	Error on deletion of the file specified in the message.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	File name
Action	Verify that the file system attributes are consistent. Gateway must have <i>write</i> privileges in the directory containing the file.	

045	SFTP E VFD_MKDE %1(%2) error creating directory: %3	
Type	Error	
Meaning	Error when creating the directory specified in the message.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Directory name
Action	Verify that the file system attributes are consistent. Gateway must have <i>write</i> privileges in the parent directory.	
046	SFTP E VFD_RMDE %1(%2) error removing directory: %3	
Type	Error	
Meaning	Error in deleting the directory specified in the message	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Directory name
Action	Verify that the file system attributes are consistent. Gateway must have <i>write</i> privileges in the parent directory. The directory must be empty.	
047	SFTP E VFD_RENE %1(%2) error renaming file: %3 to %4	
Type	Error	
Meaning	Error in the naming of the file specified in the message	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Old name
	4	New name
Action	Verify that the file system attributes are consistent. Gateway must have <i>write</i> privileges in the directory containing the file.	

050	SFTP E TRNFXERR %1(%2) [%3] transcode file to xfer error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error in transcoding the file specified in the message. This message indicates the local file system error.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the message contents for information about the cause of the problem. Check the application parameters to verify that the date code is correct.	

051	SFTP E TRNXFERR %1(%2) [%3] transcode xfer to file error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error in transcoding the file specified in the message. This message indicates the local file system error.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local request identifier
	4	Access path to the file
	5	File name
	6	Local file system diagnostic
<i>Action</i>	Check the message contents for information about the cause of the problem. Check the application parameters to verify that the date code is correct.	

101	SFTP I TRACE %1	
Type	Information	
Meaning	Use this message to trace SFTP raw data sent to the Remote Site	
Parameters	1	Variable area for editing one line of data dump
Action	Not applicable	
102	SFTP I TRACE %1 (%2) ID : %3 %4	
Type	Information	
Meaning	Use this message to trace SFTP protocol data units sent to the Remote Site.	
Parameters	1	Identification of the sender
	2	Data unit direction (Incoming/Outgoing)
	3	SFTP request identifier
	4	SFTP packet type
Action	Not applicable	
103	SFTP I TRACE %1 (%2) ID : %3 %4 %5 : %6	
Type	Information	
Meaning	Use this message to trace SFTP protocol data units sent to the Remote Site.	
Parameters	1	Identification of the sender
	2	Data unit direction (Incoming/Outgoing)
	3	SFTP request identifier
	4	SFTP packet type
	5	SFTP parameter name
	6	SFTP parameter value
Action	Not applicable	

104	SFTP I TRACE %1 (%2) ID : %3 %4 %5 :%6 %7 : %8	
<i>Type</i>	Information	
<i>Meaning</i>	This message is used to trace SFTP protocol data units sent to the Remote Site	
<i>Parameters</i>	1	Identification of the sender
	2	Data unit direction (Incoming/Outgoing)
	3	SFTP request identifier
	4	SFTP packet type
	5	SFTP parameter name
	6	SFTP parameter value
	7	SFTP parameter name
	8	SFTP parameter value
<i>Action</i>	Not applicable	

Axway Gateway: Messages and codes

Statistics and script resolution modules (SR STATS)

This topic lists SR STATS log messages.

001	SR E SBUNDEF undefined symbol %1 in %2 line %3	
<i>Type</i>	Error	
<i>Meaning</i>	The symbol name is undefined.	
<i>Parameters</i>	1	Symbol name
	2	File name or parameter
	3	Line number in the file
<i>Action</i>	Correct the symbol name.	

002	SR E SBINVAL invalid symbol in %1 line %1	
Type	Error	
Meaning	The symbol name is undefined.	
Parameters	1	File name or parameter
	2	Line number in the file
Action	Check for an invalid syntax symbol and correct the syntax.	
003	SR E LNTLONG line %1 too long in %2 script file	
Type	Error	
Meaning	The line length is greater than 240 characters.	
Parameters	1	Line number
	2	Script file name
Action	Reduce the line length to 240 characters.	
004	SR E OPENERR open error: %1 file	
Type	Error	
Meaning	Error opening the file which either does not exist or is impossible to create.	
Parameters	1	File name
Action	Check for the error and correct it.	
005	SR E READERR read error: %1 file line %2	
Type	Error	
Meaning	Error reading the file.	

Parameters	1	File name
	2	Line number in the file
Action	Check for the error and correct it.	
006	SR E SPINVAL resolved script path %1 is invalid	
Type	Error	
Meaning	The resolved script path either does not contain the x_local_ident symbol or contains other symbols.	
Parameters	1	Resolved script path
Action	Correct the resolved script path that must contain the x_local_ident symbol, and only this symbol.	
007	SR E ALLOCERR memory allocation failed: STAT record format %1	
Type	Error	
Meaning	Not enough memory to handle the STAT record format.	
Parameters	1	STAT record format number
Action	None	
008	SR E TOOFRMT too many STAT record formats defined: maximum %1	
Type	Error	
Meaning	The maximum number of STAT record format defined in the STAT data dictionary file has been exceeded.	
Parameters	1	Maximum number of STAT record formats
Action	Reduce the number of STAT record formats.	
009	SR E FRINVAL invalid or undefined STAT record format %	

<i>Type</i>	Error
<i>Meaning</i>	The maximum number of STAT record format defined in the STAT data dictionary file has been exceeded.
<i>Parameters</i>	1 STAT record format number
<i>Action</i>	If the STAT record format is invalid, a previous error message is logged describing the error. If it is undefined, check for the error in the data dictionary STAT file or in the exit function <i>ExitStatMessage</i> , and correct it.

010 SR E RCTSHRT STAT record length exceeds %1 characters: format %2

<i>Type</i>	Error
<i>Meaning</i>	The stat record length exceeds the maximum record length.
<i>Parameters</i>	1 Maximum record length
	2 STAT record format number
<i>Action</i>	Modify the stat record.

Axway Gateway: Messages and codes

Supervisor module (SUP)

This topic lists Supervisor module log messages.

001	SUP I INIT_RQ %1(%2) connection request, requester mode
<i>Type</i>	Information
<i>Meaning</i>	The Local Site is requesting a connection to a Remote Site (Initiator mode connection for the Local Site).
<i>Parameters</i>	1 Remote Site name
	2 Connection identifier (number)
<i>Action</i>	None

002	SUP I INIT_ID %1(%2) connection indication, server mode	
<i>Type</i>	Information	
<i>Meaning</i>	The Local Site is requesting a connection to a Remote Site (requester mode connection for the Local Site).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	None	
003	SUP I INIT_RP+ %1(%2) connection in server mode accepted	
<i>Type</i>	Information	
<i>Meaning</i>	The Local Site has accepted the incoming connection request (Responder mode connection for the Local Site).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
<i>Action</i>	None	
004	SUP I INIT_RP- %1(%2) connection in server mode refused: reason=%3	
<i>Type</i>	Information	
<i>Meaning</i>	The Local Site has rejected the incoming connection request (Responder mode connection for the Local Site).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Rejection reason (monitor diagnostic)
<i>Action</i>	Check the settings of the Remote Site. If necessary, also verify the correct initialization and startup of the underlying communication interface to reach the Remote Site.	

005	SUP I INIT_CF+ %1(%2) connection in requester mode accepted	
Type	Information	
Meaning	The Remote Site has accepted the incoming connection request (Initiator mode connection for the Local Site).	
Parameters	1	Remote Site name
	2	Connection identifier (number)
Action	None	
006	SUP W INIT_CF- %1(%2) connection in requester mode refused: reason=%3 / prot.diag=%4	
Type	Warning	
Meaning	The Remote Site or the file transfer module has rejected the incoming connection request (Initiator mode connection for the Local Site).	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Rejection reason (monitor diagnostic)
	4	Protocol diagnostic code (if any)
Action	Check the settings of the Remote Site. If necessary, also verify the correct initialization and start of the underlying communication interface to reach the Remote Site. Check the settings of the Transfer Request for a refusal by the transfer module.	
007	SUP I TERM_RQ %1(%2) end connection request: reason=%3	
Type	Information	
Meaning	The Local Site has requested the release of the connection with the Remote Site.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Release reason (monitor diagnostic)
Action	None	
008	SUP I TERM_ID %1(%2) end of connection indication: reason=%3 / prot.diag=%4	
Type	Information	
Meaning	The Local Site has received a connection release indication from the Remote Site. The connection is considered ended when this indication is received. The monitor does not reply to this indication.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Release reason (monitor diagnostic)
	4	Release reason (protocol diagnostic)
Action	None	
009	SUP I TERM_CF %1(%2) end of connection confirmation	
Type	Information	
Meaning	The end of connection requested by the monitor has been confirmed.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
Action	None	
010	SUP W ABORT_RQ %1(%2) connection abort request: reason=%3	
Type	Warning	

Meaning	The Local Site has requested the abort of the connection with the Remote Site.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Abort reason (monitor diagnostic)
Action	Analyze the diagnostic for the monitor to determine the reason for the unexpected end of connection.	
011	SUP W ABORT_ID %1(%2) connection abort indication: reason=%3 / prot.diag=%4 / error=%5	
Type	Warning	
Meaning	The Local Site has received a connection abort indication from the Remote Site or from the file transfer module.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Abort reason (monitor diagnostic)
	4	Abort reason (protocol diagnostic)
	5	Abort reason (other diagnostics: system error, transfer process error, ...)
Action	Analyze the diagnostics supplied to understand the reason for the unexpected end of connection.	
012	SUP I XSND_RQ %1(%2) [%3] transfer request, transmit file %4/%5, srce %6, dest %7	
Type	Information	
Meaning	The Local Site has requested the start of the file emission procedure.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Access path to the file
	5	File name
	6	Origin Site
	7	Destination Site
<i>Action</i>	None	

013	SUP I XRCV_RQ %1(%2) [%3] transfer request, receive file %4/%5, srce %6 dest %7	
<i>Type</i>	Information	
<i>Meaning</i>	The Local Site has requested the start of the file reception procedure.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Access path to the file
	5	File name
	6	Origin Site
	7	Destination Site
<i>Action</i>	None	

014	SUP I XSND_ID %1(%2) transfer indication, transmit file, srce %3 dest %4	
<i>Type</i>	Information	

<i>Meaning</i>	The Remote Site has requested the start of the file emission procedure (file transmission to the Local Site).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Origin Site
	4	Destination Site
<i>Action</i>	None	

015	SUP I XRCV_ID %1(%2) transfer indication, receive file, srce %3 dest %

<i>Type</i>	Information	
<i>Meaning</i>	The Remote Site has requested the start of the file reception procedure (file transmission from the Local Site).	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Origin Site
	4	Destination Site
<i>Action</i>	None	

016	SUP I XBEG_CF+ %1(%2) [%3] transfer in requester mode accepted

<i>Type</i>	Information	
<i>Meaning</i>	The Remote Site has accepted the file transfer.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
<i>Action</i>	None	

017	SUP E XBEG_CF- %1(%2) [%3] transfer in requester mode refused: reason=%4 / prot.diag=%5 / error=%6	
<i>Type</i>	Error	
<i>Meaning</i>	The Remote Site or the file transfer module has refused the file transfer.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Abort reason (monitor diagnostic)
	5	Abort reason (protocol diagnostic)
	6	Abort reason (other diagnostics: system error, transfer process error, ...)
<i>Action</i>	Analyze the diagnostics supplied to understand the refusal reason. Check the settings of the Remote Site. If necessary, verify the correct initialization and startup of the underlying communication interface to reach the Remote Site. Check the settings of the Transfer Request for a refusal by the transfer module.	
018	SUP I XBEG_RP+ %1(%2) [%3] transfer in server mode accepted	
<i>Type</i>	Information	
<i>Meaning</i>	The Local Site has accepted the incoming Transfer Request.	
<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
<i>Action</i>	None	
019	SUP I XBEG_RP- %1(%2) [%3] transfer in server mode refused: reason=%4	

Type	Information		
Meaning	The Local Site has rejected the incoming Transfer Request.		
Parameters	1	Remote Site name	
	2	Connection identifier (number)	
	3	Local transfer identifier (number)	
	4	Rejection reason (monitor diagnostic)	
Action	Analyze the diagnostic supplied to understand the rejection reason. Check the settings of the Remote Site. If necessary, also verify the correct initialization and start of the underlying communication interface to reach the Remote Site. Check the settings of the Transfer Request for a refusal by the transfer module.		

020	SUP I XEND_RQ %1(%2) end transfer request %3: reason=%4		
Type	Information		
Meaning	The Local Site has requested a premature end of transfer (transfer abort or suspension).		
Parameters	1	Remote Site name	
	2	Connection identifier (number)	
	3	Ending reason (monitor diagnostic)	
Action	A suspended transfer will be automatically restarted later.		

021	SUP I XEND_ID %1(%2) [%3] end of transfer indication: reason=%4 / prot.diag=%5 / error=%6		
Type	Information		
Meaning	The Local Site has received from the Remote Site an indication for the end of transfer		

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Abort reason (monitor diagnostic)
	5	Abort reason (protocol diagnostic)
	6	Abort reason (other diagnostics: system error, transfer process error, ...)
Action	In case one of the diagnostics indicates an error, do the following: • Check the settings of the Remote Site. If necessary, also verify the correct initialization and startup of the underlying communication interface to reach the Remote Site. • Check the settings of the Transfer Request for a refusal by the transfer module.	
022	SUP I XEND_RP %1(%2) [%3] effective end of transfer	
Type	Information	
Meaning	The response to an end of transfer indication has been made	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
Action	None	
023	SUP I XEND_CF %1(%2) [%3] end of transfer confirmation	
Type	Information	
Meaning	The response to an end-of-transfer indication has been made.	

Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
Action	None	
024	SUP E XEND_CF %1(%2) [%3] routing failure: %4 , ErrCod=%5	
Type	Error	
Meaning	The monitor has failed to create a Transfer Request to forward the received file. The reason for the failure indicated by %4 and % 5 gives the corresponding system error if any.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Failure reason (monitor diagnostic)
	5	Corresponding local error (if any)
Action	Analyze the diagnostics supplied to correct the problem.	
025	SUP I SITE warning! local site does not correspond to the configuration	
Type	Information	
Meaning	The existing Local Site does not correspond to the default Local Site identifier defined during the monitor installation procedure.	
Parameters	None	
Action	Verify that you have created the default Local Site for the monitor (with the name defined by the <code>local_proto_ident</code> parameter in the monitor installation file).	

026	SUP I SITE warning! no local site defined	
Type	Information	
Meaning	The default Local Site identifier for the monitor is not created. It corresponds to the Local Site identifier, defined during the monitor installation procedure.	
Parameters	None	
Action	Create the monitor default Local Site.	
027	SUP E XEND_CF %1(%2) [%3] routing failure towards %4: %5 %6, ErrCod=%7	
Type	Error	
Meaning	The monitor was not able to route the Transfer Request.	
Parameters	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Routing Site name
	5	Failure reason (monitor diagnostic)
	6	Object name link to the failure reason
	7	Corresponding local error (if any)
Action	Analyze the diagnostics supplied to correct the problem.	
028	SUP I XEND_CF %1(%2) [%3] routing towards %4 ident:%5 model:%6	
Type	Information	
Meaning	The monitor routes the transfer to the Site %4.	

<i>Parameters</i>	1	Remote Site name
	2	Connection identifier (number)
	3	Local transfer identifier (number)
	4	Routing Site name
	5	Transfer ident
	6	Model name
<i>Action</i>	None	

030	SUP E BADAPPLI %1: Receive application "%2" doesn't exist	
<i>Type</i>	Error	
<i>Meaning</i>	The monitor was not able to find the Application (FTP protocol).	
<i>Parameters</i>	1	Remote Site name
	2	Application name
<i>Action</i>	Create a Gateway Application with this name.	

031	SUP E BADACK no transfer record found for ACK msg (%1)	
<i>Type</i>	Error	
<i>Meaning</i>	The monitor was not able to find the Transfer Request associated with the acknowledgement request	
<i>Parameters</i>	1	Transfer identifier
<i>Action</i>	Check the acknowledgement request parameters	

032	SUP E ACKFAIL transfer (%1) acknowledgment update fails	
<i>Type</i>	Error	

Meaning	The monitor was not able to update the Transfer Request associated with the acknowledgement request	
Parameters	1	Transfer identifier
Action	Check the acknowledgement request parameters.	
033	SUP W ACKIGN transfer (%1) acknowledgment ignored	
Type	Warning	
Meaning	The Site acknowledgement option was set to "NOT_TO_ACK". The transfer associated with the acknowledgement request is not modified.	
Parameters	1	Transfer identifier
Action	None	
034	SUP E ACKFAIL transfer (%1) couldn't send acknowledgment ('%2')	
Type	Error	
Meaning	Gateway was unable to send acknowledgement	
Parameters	1	Transfer identifier
	2	Context specific message
Action	Action depends on parameter 2 message content. Check partner configuration.	
036	SUP I CONNHOLD connection held for upcoming transfers. wait interval: %1	
Type	Information	
Meaning	The Site acknowledgement option was set to "NOT_TO_ACK". The transfer associated with the acknowledgement request is not modified.	
Parameters	1	Wait interval in seconds
Action	None	

040	SUP E KEY Invalid product key: %1
<i>Type</i>	Error
<i>Meaning</i>	The software protection key defined during the monitor installation procedure is not valid
<i>Parameters</i>	1 Key value used
<i>Action</i>	Check the software protection key supplied by the software delivery team. If it is the same as the one contained in the message, contact Axway Support.
041	SUP E KEY Product code or site name invalid. rc=%1
<i>Type</i>	Error
<i>Meaning</i>	The software protection key is not valid.
<i>Parameters</i>	1 Error code returned by the key checking routine
<i>Action</i>	Contact Axway Support.
042	SUP E KEY Product code or site name invalid. rc=%1
<i>Type</i>	Error
<i>Meaning</i>	The software protection key is not valid
<i>Parameters</i>	1 Error code returned by the key checking routine
<i>Action</i>	Contact Axway Support.
043	SUP E KEY Retention date expired
<i>Type</i>	Error
<i>Meaning</i>	The software protection key is not valid, because the retention date has expired
<i>Parameters</i>	None
<i>Action</i>	Contact Axway Support.

044	SUP W KEY Retention date about to expire. credit=%1 days	
Type	Warning	
Meaning	Warning message indicating that the retention date expires in %1 days. After this delay, the software protection key becomes invalid.	
Parameters	1	Number of days remaining for the validity of the protection key
Action	Contact Axway Support	
045	SUP E KEY Number of authorized connections exceeded. max=%1	
Type	Error	
Meaning	The number of simultaneous connections has exceeded the maximum value defined in the software protection key. Above the maximum, further connection requests are rejected.	
Parameters	1	Maximum number of simultaneous connections authorized
Action	Contact Axway Support.	
046	SUP E KEY Number of authorized sites exceeded. max=%1	
Type	Error	
Meaning	The number of simultaneous connections has exceeded the maximum value defined in the software protection key. Above the maximum, further connection requests are rejected.	
Parameters	1	Maximum number of simultaneous connections authorized
Action	Contact Axway Support.	
047	SUP E KEY Protocol not allowed or invalid	
Type	Error	
Meaning	The requested protocol is not allowed in the software protection key	
Parameters	None	
Action	Contact Axway Support.	

048	SUP I KEY key=%1, site=%2	
<i>Type</i>	Information	
<i>Meaning</i>	Information message printed after a successful verification of the software protection key	
<i>Parameters</i>	1	Protection key
	2	Default Local Site defined during the installation
<i>Action</i>	None	
049	SUP I KEY protocol option: %1	
<i>Type</i>	Information	
<i>Meaning</i>	List of the file transfer protocols authorized by the software protection key	
<i>Parameters</i>	1	List of file transfer protocols
	2	Default Local Site defined during the installation
<i>Action</i>	None	
050	SUP I KEY retention date=%1 %2	
<i>Type</i>	Information	
<i>Meaning</i>	Retention date defined in the software protection key	
<i>Parameters</i>	1	Two digits specifying the year (ex. 97 for 1997)
	2	Three digits specifying the Julian day of the year (001 to 365)
<i>Action</i>	None	
051	SUP I KEY connection limits: connection=%1, site=%2	
<i>Type</i>	Information	

<i>Meaning</i>	Maximum number of active connections and Sites allowed. They are defined in the software protection key.	
<i>Parameters</i>	1	Maximum number of simultaneous connections
	2	Maximum number of Sites
<i>Action</i>	None	

054	SUP E KEY Protocol %1 not allowed, canceling transfer %2	
<i>Type</i>	Error	
<i>Meaning</i>	The requested protocol is not allowed in the software protection key, therefore the transfer was canceled.	
<i>Parameters</i>	1	Which protocol is not allowed
	2	Number of transfer that was canceled
<i>Action</i>	Contact Axway Support.	

060	SUP I INI-CTOR Initializing connector '%1'	
<i>Type</i>	Information	
<i>Meaning</i>	Initializing specified connector. A connector manages a queue associated with a file.	
<i>Parameters</i>	1	One of the following: • Sentinel • Router (connector responsible for routing)
<i>Action</i>	None	

061	SUP I ST-CPROC Starting '%1' connector process number %2	
<i>Type</i>	Information	
<i>Meaning</i>	Starting connector associated with specified process	

Parameters	1	Name of connector: either Sentinel or Router
	2	Process number (a single process is allowed per queue)
Action	None	
062	SUP I CPROCUP '%1' connector process number %2 now UP (pid=%3)	
Type	Information	
Meaning	Specified connector is ready	
Parameters	1	Name of connector: either Sentinel or Router
	2	Process number (a single process is allowed per queue)
	3	Process system identifier
Action	None	
063	SUP I CPROCDOWN '%1' connector process number %2 now DOWN (pid=%3)	
Type	Information	
Meaning	Specified connector is stopped	
Parameters	1	Name of connector: either Sentinel or Router
	2	Process number (a single process is allowed per queue)
	3	Process system identifier
Action	None	
064	SUP W SUSSCHED '%1' %2 %% full, transfer scheduling suspended	
Type	Warning	

<i>Meaning</i>	The specified queue has reached a critical threshold. To avoid congestion, transfer scheduling is suspended.	
<i>Parameters</i>	1	Path to the queue
	2	Percentage of queue filled
<i>Action</i>	If queue congestion is temporary, transfer scheduling will restart when the queue is no longer full. If the queue corresponds to a Sentinel connector, check that Sentinel is active.	

065	SUP I SUSSCHED Transfer scheduling resumed	
<i>Type</i>	Information	
<i>Meaning</i>	Queue level below critical threshold: transfer scheduling resumed.	
<i>Parameters</i>	None	
<i>Action</i>	None	

066	SUP W REFUCONN '%1' %2 %% full, new connections will be refused	
<i>Type</i>	Warning	
<i>Meaning</i>	The specified queue has reached a critical threshold. All new connections are refused.	
<i>Parameters</i>	1	Path to the queue
	2	Percentage of queue filled
<i>Action</i>	If queue congestion is temporary, connections will be accepted when the queue is no longer full. If the queue corresponds to a Sentinel connector, check that Sentinel is active.	

067	SUP I ACCPCONN New connections will be accepted again	
<i>Type</i>	Information	
<i>Meaning</i>	Queue level below critical threshold: new connections accepted again.	

<i>Parameters</i>	None
<i>Action</i>	None
068	SUP W REFUXFER '%1' %2 %%% full, new transfers will be refused
<i>Type</i>	Warning
<i>Meaning</i>	The specified queue has reached a critical threshold. All new transfers are refused.
<i>Parameters</i>	1 Path to the queue
	2 Percentage of queue filled
<i>Action</i>	If queue congestion is temporary, transfers will be accepted when the queue is no longer full. If the queue corresponds to a Sentinel connector, check that Sentinel is active.
069	SUP I ACCPXFER New transfers will be accepted again
<i>Type</i>	Information
<i>Meaning</i>	Queue level below critical threshold: new transfers accepted again.
<i>Parameters</i>	None
<i>Action</i>	None
070	SUP W REFUUSER '%1' %2 %%% full, new users will be refused
<i>Type</i>	Warning
<i>Meaning</i>	The specified queue has reached a critical threshold. All new user logins are refused.
<i>Parameters</i>	1 Path to the queue
	2 Percentage of queue filled
<i>Action</i>	If queue congestion is temporary, user logins will be accepted when the queue is no longer full. If the queue corresponds to a Sentinel connector, check that Sentinel is active.

071	SUP I ACCUSER New users will be accepted again	
Type	Information	
Meaning	Queue level below critical threshold: new users accepted again	
Parameters	None	
Action	None	
072	SUP E PMQWRERR Write to file '%1' failed (ErrCod=%2)	
Type	Error	
Meaning	File write error	
Parameters	1	File name
	2	Error code number & message
Action	Check that the file is not full	
073	SUP I RE-CPROC Restarting '%1' connector process number %2	
Type	Information	
Meaning	Restarting a process	
Parameters	1	Connector name
	2	Process number
Action	None	
080	SUP I INI-HP Initializing helper pool '%1'	
Type	Information	
Meaning	The SUP process is initializing the helper pool	
Parameters	1	Name of process

Action	None	
081	SUP I ST-HPROC Starting '%1' helper process number %2	
Type	Information	
Meaning	Helper process is starting up	
Parameters	1	Helper process name
	2	Helper process number
Action	None	
082	SUP I HPROCUP '%1' helper process number %2 now UP (pid=%3)	
Type	Information	
Meaning	Helper process is starting operating	
Parameters	1	Helper process name
	2	Helper process number
	3	Helper process identification
Action	None	
083	SUP I HPROCDOWN '%1' helper process number %2 now DOWN (pid=%3)	
Type	Information	
Meaning	Helper process is not operating	
Parameters	1	Helper process name
	2	Helper process number
	3	Helper process identification
Action	None	

084	SUP E EDIPRERR [%1] EDI processing error: reason=%2 ErrCod=%3 errno=%4	
Type	Error	
Meaning	Error on processing EDI protocol	
Parameters	1	Transfer Id
	2	Reason for error (string)
	3	Error code
	4	Error number
Action	The solution for the error depends on the returned message. In most cases this error indicates a problem in the Transfer partner security configuration.	
085	SUP I RE-HPROC Recycling '%1' helper process number %2	
Type	Information	
Meaning	A process is stopped and then restarted after a certain number of operations to free up memory.	
Parameters	1	Connector name
	2	Process number
Action	None	
101	SUP W SITE-DOWN Site %1 disabled due to maximum retry count (%2) reached	
Type	Warning	
Meaning	The monitor has disabled the Remote Site because the maximum number of consecutive connection attempts to that site has been reached. The monitor does not attempt further connection to that Site.	
Parameters	1	Remote Site name
	2	Maximum number of connection retries

<i>Action</i>	The user must enable that Site using online commands or a graphical user interface, so the monitor can again schedule file transfer on the Site. Enabling a Site resets its connection counters (retry count and retry delays).	
102	SUP I U-LOGIN User %1 (%2) pid %3 has logged in	
<i>Type</i>	Information	
<i>Meaning</i>	A remote user has logged in.	
<i>Parameters</i>	1	User
	2	IP of the user
	3	Process id
<i>Action</i>	None	
103	SUP I U-LOGOUT User %1 (%2) pid %3 logged out	
<i>Type</i>	Information	
<i>Meaning</i>	A remote user has logged out	
<i>Parameters</i>	1	User
	2	IP of the user
	3	Process id
<i>Action</i>	None	
104	SUP W U-EXITED User %1 pid %2 abnormally exited	
<i>Type</i>	Warning	
<i>Meaning</i>	A user process exited without properly closing its resources. This message typically appears when a user Application program does not call ItpCloseSession() before exiting, or when such a program or a Gateway tool is killed.	

<i>Parameters</i>	1	No meaning (typically "unknown")
	2	Process id
<i>Action</i>	None	
105	SUP W U-FAIL login refused to %1: unauthorized access	
<i>Type</i>	Warning	
<i>Meaning</i>	The user name and/or password is invalid	
<i>Parameters</i>	1	User name
<i>Action</i>	Use a correct user name and a correct password.	
106	SUP I ALT_COMM Using alternate communication path Nb: %1 to site %2	
<i>Type</i>	Information	
<i>Meaning</i>	All the connection attempts to the Site have failed with the last communication path. So the monitor tries to reach the Site by using the next communication path numbered %1 available.	
<i>Parameters</i>	1	Path number (1, 2, or 3)
	2	Alias Site identifier
<i>Action</i>	Solve the communication error on the other communication paths	
107	SUP I MAIN_COMM Using main communication path to site %2	
<i>Type</i>	Information	
<i>Meaning</i>	The communication path has been changed by a user action. It is the main one that is used to reach the Remote Site.	
<i>Parameters</i>	1	Alias Site identifier
<i>Action</i>	None	

108	SUP I SITE-RTRY Site %1 [%2] retry connection (%3/%4) in %5 seconds	
Type	Information	
Meaning	Indicates a connection retry event in <i>n</i> seconds	
Parameters	1	Site alias
	2	Transfer ID
	3	Retry count
	4	Maximum allowable retry attempts
	5	Seconds before next retry
Action	None	
109	SUP I TO_ALTER Site %1 [%2] maximum retry(%3) reached. Switch to alternate communication path Nb: %4	
Type	Information	
Meaning	Indicates a switch to connection attempts on a new communication path	
Parameters	1	Site alias
	2	Transfer ID
	3	Maximum allowable retry attempts
	4	New communication path number
Action	None	
110	SUP I TP_MAIN Site %1 [%2] maximum retry(%3) reached. Switch to main communication path	
Type	Information	
Meaning	Indicates a switch to connection attempts on the main communication path	

Parameters	1	Site alias
	2	Transfer ID
	3	Maximum allowable retry attempts
Action	None	
111	SUP I N-ATTACH Notification script attached to user %1 pid %2	
112	SUP I N-DETACH Notification script detached from user %1 pid %2	
113	SUP I N-STARTED User notification process %1 started	
Type	Information	
Meaning	A user process dedicated to an external user exit routine has been started.	
Parameters	1	Path name of the program to execute
Action	None	
114	SUP W N-HIGH-WM Notif high water mark reached(%1)	
Type	Warning	
Meaning	This message indicates that user processes executing external exits are not able to dispatch execution of exit frames at the speed of incoming events. It means that the number of awaiting exit frames queued by Gateway has reached a maximum value defined in the configuration file ("high water mark").	
Parameters	1	Actual number of awaiting exit frames queued by Gateway
Action	Gateway normally recovers from this warning automatically. It may have to reject incoming connections in order to decrease the amount of its resources actually used. You can try to decrease the average response time of the user exit routines.	
115	SUP I N-LOW-WM Notif low water mark reached(%1)	
Type	Information	

Meaning	This message indicates that user processes executing external exits have dispatched enough awaiting exit frames queued by Gateway to decrease their number under a minimal value defined in the configuration file ("low water mark").	
Parameters	1	Actual number of awaiting exit frames queued by Gateway
Action	None	
116	SUP I N-NO-PROC No user process available for notifications	
Type	Information	
Meaning	This message indicates that all process context resources are used, and that Gateway is not allowed to create additional user processes	
Parameters	None	
Action	Gateway normally recovers from this warning automatically. It may have to reject incoming connections in order to decrease the amount of its resources actually used. You can allow Gateway to start more user processes by increasing the configuration parameter <code>notif_process_started_max</code> .	
117	SUP I N-CONFIG Origin set enabled=(%1), processes to start=%2, max processes=%3	
Type	Information	
Meaning	Information message written by the external user exit manager at Gateway startup	
Parameters	1	Exit families allowed Default families are (ft, xfer)
	2	Number of user exit processes to create with Gateway startup
	3	Maximum number of user exit processes Gateway is allowed to manage simultaneously
Action	None	

118	SUP I N-OPENSET NotifOpenSet, type set=(%1), script max=(%2)	
<i>Type</i>	Information	
<i>Meaning</i>	Information message written once a user exit process has been restarted successfully	
<i>Parameters</i>	1	Exit families allowed Default families are (ft, xfer)
	2	Maximum number of transfer contexts (active transfers) this process can manage simultaneously
<i>Action</i>	None	
119	SUP I N-CLOSESET NotifCloseSet, type set=(%1), script max=(%2)	
<i>Type</i>	Information	
<i>Meaning</i>	Information message written when a user exit process has freed its resources before exiting	
<i>Parameters</i>	1	Exit families allowed Default families are (ft, xfer)
	2	Maximum number of transfer contexts (active transfers) this process can manage simultaneously
<i>Action</i>	None	
120	SUP W N-BUGGY User notification process problem, type set=(%1), required=(%3), script max=(%2)	
<i>Type</i>	Warning	
<i>Meaning</i>	Gateway has detected abnormal user exit process	

<i>Parameters</i>	1	Exit families allowed Default families are (ft, xfer)
	2	Exit families this process is able to handle
	3	Maximum number of transfer contexts (active transfers) the process can manage simultaneously
<i>Action</i>	None	

121	SUP I UN-INIT NotifInit by user %1 pid %2 max scripts %3	
<i>Type</i>	Information	
<i>Meaning</i>	<i>Message reserved for Gateway Support debug mode.</i> A user exit process has exited before performing its initialization.	
<i>Parameters</i>	1	Command line used to start the process
<i>Action</i>	Check the command line and the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	

122	SUP I UN-END NotifEnd by user %1 pid %2	
<i>Type</i>	Information	
<i>Meaning</i>	<i>Message reserved for Gateway Support debug mode.</i> A user exit process has exited before performing its initialization.	
<i>Parameters</i>	1	Command line used to start the process
<i>Action</i>	Check the command line and the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	

123	SUP I UN-GETIND NotifGetInd ident=%3 context=%4 get by user %1 pid %2	
<i>Type</i>	Information	
<i>Meaning</i>	<i>Message reserved for Gateway Support debug mode.</i> A user exit process has exited before performing its initialization.	

Parameters	1	Command line used to start the process
Action	Check the command line and the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	
124	SUP I UN-PUTREP NotifPutResp ident=%3 context=%4 put by user %1 pid %2	
Type	Information	
Meaning	Message reserved for Gateway Support debug mode. A user exit process has exited before performing its initialization.	
Parameters	1	Command line used to start the process
Action	Check the command line and the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	
125	SUP I UN-GETPEN NotifGet pending for user %1 pid %2	
Type	Information	
Meaning	Message reserved for Gateway Support debug mode. A user exit process has exited before performing its initialization.	
Parameters	1	Command line used to start the process
Action	Check the command line and the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	
126	SUP I N-FAILURE User notif process start failure %1	
Type	Information	
Meaning	A user exit process has exited before performing its initialization	
Parameters	1	Command line used to start the process
Action	Check the command line and the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	

127	SUP I N-EXITED Abnormal exiting of a user exit process %1 pid %2	
<i>Type</i>	Information	
<i>Meaning</i>	Gateway has detected an abnormal exiting of a user exit process	
<i>Parameters</i>	1	Exit families allowed. Default families are (ft, xfer)
	2	Exit process id
<i>Action</i>	Check the output of the corresponding process. If you cannot correct the problem, contact Axway Support.	
151	SUP E FN-XFERER Notif=%1, Ident=%2, ERROR=%7, Site=%3, Appli=%4, Yday=%5,Nth=%6	
<i>Type</i>	Error	
<i>Meaning</i>	This message is provided for user external exit debugging. This message is written when the configuration parameter <code>notif_log_level</code> is set to a value of 3 or greater.	
<i>Parameters</i>	1	Name of the user exit notification frame. Other parameters correspond to different fields of the parameter area passed to the exit routine.
<i>Action</i>	No special action	
150	SUP I FN-XFEROK Notif=%1, Ident=%2, Site=%3, Appli=%4, Yday=%5, Nth=%6	
152	SUP I FN-XFERUN Notif=%1, Ident=%2, NO CHANGED Todo=%3, Reason=%4,Reason lib=%5	
154	SUP I XN-XFEROK Notif=%1, Ident=%2, State=%7, Site=%3, Appli=%4, Yday=%5, Nth=%6	
157	SUP I FN-INITUN Notif=%1, %2 <-> %3, Todo=%4, Reason=%5, Reason lib=%6	
160	SUP I IN-INITOK Notif=%1, org=%3, dest=%4	
161	SUP I IN-INITER Notif=%1, INIT FAILED org=%3, dest=%4	

162	SUP I FN-TERABOK Notif=%1, site %2	
163	SUP I FN-TERABER Notif=%1, FAILED site %2	
164	SUP I FN-PCNXUN Type=%1, Protocol=%2, User=%3, NOT CHANGED	
165	SUP I FN-PCNXCH Type=%1, Protocol=%2, User=%3, CHANGED Todo=%4, Reason=%5, Reason lib=%6	
167	SUP I FN-PCNXOK Type=%1, Protocol=%2, User=%3, NO RESPONSE	
169	SUP I N-RECYCL Recycling user notification process (ops_ count=%1)	
<i>Type</i>	Information	
<i>Meaning</i>	This message set is provided for user external exit debugging. These messages are written when the configuration parameter <code>notif_log_level</code> is set to a value of 3 or greater.	
<i>Parameters</i>	1	Name of the user exit notification frame. Other parameters correspond to different fields of the parameter area passed to the exit routine.
<i>Action</i>	No special action	
153	SUP W FN-XFERCH Notif=%1, Ident=%2, CHANGED Todo=%3, Reason=%4, Reason lib=%5	
155	SUP W XN-XFERER Notif=%1, Ident=%2, State=%7, ERROR=%8, Site=%3, Yday=%5, Nth=%6	
156	SUP W FN-XFERCN Notif=%1, Ident=%2, XFER CANCELED Todo=%3, Reason=%4, Reason lib=%5	
158	SUP W FN-INITCH Notif=%1, %2, CHANGED Todo=%3, Reason=%4, Reason lib=%5	
159	SUP W FN-INITCN Notif=%1, %2, INIT DELETED Todo=%3, Reason=%4, Reason lib=%5	
166	SUP W FN-PCNXCN Type=%1, Protocol=%2, User=%3, REJECTED Todo=%4, Reason=%5, Reason lib=%6	
168	SUP W FN-PCNXER Type=%1, Protocol=%2, User=%3, FAILURE	

<i>Type</i>	Warning	
<i>Meaning</i>	This message set is provided for user external exit debugging. These messages are written when the configuration parameter <code>notif_log_level</code> is set to a value of 3 or greater.	
<i>Parameters</i>	1	Name of the user exit notification frame. Other parameters correspond to different fields of the parameter area passed to the exit routine.
<i>Action</i>	No special action	

218 SUP I USER-SECU User:%1 %2:%3 Action:%4 Allowed

<i>Type</i>	Information	
<i>Meaning</i>	A user action on Gateway objects has been allowed.	
<i>Parameters</i>	1	User name
	2	Object type
	3	Object name
	4	Action type
<i>Action</i>	None	

219 SUP I USER-SECU User:%1 %2:%3 Action:%4 Denied

<i>Type</i>	Information	
<i>Meaning</i>	A user action on Gateway objects has been denied.	
<i>Parameters</i>	1	User name
	2	Object type
	3	Object name
	4	Action type
<i>Action</i>	Check the user profile.	

220	SUP W BEFNTF Before Call %1, State FT %2, type Notif %3	
221	SUP W AFTNTF State: %1, status: %2, After Call: %3	
222	SUP W NTFRESP Response of %1, User todo %2, reason lib %3, reason %4	
223	SUP W NTFNORESP After %1 without response	
<i>Type</i>	Warning	
<i>Meaning</i>	This message set is provided for user external exit debugging. These messages are written when the configuration parameter <code>notif_log_level</code> is set to a value of 3 or greater.	
<i>Parameters</i>	1	Name of the user exit notification frame. Other parameters correspond to different fields of the parameter area passed to the exit routine.
<i>Action</i>	No special action	
224	SUP W SITE-DOWN Site %1 disabled due to protocol error: %2	
<i>Type</i>	Warning	
<i>Meaning</i>	The Site is disabled due to protocol error.	
<i>Parameters</i>	1	Alias Site name
	2	Error code
<i>Action</i>	Analyze protocol trace and modify some definitions.	
225	SUP I XFERDELAY Transfer from %1 to %2, filename=%3, will start at %4in order to keep unicity of time	
<i>Type</i>	Information	
<i>Meaning</i>	This message occurs when several transfers in Odette protocol are submitted at the same second. The monitor defers the transfer to keep the Odette key identifier.	

<i>Parameters</i>	1	Alias name of the originator Site
	2	Alias name of the destination Site
	3	Protocol-level file name
	4	Beginning date
<i>Action</i>	None	

226	SUP W SDISABLC Site %1 disabled due to a missing option in the product license	
<i>Type</i>	Warning	
<i>Meaning</i>	The Site exists but the user does not have the right to use it.	
<i>Parameters</i>	1	Site name
<i>Action</i>	Check option and license.	

227	SUP E NOSITE (%1) Site %2 not found for protocol %3	
<i>Type</i>	Error	
<i>Meaning</i>	Template not found on incoming connection.	
<i>Parameters</i>	1	internal reference
	2	Site name
	3	Protocol
<i>Action</i>	Check or create Site.	

256	SUP W MIR-DOWN Mirroring of transfer database disabled due to file error	
<i>Type</i>	Warning	
<i>Meaning</i>	The Mirror Mailbox cannot be used anymore.	

<i>Parameters</i>	None
<i>Action</i>	Stop Gateway and try to recover the Mailbox.
257	SUP W MIR-ERR Error on transfer mirror database ignored
<i>Type</i>	Warning
<i>Meaning</i>	An error occurred but the <code>mirror_action_on_err</code> configuration parameter value is different from 0 (stop the monitor) and 1 (stop the mirroring)
<i>Parameters</i>	None
<i>Action</i>	None
258	SUP I PUR-BEG Begin of transfer purge session
<i>Type</i>	Information
<i>Meaning</i>	This message occurs either when a cyclic purge starts, or when a user submits a purge request.
<i>Parameters</i>	None
<i>Action</i>	None
259	SUP I PUR-END End of transfer purge session
<i>Type</i>	Information
<i>Meaning</i>	This message occurs when all purge requests of the session are completed
<i>Parameters</i>	None
<i>Action</i>	None
260	SUP I PUR-STAT %1 transfers purged for '%2' request
<i>Type</i>	Information

Meaning	This message occurs when a purge request is ended	
Parameters	1	Number of transfers purged
	2	Name of purged request
Action	None	
261	SUP E PUR-ERR Error while purging transfer, skipping request '%1' (%2 transfers purged	
Type	Error	
Meaning	An error occurs in a purge request. This purge request is skipped.	
Parameters	1	Name of purged request
	2	Number of transfers purged before error
Action	The error may be a Mailbox integrity error. Stop monitor and check Mailbox integrity (refer to the <code>pelmon</code> online command).	
262	SUP I PUR-LOAD Adding periodic purge request '%1'	
Type	Information	
Meaning	This message occurs at the startup of monitor if periodic purge parameter file is supplied	
Parameters	1	Name of purged request
Action	None	
263	SUP I PUR-ERR Error loading periodic purge request file '%1': %2	
Type	Information	
Meaning	This message occurs at the startup of monitor if periodic purge parameter file is supplied	

Parameters	1	File path of purge parameter file
	2	System error code
Action	Check purge parameter file (access rights,...)	
270	SUP I XEND_CF %1(%2) [%3] EERP routed towards %4 ident:%5	
Type	Information	
Meaning	The acknowledgement request has been routed successfully	
Parameters	1	Remote agent name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Remote agent of the routed acknowledgement request
	5	Local identifier of the routed acknowledgement request
Action	None	
271	SUP W NEERP_NR [%2] Negative EERP not routed due to negative acknowledgement option not set on site %1	
Type	Warning	
Meaning	Corresponding option not set for this Site.	
Parameters	1	Site name
	2	Transfer reference number
Action	Check license. Check/create option.	
280	SUP I XSR_LINOK SecureRelay login accepted for user %1	
Type	Information	
Meaning	A login request from Secure Relay has been accepted.	

Parameters	1	User name
Action	None	
281	SUP I XSR_LINKO SecureRelay login refused for user %1 (reason %2)	
Type	Information	
Meaning	A login request from Secure Relay has been refused.	
Parameters	1	User name
	2	Reason code
Action	Check previous message logs for further explanation.	
300	SUP E TLS_ERR %1(%2) Site sprof checkfailed (%3)	
Type	Error	
Meaning	Error while verifying matching between the used incoming security profile and the security profile in the connection Site field. The matching process takes place just after the protocol connection.	

<i>Parameters</i>	1	Connected Site protocol identifier

2	Connection identifier

3

Reason

Reason codes:

- **1:** Security context not found. Disconnection happens (and security context is deleted) before the supervisor tries to check the security profile used in the connection.
 - **10:** Incompatible type between the Site Security Profile type and the actual mode
 - **11:** Client authentication is mandatory in the Security Profile for the Site, but the client failed to identify itself during the handshake phase
 - **12:** Client authentication is imposed as anonymous in the Security Profile for the Site but the client identified itself during the handshake phase
 - **13:** Server authentication is mandatory in the Security Profile for the Site but the server did not identify itself during the handshake phase
 - **14:** Server authentication imposed as anonymous in the Security Profile for the Site but the server identified itself during the handshake phase
 - **17:** The Security Profile for the Site has an incompatible exit scheduling value with the Security Profile used
 - **18:** The Security Profile for the Site does not allow the cipher suite used
 - **19:** The Security Profile for the Site does not allow the TLS version used
 - **20:** Section "UserLocalCertificates" of the Security Profile for the Site does not contain the sent certificate (%4)
 - **21:** Templates of new Security Profile do not match received certification chain
 - **30:** The common name of the certificate received does not match the one set in the **Subject certificate alias** field of the Site
 - **31:** The common name of the issuer certificate received does not match the one set in the **Issuer certificate alias** field of the Site
 - **32:** The distinguished name of the certificate received does not match the pattern set in the **Subject name pattern** field of the Site
 - **33:** The distinguished name of the issuer certificate received does not match the pattern set in the **Issuer name pattern**
-

**Reason
codes:**

- **1:**
Security
context
not
found.
Dis-
connect
ion
has
occurred
(and
security
context
is
deleted)
before
the
supervisor
tries
to
check
-

the
sec
urit
y
pro
file
us
ed
in
the
co
nn
ect
io
n.

- **10**
:
Inc
om
pat
ible
typ
e
bet
we
en
the
Sit
e
Se
cur
ity
Pro
file
typ
e
an
d
the
act
ual
mo
de
-

- **11**
:
Cli
ent
aut
he
nti
cat
ion
is
ma
nd
ato
ry
in
the
Se
cur
ity
Pro
file
for
the
Sit
e,
but
the
clie
nt
fail
ed
to
ide
ntif
y
its
elf
dur
ing
the
ha
nd
sh
ake
ph

ase

- **12**

:
Cli
ent
aut
he
nti
cat
ion
is
im
po
sed
as
an
on
ym
ous
in
the
Se
cur
ity
Pro
file
for
the
Sit
e
but
the
clie
nt
ide
ntif
ied
its
elf
dur
ing
the
ha
nd
sh

ake
ph
ase

- **13**

:
Ser
ver
aut
he
nti
cat
ion
is
ma
nd
ato
ry
in
the
Se
cur
ity
Pro
file
for
the
Sit
e
but
the
ser
ver
did
not
ide
ntif
y
its
elf
dur
ing
the
ha
nd
sh

ake
ph
ase

- **14**

:
Ser
ver
aut
he
nti
cat
ion
im
po
sed
as
an
on
ym
ous
in
the
Se
cur
ity
Pro
file
for
the
Sit
e
but
the
ser
ver
ide
ntif
ied
its
elf
dur
ing
the
ha
nd

sh
ake
ph
ase

- **17**

:
Th
e
Se
cur
ity
Pro
file
for
the
Sit
e
has
an
inc
om
pat
ible
exi
t
sch
ed
uli
ng
val
ue
wit
h
the
Se
cur
ity
Pro
file
us
ed

- **18**

:
Th

e
Se
cur
ity
Pro
file
for
the
Sit
e
do
es
not
all
ow
the
cip
her
sui
te
us
ed

- **19**

:
Th
e
Se
cur
ity
Pro
file
for
the
Sit
e
do
es
not
all
ow
the
TL
S
ver
sio

-
- n
us
ed
 - **20**
:
Se
cti
on
"U
ser
Lo
cal
Cer
tifi
cat
es"
of
the
Se
cur
ity
Pro
file
for
the
Sit
e
do
es
not
co
nta
in
the
se
nt
cer
tifi
cat
e
(%
4)
 - **21**
-

:
Te
mp
lat
es
of
ne
w
Se
cur
ity
Pro
file
do
not
ma
tch
rec
eiv
ed
cer
tifi
cat
ion
ch
ain

- **30**

:
Th
e
co
m
mo
n
na
me
of
the
cer
tifi
cat
e
rec
eiv

ed
do
es
not
ma
tch
the
on
e
set
in
the
**Su
bj
ect
ce
rti
fic
at
e
ali
as**
fiel
d
of
the
Sit
e

- **31**
:
Th
e
co
m
mo
n
na
me
of
the
iss
uer
cer

tifi
cat
e
rec
eiv
ed
do
es
not
ma
tch
the
on
e
set
in
the
Is
su
er
ce
rti
fic
at
e
ali
as
fiel
d
of
the
Sit
e

- **32**
:
Th
e
dis
tin
gui
sh
ed
na
-

me
of
the
cer
tifi
cat
e
rec
eiv
ed
do
es
not
ma
tch
the
pat
ter
n
set
in
the
**Su
bj
ect
na
me
pa
tte
rn**
fiel
d
of
the
Sit
e

- **33**
:
Th
e
dis
tin
gui

	sh ed na me of the iss uer cer tifi cat e rec eiv ed do es not ma tch the pat ter n set in the Is su er na me pa tte rn fiel d of the Sit e
<i>Action</i>	Check the security profile defined in the <code>tls_sprof_in</code> attribute of the Site and the Security Profile used to establish the session.

371	SUP W DUPFILE Could not create transfer: transfer [%1] with duplicate file already exists	
<i>Type</i>	Error	
<i>Meaning</i>	A transfer could not be created as the VFD record exists.	
<i>Parameters</i>	1	Identifier of the new transfer
	2	Identifier of the old transfer
<i>Action</i>	Check the VFD configuration.	
372	372 SUP W DUPFILE Could not create transfer: transfer [%1] with duplicate file is in use	
<i>Type</i>	Error	
<i>Meaning</i>	A transfer could not be created as the VFD record is in use.	
<i>Parameters</i>	1	Identifier of the new transfer
	2	Identifier of the old transfer
<i>Action</i>	Check the VFD configuration.	
373	SUP E DUPFILE [%1] transfer update failed: transfer [%2] with duplicate file already exists	
<i>Type</i>	Error	
<i>Meaning</i>	A transfer could not be updated as the VFD record exists.	
<i>Parameters</i>	1	Identifier of the new transfer
	2	Identifier of the old transfer
<i>Action</i>	Check the VFD configuration.	
374	SUP E DUPFILE [%1] transfer update failed: transfer [%2] with duplicate file is in use	

Type	Error	
Meaning	A transfer could not be updated as a duplicate VFD record is in use.	
Parameters	1	Identifier of the new transfer
	2	Identifier of the old transfer
Action	Check the VFD configuration.	
375	SUP W DUPFILE [%1] transfer request canceled by transfer request [%2]	
Type	Warning	
Meaning	Supervisor canceled a transfer as duplicate record exists in VFD.	
Parameters	1	Identifier of the new transfer
	2	Identifier of the old transfer
Action	Check the VFD configuration.	
376	SUP E DUPFILE [%1] transfer request could not be canceled by transfer request [%2]	
Type	Error	
Meaning	Supervisor tried to cancel a transfer due to a VFD duplicate record but the transfer was locked.	
Parameters	1	Identifier of the new transfer
	2	Identifier of the old transfer
Action	Check the VFD configuration.	
377	SUP W XCOMSCR %1 [%3] transfer request canceled on max site connection retry	
Type	Warning	

Meaning	The transfer is canceled as the maximum number of retries on the outgoing site has been reached.	
Parameters	1	Site alias that has reached maximum number of retries
	2	Identifier of the cancelled transfer
Action	Check the site configuration.	
378	SUP W XRSINUSE [%1] Transfer IN USE on RESEND (refcount=%2)	
Type	Warning	
Meaning	Warning that the transfer is in use on a RESEND.	
Parameters	1	Transfer ID
	2	EDI reference count (technical message for diagnostic)
Action	Contact Axway Support.	
379	SUP W XRSLOCKED [%1] Transfer LOCKED on RESEND	
Type	Warning	
Meaning	Warning that the transfer is in locked on a RESEND.	
Parameters	1	Transfer ID
Action	Contact Axway Support.	
380	SUP W XRSUPDERR [%1] Transfer UPDATE FAILURE on RESEND: %2	
Type	Warning	
Meaning	Warning that the transfer update failed on a RESEND.	
Parameters	1	Transfer IS
	2	Text message
Action	Diagnose according to text message (parameter 2).	

381	SUP W XRSGETERR [%1] Transfer GET FAILURE on RESEND: %2	
Type	Warning	
Meaning	Warning that the transfer GET failed on a RESEND.	
Parameters	1	Transfer ID
	2	Text message
Action	Diagnose according to text message (parameter 2).	
382	SUP I XRSRSTOK [%1] Transfer successfully restarted for RESEND	
Type	Information (trace message)	
Meaning	The transfer successfully restarted on a RESEND.	
Parameters	1	Transfer ID
Action	None	
383	SUP I XRSCANOK [%1] Transfer successfully canceled for RESEND	
Type	Information (trace message)	
Meaning	The transfer successfully cancelled on a RESEND	
Parameters	1	Transfer ID
Action	None	
384	SUP W XTBINUSE [%1] Transfer IN USE on DATE TO BEGIN reached (refcount=%2)	
Type	Warning	
Meaning	Warning that the transfer is already in use on the <i>date to begin</i> trigger	

Parameters	1	Transfer ID
	2	Reference count (for diagnostic)
Action	None	

385	SUP W XTBLOCKED [%1] Transfer LOCKED on DATE TO BEGIN reached	
Type	Warning	
Meaning	Warning that the transfer is locked, when reaching the <i>date to begin</i> trigger	
Parameters	1	Transfer ID
Action	None	

386	SUP W XTBUDDERR [%1] Transfer UPDATE FAILURE on DATE TO BEGIN reached: %2	
Type	Warning	
Meaning	Warning that the transfer has had an <i>update failure</i> , before reaching the <i>date to begin</i> trigger	
Parameters	1	Transfer ID
	2	Text message indicating error context
Action	Action depends on text message content.	

387	SUP W XTGETERR [%1] Transfer GET FAILURE on DATE TO BEGIN reached: %2	
Type	Warning	
Meaning	Warning that the transfer has registered a <i>get failure</i> , when reaching the <i>date to begin</i> trigger	
Parameters	1	Transfer ID
	2	Text message indicating error context
Action	Action depends on text message content	

388	SUP E XTBGIVEUP [%1] Transfer NOT UPDATED on DATE TO BEGIN reached due to unrecoverable error	
<i>Type</i>	Error	
<i>Meaning</i>	The transfer has not been updated when reaching the <i>date to begin</i> trigger	
<i>Parameters</i>	1	Transfer ID
<i>Action</i>	Analyze the log messages that immediately precede this message to understand the failure context. This message is always preceded in the log by one of the SUP warning messages: 384, 385, 386, 387 .	
389	SUP I XTBDUPDOK [%1] Transfer state successfully changed on date to begin reached	
<i>Type</i>	Information	
<i>Meaning</i>	This message only occurs when the File Transfer log level is >3. The transfer state has been changed when reaching the <i>date to begin</i> trigger.	
<i>Parameters</i>	1	Transfer ID
<i>Action</i>	None	
394	SUP W XTEINUSE [%1] Transfer IN USE on DATE TO END reached (refcount=%2)	
<i>Type</i>	Warning	
<i>Meaning</i>	Warning that the transfer was in use when the <i>date to end</i> trigger was reached.	
<i>Parameters</i>	1	Transfer ID
	2	Reference counter number
<i>Action</i>	No user action. Gateway initiates an automatic retry after 30 seconds	

395	SUP W XTELOCKED [%1] Transfer LOCKED on DATE TO END reached	
Type	Warning	
Meaning	Warning that the transfer was locked when the <i>date to end</i> trigger was reached.	
Parameters	1	Transfer ID
Action	No user action. Gateway initiates an automatic retry after 30 seconds.	
396	SUP W XTEUPDERR [%1] Transfer UPDATE FAILURE on DATE TO END reached: %2	
Type	Warning	
Meaning	Warning that the transfer update failed when the <i>date to end</i> trigger was reached.	
Parameters	1	Transfer ID
	2	Text message describing failure context, for diagnosis
Action	Analyze failure context.	
397	SUP W XTEGETERR [%1] Transfer GET FAILURE on DATE TO END reached: %2	
Type	Warning	
Meaning	Warning that there was a transfer <i>get</i> failure when the <i>date to end</i> trigger was reached.	
Parameters	1	Transfer ID
	2	Text message describing failure context, for diagnosis
Action	Analyze failure context.	
398	SUP E XTEGIVEUP [%1] Transfer NOT CANCELLED on DATE TO END reached due to unrecoverable error	

<i>Type</i>	Error
<i>Meaning</i>	The transfer was not cancelled, due to an unrecoverable error, when the <i>date to end</i> trigger was reached.
<i>Parameters</i>	1 Transfer ID
<i>Action</i>	Analyze the log messages that immediately precede this message to understand the failure context. This message is always preceded in the log by one of the SUP warning messages: 394, 395, 396, 397 .

399	SUP I XTECANOK [%1] Transfer successfully cancelled on date to end reached
<i>Type</i>	Information (trace message)
<i>Meaning</i>	The transfer was successfully canceled when the <i>date to end</i> trigger was reached.
<i>Parameters</i>	1 Transfer ID
<i>Action</i>	None

400	SUP I XEND_CF %1(%2) [%3] Not routed, %4: %5
<i>Type</i>	Information
<i>Meaning</i>	Routing was not successful because a condition was not met.
<i>Parameters</i>	1 Remote agent name
	2 Connection identifier (number)
	3 Local transfer identifier
	4 Parameter whose condition was not met
	5 Value of the parameter
<i>Action</i>	Analyze why the routing was not successful.

401	SUP I XEND_CF %1(%2) [%3] Not routed, local_ident and remote_agent are not different	
Type	Information	
Meaning	Routing was not successful because the transfer destination Site (remote_agent) is not different from the Local Site (local_agent).	
Parameters	1	Remote agent name
	2	Connection identifier (number)
	3	Local transfer identifier
Action	None	
411	SUP I XEND_CF %1(%2) [%3] Not Routed, %4 : %5 is a local site	
Type	Information	
Meaning	The selected destination is a Local Site.	
Parameters	1	Remote agent name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Object and parameters: - Model remote_agent - Model org_alias - Model dest_alias - CGate route_remote_agent - VFD route_remote_agent - Site route_remote_agent - Destination
	5	Selected Site alias
Action	Analyze why routing was not successful.	

412	SUP I XEND_CF %1(%2) [%3] Not Routed, %4 : site %5 is not defined	
Type	Information	
Meaning	The selected destination is not defined.	
Parameters	1	Remote agent name
	2	Connection identifier (number)
	3	Local transfer identifier
	4	Object and parameters: • Model remote_agent • Model org_alias • Model dest_alias • CGate route_remote_agent • VFD route_remote_agent • Site route_remote_agent • Destination
	5	Selected Site alias
Action	Analyze why the routing was not successful.	

470	SUP W COUS User %1: max upload rate exceeded, aborting transfers	
Type	Warning	
Meaning	The given user exceeded upload rate quota. All current transfers are aborted and new upload transfers are not allowed for this user until the user's total upload use decreases.	
Parameters	1	User name
Action	An attack was detected according to the quotas given. Either increase the quota for this user, reset the Connected User, or, forbid access to this user.	

471	SUP W COUS User %1: max download rate exceeded, aborting transfers	
<i>Type</i>	Warning	
<i>Meaning</i>	The given user exceeded the download rate quota. All current transfers are aborted and new download transfers are not allowed for this user until the user's total download use decreases.	
<i>Parameters</i>	1	User name
<i>Action</i>	An attack was detected according to the quotas given. Either increase the quota for this user, reset the Connected User, or forbid access to this user.	
472	SUP W COUS User %1: max command rate exceeded, aborting connections	
<i>Type</i>	Warning	
<i>Meaning</i>	The given user exceeded the command sensitivity quota. All this user's connections are closed, and no new connections are allowed for this user until the user's total command use decreases.	
<i>Parameters</i>	1	User name
<i>Action</i>	An attack was detected according to the quotas given. Either increase the quota for this user, reset the connected user, or forbid access to this user.	
480	SUP W SUSSCPEN Pending connections at max level, transfer scheduling suspended	
<i>Type</i>	Warning	
<i>Meaning</i>	The number of pending connections has reached the maximum limit. Transfer scheduling is suspended.	
<i>Parameters</i>	None	
<i>Action</i>	None	
481	SUP I RESSCPEN Pending connections below max level, transfer scheduling resumed	

<i>Type</i>	Info
<i>Meaning</i>	The number of pending connections is now below the maximum limit. Transfer scheduling is resumed.
<i>Parameters</i>	None
<i>Action</i>	None

**520 AMT I AMTS_ROUT %1[%2] transfer routed to AMTRIX/XIB.
file=%3**

<i>Type</i>	Information
<i>Meaning</i>	Indicates that the transfer was routed to the Axway Integrator connector.
<i>Parameters</i>	1 Remote Site name
	2 Transfer ID
	3 File name
<i>Action</i>	None

521 AMT E AMTS_ROUT %1[%2] AMTRIX/XIB route failed. file=%3

<i>Type</i>	Error
<i>Meaning</i>	Error message indicating that the transfer intended for the Axway Integrator connector cannot be routed.
<i>Parameters</i>	1 Remote Site name
	2 Transfer ID
	3 File name
<i>Action</i>	None

**522 AMT E AMTS_API %1[%2] XFBCONN connector error. Verb=%3.
Reason=%4 [%5]**

<i>Type</i>	Error	
<i>Meaning</i>	Error message indicating that the transfer intended for the Axway Integrator connector cannot be routed.	
<i>Parameters</i>	1	Remote Site name
	2	Transfer ID
	3	XFBCONN API provoking the error
<i>Action</i>	None	

900 SUP E INTHP [%1] Unable to read transfer %2

<i>Type</i>	Error	
<i>Meaning</i>	Transfer not found in database	
<i>Parameters</i>	1	Transfer local identifier
	2	Error code (number)

901 SUP E INTHP [%1] Unable to write signature data %2

<i>Type</i>	Error	
<i>Meaning</i>	Can't attach signature information to the transfer	
<i>Parameters</i>	1	Transfer local identifier
	2	Error code (number)

903 SUP E INTHP [%1] Unable to write transfer %2

<i>Type</i>	Error	
<i>Meaning</i>	Can't update transfer	
<i>Parameters</i>	1	Transfer local identifier
	2	Error code (number)

904	SUP E INTHP Internal error (%1) %2	
<i>Type</i>	Error	
<i>Meaning</i>	Unexpected error	
<i>Parameters</i>	1	Error code (number)
	2	Error description
906	SUP W INTHP [%1] No payload, skip integrity processing	
<i>Type</i>	Warning	
<i>Meaning</i>	Transfer doesn't have a payload, no integrity operation is performed	
<i>Parameters</i>	1	Transfer local identifier
907	SUP I INTHP [%1] Signature %2 successfully	
<i>Type</i>	Information	
<i>Meaning</i>	Signature was either generated or verified successfully	
<i>Parameters</i>	1	integrity operation performed
908	SUP E INTHP [%1] Signature %2 failed	
<i>Type</i>	Error	
<i>Meaning</i>	Signature generation or verification failed	
<i>Parameters</i>	1	integrity operation performed

Axway Gateway: Messages and codes

SWIFTNet (SNET)

This topic lists SWIFTNet log messages.

001	SNET E SELOPERR %1(%2) [%3] select-open error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file selection or open operations. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description
<i>Action</i>	Check the parameters of the Transfer Request and of the Application used for the compatibility of the file characteristics with the file system attributes referenced by the transfer. Check the access security and path to make sure that Gateway can access the file. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
002	SNET E CREOPERR %1(%2) [%3] create-open error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file creation operations for incoming transmission. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description

<i>Action</i>	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is used. You must have enough space on your disk to receive the file. Refer to your platform documentation to understand the local file system error.	
003	SNET E CLODSERR %1(%2) [%3] close-deselect error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Error on the file closing or de-selection operations. The local file system error is returned in parameter %6.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description
<i>Action</i>	Refer to your platform documentation to understand the local file system error.	
004	SNET E READ_ERR %1(%2) [%3] read error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Read error encountered on the file to send. The local file system error is returned in parameter %6.	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description
Action	Check the parameters of the Application used and make sure that the file system attributes are consistent. Gateway must have the <i>read</i> privilege for the file to send. If a system dependent parameter is requested for file access, you must check that the request contains its description. Refer to your platform documentation to understand the local file system error.	
005	SNET E WRITEERR %1(%2) [%3] write error, file %4/%5: %6	
Type	Error	
Meaning	Write error encountered on the received file. The local file system error is returned in parameter %6.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description
Action	Check the parameters of the Application used by Gateway for the file reception, and make sure that the file system attributes are consistent. Gateway must have the <i>write</i> privilege in the directory defined (in the Application) for the creation of the files in reception. If a system dependent parameter is requested for file access, you must make sure that it is actually used. There must be enough disk space to receive the file. Refer to your platform documentation to understand the local file system error.	

006	SNET E CMPF_ERR %1(%2) [%3] compression error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	The local file could not be compressed.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description
<i>Action</i>	Gateway may not have enough free space to achieve the compression. Check the available disk space.	

007	SNET E UCMPFERR %1(%2) [%3] decompression error, file %4/%5: %6	
<i>Type</i>	Error	
<i>Meaning</i>	The local file could not be decompressed.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Physical file directory
	5	Physical file name
	6	Error description
<i>Action</i>	Gateway may not have enough free space to achieve the decompression. Check the available disk space. The file may not have been compressed properly.	

008	SNET W QUEUEACQ %1(%2) queue %3 acquire request rejected		
Type	Warning		
Meaning	Warning message indicating that the request for acquiring a queue has been rejected because there is another active output session on it.		
Parameters	1	Access Point name	
	2	Message partner	
	3	SnF queue name	
Action	Several possible actions: • Identify messaging interface already using the queue. • Use another queue.		
010	SNET E ABORTIND %1(%2) ABORT received: %3 %4		
Type	Error		
Meaning	Abort received on SWIFTNet session.		
Parameters	1	Access Point name	
	2	Message partner	
	3	Error code	
	4	Error code number (reason for the abort)	
Action	If the problem is a recurring one, check the Remote and Local Trading Partner definitions. The transfer may have been canceled locally or remotely.		
011	SNET E ABORTREQ %1(%2) ABORT sent: %3 %4		
Type	Error		
Meaning	Abort sent on SWIFTNet session.		

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Error code
	4	Error code number (reason for the abort)
<i>Action</i>	If the problem is a recurring one, check the Remote and Local Trading Partner definitions. The transfer may have been aborted locally.	

015	SNET I SITEACTN %1(%2) site has been %3	
<i>Type</i>	Information	
<i>Meaning</i>	AA SWIFTNet site state has changed.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	New state:
		• deleted • disconnected • initialized • started • stopped
<i>Action</i>	None	

016	SNET E SITEACTNE %1(%2) site has been %3. Reason: %4	
<i>Type</i>	Error	
<i>Meaning</i>	A SWIFTNet site state has changed due to a configuration error.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	New state: • disabled
	4	Error reason
<i>Action</i>	Contact the Gateway Administrator.	

021	SNET I XFERRCV1 %1(%2) [%3] begin receiving from %4,%5 file: %6 %7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the start of reception.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
<i>Action</i>	None	

022	SNET I XFERSND1 %1(%2) [%3] begin sending to %4,%5 file: %6 %7	

<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the start of transmission.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
<i>Action</i>	None	
023	SNET I XENDRCV1 %1(%2) [%3] end receiving from %4,%5 file: %6 %7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the end of reception.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
<i>Action</i>	None	
024	SNET I XENDSND1 %1(%2) [%3] end sending to %4,%5 file: %6 %7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the end of transmission.	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
Action	None	
025	SNET E XERRRCV1 %1(%2) [%3] error receiving from %4,%5 file: %6 %7	
Type	Error	
Meaning	Reception error	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
Action	Check the messages indicating the cause of the problem. If no solution is found, contact Axway Support.	
026	SNET E XERRSND1 %1(%2) [%3] error sending to %4,%5 file: %6 %7	
Type	Error	
Meaning	Transmission error	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
Action	Check the messages indicating the cause of the problem. If no solution is found, contact Axway Support.	
027	SNET I MESGSEND %1(%2) [%3] sending to %4,%5 message: %6 %7	
Type	Information	
Meaning	Information message indicating the transmission of a message.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Copied or distributed transfer indicator. Possible values: • "(copied)" • "(distributed)" • "" This parameter is empty for normal transfers, which are not copied or distributed.
	6	Physical file directory
	7	Physical file name
<i>Action</i>	None	
028	SNET I MESGRECV %1(%2) [%3] receiving from %4,%5 message: %6 %7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the reception of a message.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Retrieved message indicator. Copied or distributed transfer indicator. Possible values: • "(copied)" • "(retrieved, sent)(copied)" or "(retrieved, received)(copied)" • "(distributed)" • "(retrieved, sent)(distributed)" or "(retrieved, received)(distributed)" • "" • "(retrieved, sent)" or "(retrieved, received)" This parameter is empty for normal transfers, which are not retrieved, copied or distributed.
	6	Physical file directory
	7	Physical file name
<i>Action</i>	None	

029	SNET E MESGEXCH %1(%2) [%3] error during message exchange with %4	
<i>Type</i>	Error	
<i>Meaning</i>	Error during message exchange. The recipient may have received the message.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
<i>Action</i>	Check the messages indicating the cause of the problem. If no solution is found, contact Axway Support.	

030	SNET I MESGEXCH %1(%2) [%3] message exchange with %4 successful	
Type	Information	
Meaning	Information message indicating the success of the message exchange phase.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
Action	None	
031	SNET I QUEUEACQ %1(%2) acquired queue %3 session %4	
Type	Information	
Meaning	Information message indicating the acquisition of an SnF queue.	
Parameters	1	Access Point name
	2	Message partner
	3	Name of the SnF queue
	4	SnF session number
Action	None	
032	SNET I QUEUEREL %1(%2) released session %3	
Type	Information	
Meaning	Information message indicating the release of a queue identified by its session number.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	SnF session number
<i>Action</i>	None	

033	SNET I QUEUEFET %1(%2) [%3] fetching file %4 session %5	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an attempt to fetch a file from an SnF queue.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Local physical file name
	5	SnF session number
<i>Action</i>	None	

034	SNET I QUEUEACK %1(%2) [%3] acknowledge slot session %4	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the acknowledgment of a previously-fetched file.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	SnF session number
<i>Action</i>	None	

035	SNET E QUEUEACQ %1(%2) error acquiring queue %3	
Type	Error	
Meaning	An error occurred during the acquisition of a queue.	
Parameters	1	Access Point name
	2	Message partner
	3	Name of the SnF queue
Action	Check that the queue name is correct and the SWIFTNet user has the right to handle the queue.	
036	SNET E QUEUEREL %1(%2) error releasing session %3	
Type	Error	
Meaning	An error occurred during the release of a queue identified by its session number.	
Parameters	1	Access Point name
	2	Message partner
	3	SnF session number
Action	The session may already have been released.	
037	SNET E QUEUEFET %1(%2) [%3] error fetching file %4 session %5	
Type	Error	
Meaning	Error during the attempt to fetch a file from an SnF queue.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Local physical file name
	5	SnF session number
<i>Action</i>	The file may already have been fetched by another application.	

038	SNET E QUEUEACK %1(%2) [%3] slot acknowledgment error session %4	
<i>Type</i>	Error	
<i>Meaning</i>	The fetched file cannot be acknowledged.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	SnF session number
<i>Action</i>	None (the transfer will be restarted automatically).	

039	SNET I QUEUESTS %1(%2) SnF session %3 returned status: %4	
<i>Type</i>	Information	
<i>Meaning</i>	Message information indicating the current status of an SnF session received from SWIFT.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	SnF session number
	4	Session status
<i>Action</i>	None	

040	SNET E QUEUESTS %1(%2) SnF session %3 returned error status: %4	
Type	Error	
Meaning	Error message indicating the current error status of an SnF session received from SWIFT.	
Parameters	1	Access Point name
	2	Message partner
	3	SnF session number
	4	Session error status
Action	None	
041	SNET E APINIERR %1(%2) error during access-point initialization: %3	
Type	Error	
Meaning	An Access Point could not be initialized.	
Parameters	1	Access Point name
	2	Message partner
	3	Error description
Action	Check the Remote Site information given in the Access Point name. Verify the RA and SAG configuration.	
042	SNET E APCONERR %1(%2) access-point configuration error: %3	
Type	Error	
Meaning	Problem with the Access Point configuration.	

Parameters	1	Access Point name
	2	Message partner
	3	Error description
Action	Check the Remote Site information given in the Access Point name. Verify the RA and SAG configuration.	
043	SNET W APCONWAR %1(%2) access-point configuration warning: %3	
Type	Warning	
Meaning	Problem encountered when initializing the Access Point.	
Parameters	1	Access Point name
	2	Message partner
	3	Error description
Action	Check the Remote Site information given in the Access Point name. Verify the RA and SAG configuration.	
044	SNET I SYSMESS %1(%2) %3 system message identified at check-in (looking for Trading Partner)	
Type	Information	
Meaning	Incoming system message.	
Parameters	1	Access Point name
	2	Message partner
	3	Type of message
Action	None	
045	SNET W QUEUESTS Swift received %1 (%2) for resource name %3 generated on SNLId %4 at time %5	

Type	Warning	
Meaning	Warning message received from SNL indicating that an unsolicited event occurred. Currently there is only a single event, generated when SNL detects that the output session was closed by SWIFT.	
Parameters	1	Event type
	2	Event description
	3	Resource name
	4	SNL ID on which the event occurred
	5	Time when the event occurred
Action	None	
050	SNET I SNFDNREQ %1(%2) [%3] received SnF delivery notification from %4 with status %5	
Type	Information	
Meaning	Information message indicating the receipt of an SnF delivery notification.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Source of delivery notification (Third Party, Counter Party)
	5	Status (accepted, rejected, failed)
Action	None	
051	SNET I SNFDNUPD %1(%2) [%3] updating transfer	
Type	Information	
Meaning	Information message indicating the update of a transfer.	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	None	
052	SNET E SNFDNERR %1(%2) no transfer match for SnF reference %3	
Type	Error	
Meaning	Received SnF delivery notification does not match any existing transfer in the Mailbox. The transfer has been deleted or the Mailbox has been purged.	
Parameters	1	Access Point name
	2	Message partner
	3	SnF reference number
Action	None	
053	SNET W SNFDNDPL %1(%2) [%3] transfer already notified	
Type	Warning	
Meaning	Warning message indicating that a transfer notification has already been received (technical duplicate).	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	None	
055	SNET E XSRVRJCT %1(%2) [%3] Server rejected the transfer %4	
Type	Error	

<i>Meaning</i>	Incoming transfer rejected by Gateway. Possible reasons: • An unknown partner has sent some unexpected traffic. • The configuration might not be correct.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Message ID
<i>Action</i>	If this is a configuration issue, review and modify the Trading Partner configuration.	
056	SNET E XENDRERR %1(%2) [%3] end reason: %4, end diag: %5, end error: %6	
<i>Type</i>	Error	
<i>Meaning</i>	Generic error message for a transfer that failed.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Error code number (reason)
	5	Error code number (diagnostic)
	6	Error message
<i>Action</i>	Check the messages indicating the cause of the problem. If no solution is found, contact Axway Support.	
060	SNET I RCPTSEND %1(%2) [%3] begin sending to %4, %5 receipt for transfer number [%6]	
<i>Type</i>	Information	

Meaning	An ack delivery notification request is being sent.	
Parameters	1	Access Point name
	2	Message partner
	3	Receipt transfer number
	4	Remote Trading Partner
	5	Receipt type (positive/negative)
	6	Original transfer that is being acknowledged
Action	None	
061	SNET I RCPTSND %1(%2) [%3] end sending to %4, %5 receipt for transfer number [%6]	
Type	Information	
Meaning	An ack delivery notification request has been sent.	
Parameters	1	Access Point name
	2	Message partner
	3	Receipt transfer number
	4	Remote Trading Partner
	5	Receipt type (positive/negative)
	6	Original transfer that is being acknowledged
Action	None	
062	SNET E RCPTSND %1(%2) [%3] error sending to %4, %5 receipt for transfer number [%6]	
Type	Error	
Meaning	An error occurred during the exchange of the ack delivery notification request.	

Parameters	1	Access Point name
	2	Message partner
	3	Receipt transfer number
	4	Remote Trading Partner
	5	Receipt type (positive/negative)
	6	Original transfer that is being acknowledged
Action	The receipt transfer is automatically retried. If the error persists check traces for more details.	
063	SNET I RCPTRECV %1(%2) begin receiving from %3, %4 receipt for transfer reference %5	
Type	Information	
Meaning	An ack delivery notification request has been received and accepted into the Mailbox.	
Parameters	1	Access Point name
	2	Message partner
	3	Receipt transfer number
	4	Remote Trading Partner
	5	SNL transfer reference of the original transfer
Action	None	
064	SNET I RCPTRCVS %1(%2)%3%4%5 end receiving from %6, %7 receipt for transfer number [%8]	
Type	Information	
Meaning	An ack delivery notification request has been received.	

Parameters	1	Access Point name
	2	Message partner
	3	Receipt transfer number, if Trading Partner found
	4	Receipt transfer number, if Trading Partner found
	5	Receipt transfer number, if Trading Partner found
	6	Remote Trading Partner
	7	Receipt type (positive/negative)
	8	Original transfer that has been acknowledged
Action	None	
065 SNET E RCPTRCVE %1(%2) error occurred while receiving %3 receipt from %4		
Type	Error	
Meaning	An error has occurred during the ack delivery notification exchange.	
Parameters	1	Access Point name
	2	Message partner
	3	Receipt type (positive/negative)
	4	Remote Trading Partner
Action	None	
069 SNET I RCPTACPT %1(%2)%3%4%5 %6 receipt from %7 has been %8%9		
Type	Information	
Meaning	An acknowledgment delivery notification protocol request has been received. Check to see if is accepted in the Mailbox.	

Parameters	1	Access Point name
	2	Message partner
	3	Receipt transfer number, if Trading Partner found
	4	Receipt transfer number, if Trading Partner found
	5	Receipt transfer number, if Trading Partner found
	6	Receipt type (positive/negative)
	7	Remote Trading partner
	8	Status (accepted/rejected/rejected as duplicated)
	9	Reason (present if Status is rejected)
Action	None	
070	SNET E ASPCHKERR %1(%2) [%3] ASP Check to validate outgoing request parameters and to get RMA feature failed. Reason: %4	
Type	Error	
Meaning	Validation of outgoing request parameters against service profile failed or checks if RMA filtering required failed. The reason for the error should be in the log.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Message containing failure reason
Action	If there is an error related to incorrect request parameters usage, correct the problem and try again to initiate the transfer. For other failure reason, contact Axway support.	
071	SNET I ASPCHKDEF %1(%2) [%3] ASP Check to validate outgoing request parameters and to get RMA feature successful, but the default profile was used %4	

Type	Information	
Meaning	Information message indicating outgoing transfer parameters were successfully checked against the default service profile, including if RMA filtering is required by ASP for current transfer.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Service profile is requesting RMA filtering or not. Possible values: • RMA check required • RMA check required (in trial period) • RMA check not required This parameter is empty if RMA check is not enabled from SWIFTNet configuration.
	Action	Because the default service profile was used, you must ensure Gateway is using an updated ASP package. Meanwhile, it is possible SWIFT may have added the profile for the requested service to a more recent ASP package.

072	SNET I ASPLDSTS Swift ASP package %1 done. Files: %2, Date: %3, Start: %4, Stop: %5	
Type	Information	
Meaning	Information message indicating ASP package was loaded successfully.	
Parameters	1	Operation applied to ASP package: load or reload
	2	Number of service profiles loaded or reloaded
	3	Date when load or reload operation occurred
	4	Time when the load or reload operation started
	5	Time when the load or reload operation ended
Action	None	

073	SNET W ASPLDSTS Swift ASP package %1 done with warning %2. Files: %3, Date: %4, Start: %5, Stop: %6	
Type	Warning	
Meaning	ASP package was loaded or reloaded, but a warning was raised during the operation.	
Parameters	1	Operation applied to ASP package: load or reload
	2	Warning description
	3	Number of service profiles loaded or reloaded
	4	Date when load or reload operation occurred
	5	Time when the load or reload operation started
	6	Time when the load or reload operation ended
Action	Try to find the cause of the error and correct it.	
074	SNET E ASPLDSTS Swift ASP package %1 failed. Reason: %2, ErrNo: %3, Date: %4, Start: %5, Stop: %6, Previous active load: %7	
Type	Error	
Meaning	ASP package could not be loaded or reloaded because an error occurred.	
Parameters	1	Operation applied to ASP package: load or reload
	2	Failure reason
	3	Error code number
	4	Date when load or reload operation occurred
	5	Time when the load or reload operation started
	6	Time when the load or reload operation ended
	7	Date and time of the last active load
Action	Try to find the cause of the error and correct it.	

075	SNET E ASPLDINI Swift ASP package %1 initialization failed. Previous active load: %2	
<i>Type</i>	Error	
<i>Meaning</i>	The initialization operation for loading or reloading the ASP package failed.	
<i>Parameters</i>	1	Operation applied to ASP package: load or reload
	2	Date and time of the last active load
<i>Action</i>	None	
080	SNET I OPOUTCH %1(%2) opened output channel %3 for queue %4 session %5	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the success of the opening of an output channel.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Queue name
	5	SnF session number
<i>Action</i>	None	
081	SNET W OPOUTCHR %1(%2) output channel %3 opening for queue %4 request rejected	
<i>Type</i>	Warning	
<i>Meaning</i>	Warning message indicated that the request for opening an output channel has been rejected because the queue is in exclusive mode and there is an active output session on it.	

Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Queue name
Action	Possible actions: • Identify messaging interface already using the queue • Change queue sharing mode • Use another queue	
082	SNET E OPOUTCHE %1(%2) error opening output channel %3 for queue %4	
Type	Error	
Meaning	An error occurred during the opening of an output channel.	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Queue name
Action	Check that the channel and queue name are correct and the SWIFTNet user has the right to handle them.	
083	SNET I CLOUTCH %1(%2) released session %3	
Type	Information	
Meaning	Information message indicating the releasing of a session.	
Parameters	1	Access Point name
	2	Message partner
	3	SnF session number
Action	None	

084	SNET E CLOUTCHE %1(%2) error releasing session %3	
Type	Error	
Meaning	Error message indicating that an error occurred while releasing session.	
Parameters	1	Access Point name
	2	Message partner
	3	SnF session number
Action	None	
085	SNET E XERRSND1 %1(%2) [%3] SnFInputTime could not be saved (not received or internal error occurred)	
Type	Error	
Meaning	SnFInputTime was not received from SWIFT or an internal error occurred while saving it.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	Contact Axway Support.	
086	SNET W OVDUETM %1(%2) [%3] Transfer is overdue, due date was: %4	
Type	Warning	
Meaning	Overdue transfer warning message.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Transfer ID
	4	Due time
<i>Action</i>	No action required. SWIFT detected that a file or message was not delivered before the specified due time (OverdueTime specified by the sender). SWIFT sends this warning message to the sender to inform him that the receiver has not yet received the file or message. This warning message provides no information as to whether the transfer will or will not be made.	
087	SNET I XFAUTH %1(%2) [%3] Transfer is authorised by a third party	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating an authorization of the transfer by a third party.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
<i>Action</i>	None	
088	SNET W XFAUTH %1(%2) [%3] Transfer is refused by a third party	
<i>Type</i>	Warning	
<i>Meaning</i>	Warning message indicating a refusal of the transfer by a third party.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)

Action	None	
089	SNET E SMDNERR %1(%2) no transfer match for System Message reference %3	
Type	Error	
Meaning	No transfer found in Mailbox for the incoming System Message. Transfer referenced by the system message could have been deleted or broken routing rules inside the SAG.	
Parameters	1	Access Point name
	2	Message partner
	3	Original SNL reference
Action	Verify the SAG routing configuration.	
091	SNET I OPINCH %1(%2) opening input channel %3, window=%4, force=%5	
Type	Information	
Meaning	Information message indicating the opening of an input channel.	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Input session suggested window size
	5	Force opening of input channel
Action	None	
092	SNET I OPINCHO %1(%2) input channel %3 opened, window=%4, nextInputSeq=%5	
Type	Information	

<i>Meaning</i>	Information message indicating the success of the opening of an input channel	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Input session window size
	5	Input session first available sequence number
<i>Action</i>	None	

093	SNET E OPINCHE %1(%2) open-error for input channel %3	
<i>Type</i>	Error	
<i>Meaning</i>	Error message indicating a failure while opening an input channel	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Input channel name
<i>Action</i>	Check that the input channel name already exists. If the channel is opened without the force parameter, it might be already opened or locked.	

094	SNET I CLINCH %1(%2) closing input channel %3, token=%4	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the closure of an input channel.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Input session token
<i>Action</i>	None	

095	SNET I CLINCHO %1(%2) input channel %3 closed, token=%4	
Type	Information	
Meaning	Information message indicating the success of the closure of an input channel	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Input session token
Action	None	
096	SNET E CLINCHE %1(%2) close error for input channel %3, token=%4	
Type	Error	
Meaning	Error message indicating a failure while closing an input channel	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Input session token
Action	Resolve gaps if any. It might also already have been closed	
097	SNET I CRINCHRQ %1(%2) creating input channel %3	
Type	Information	
Meaning	Input channel was scheduled for creation.	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
Action	None	

098	SNET I CRINCHCF %1(%2) input channel %3 created	
Type	Information	
Meaning	An input channel was created.	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
Action	None	
099	SNET E CRINCHER %1(%2) error creating input channel %3: %4	
Type	Error	
Meaning	Creation of an input channel failed.	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Error message
Action	Check whether the channel name is valid and complies with SWIFT specifications. Also, the input channel might already exist.	
100	SNET E SWERROR %1(%2) SwGbl:Status Severity=%3, Code=%4, Text=%5	
Type	Error	
Meaning	Display content of SwGbl:Status element received from SWIFT, indicating failure.	

Parameters	1	Access Point name
	2	Message partner
	3	Severity of the error
	4	Error code
	5	Error description
Action	Check the messages indicating the cause of the problem. If no solution is found, contact Axway Support.	
101	SNET I TRACE %1	
Type	Information	
Meaning	Trace message.	
Parameters	1	Trace message
Action	Use this message to trace a SWIFTNet transfer.	
102	SNET E TRACE %1	
Type	Error	
Meaning	Trace message.	
Parameters	1	Trace message
Action	Use this message to trace a SWIFTNet transfer.	
103	SNET W TRACE %1	
Type	Warning	
Meaning	Trace message.	
Parameters	1	Trace message
Action	Use this message to trace a SWIFTNet transfer.	

104	SNET E TRACEX %1(%2) [%3] %4	
Type	Error	
Meaning	Trace message	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Trace message
Action	Use this message to trace a SWIFTNet transfer.	
110	SNET I SYSRECMS System recovery detection started manually by user	
Type	Information	
Meaning	Information message indicating that the user has started system recovery detection manually.	
Action	None	
111	SNET I SYSRECMS System recovery detection started automatically	
Type	Information	
Meaning	Information message indicating that system recovery detection started automatically. This happens every time Gateway opens an output session, after a restart of Gateway, or after a potential SWIFT crash.	
Action	None	
112	SNET W SYSRECIP System recovery detection already in progress, new manual request ignored	
Type	Warning	

<i>Meaning</i>	Warning message indicating that system recovery detection is already in progress.
<i>Action</i>	None
113	SNET W SYSRECPC Possible system recovery detected, inconclusive resolution due to insufficient data
<i>Type</i>	Warning
<i>Meaning</i>	Gateway stores a list of previous SWIFT crashes to help detect a new one. If Gateway is started for the first time, or was offline for a considerable amount of time (weeks, maybe months), it would have no data or the data would be obsolete. This message does not mean there was a SWIFT crash but rather that the detection was inconclusive.
<i>Action</i>	None
114	SNET E SYSRECAG System recovery detection aborted: no active end-points (sites) found to schedule detection
<i>Type</i>	Error
<i>Meaning</i>	System recovery detection could not be scheduled because no active end-points (Sites) have been found.
<i>Action</i>	Make sure you have at least one active end-point (Site) to schedule system recovery detection.
115	SNET E SYSRECAS System recovery detection aborted: failed to get system recovery information from Swift
<i>Type</i>	Error
<i>Meaning</i>	An error occurred when Gateway asked SWIFT for previous crash timestamps (last replication time). This might be the result of a problem with SWIFT, SAG, SNL, network etc.
<i>Action</i>	None

116	SNET I SYSRECSN System recovery detection ended: no action required (no transfer to recover)	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating that system recovery detection ended and no action is required (no crash occurred).	
<i>Action</i>	None	
117	SNET W SYSRECFS System recovery confirmed, recovery procedure initiated for Store-and-Forward transfers after %1	
<i>Type</i>	Warning	
<i>Meaning</i>	Warning message indicating that a system recovery procedure has been initiated for Store-and-Forward transfers that were sent after the last replication timestamp.	
<i>Parameters</i>	1	Last replication time
<i>Action</i>	None	
118	SNET E SYSRECAF System recovery procedure aborted: no active end-points (sites) found to schedule transfer flagging	
<i>Type</i>	Error	
<i>Meaning</i>	System recovery procedure was aborted because no active end-points (Sites) were found to schedule transfer flagging (identify transfers that might have been lost in the crash).	
<i>Action</i>	Make sure you have at least one active end-point (Site) to schedule transfer flagging during a system recovery procedure.	
119	SNET E SYSRECEF System recovery procedure aborted: internal error or corrupted data files	
<i>Type</i>	Error	
<i>Meaning</i>	System recovery procedure was aborted because of an internal error or corrupted data files.	

<i>Action</i>	Contact Axway Support.	
120	SNET I SYSRECXT System recovery procedure ended: %1 transfers were marked by system recovery detection	
<i>Type</i>	Information	
<i>Meaning</i>	Message information indicating the ending of the system recovery procedure and the number of transfers marked by the system recovery detection.	
<i>Parameters</i>	1	Number of transfers
<i>Action</i>	None	
121	SNET I MESGSNDS %1(%2) [%3] sending to %4,%5 message: %6 %7 with sequence number %8	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the transmission of a message through an input channel.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Alias of the remote partner
	5	Remote Trading Partner
	6	Physical file directory
	7	Physical file name
	8	Input session sequence number
<i>Action</i>	None	
122	SNET W MESGEXGP %1(%2) [%3] message exchange with %4 successful but located after a gap	

Type	Warning		
Meaning	Warning message indicating the success of the message exchange phase although the message is located after a gap.		
Parameters	1	Access Point name	
	2	Message partner	
	3	Local request identifier (number)	
	4	Alias of the remote partner	
Action	This message can be displayed when several messages are handled simultaneously. The gap will be solved by itself. In the case that a transfer has not been accepted by the input channel, the gap will be solved when the transfer is canceled or when the Site or the product is restarted.		
123	SNET I OPINCH %1(%2) resolving gap for input channel %3, sequence=%4		
Type	Information		
Meaning	Information message indicating the resolution of a gap		
Parameters	1	Access Point name	
	2	Message partner	
	3	Input channel name	
	4	Input session sequence number	
Action	None		
124	SNET I OPINCHO %1(%2) resolve gap succeeds for input channel %3, sequence=(%4,%5)		
Type	Information		
Meaning	Information message indicating the success of the resolution of a gap		

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Input channel name
	4	First input session sequence number
	5	Last input session sequence number
<i>Action</i>	None	

125	SNET E OPINCHE %1(%2) resolve gap fails for input channel %3, sequence=%4	
<i>Type</i>	Error	
<i>Meaning</i>	Error message indicating the failure of the resolution of a gap	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Input session sequence number
<i>Action</i>	The input channel might be closed or the gap might have been already resolved.	

126	SNET W PDTRDUP %1(%2) [%3] Duplicated incoming request, with message id "%4", ignored	
<i>Type</i>	Warning	
<i>Meaning</i>	Warning message indicating that an incoming request which was detected as being duplicated was ignored because the configuration option Accept possible duplicate transfers was disabled.	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Message ID
Action	If you want the incoming requests detected as duplicated to be accepted, enable the configuration option Accept possible duplicate transfers .	
127	SNET W PDTRACC %1(%2) [%3] Duplicated incoming request, with message id "%4", accepted. The configuration option "accept_possible_duplicate" is set	
Type	Warning	
Meaning	Warning message indicating that an incoming request which was detected as being duplicated was accepted because the configuration option Accept possible duplicate transfers was enabled.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Message ID
Action	If you want the incoming requests detected as duplicated to be ignored, disable the configuration option Accept possible duplicate transfers .	
128	SNET I PDTRNFE %1(%2) [%3] Duplicated SnF request. No fetch, acknowledging slot session	
Type	Information	
Meaning	Information message indicating that the incoming SWIFT SnF request is duplicated (redelivered message). Therefore no file fetching operation is performed, but the acknowledgment is sent.	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	None	
129	SNET W PDTREID %1(%2) [%3] Incoming request with empty message id accepted, sender considers it might be a duplicate, but message id is unknown	
Type	Warning	
Meaning	Warning message indicating that an incoming request with empty message ID was received and accepted. However, it was considered as being a possible duplicate message on the <i>sender</i> side, which does not have the message ID information.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	Manually check if the incoming transfer is duplicated (file size, timestamp).	
130	SNET W SYSMESS %1(%2) [%3] Transfer authorisation status was already updated	
Type	Warning	
Meaning	Warning message indicating that a transfer authorization status was already updated after receiving a (previous) refusal system notification or authorization system notification.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	None	

131	SNET W TECHDUP %1(%2) [%3] Duplicated incoming request. Is a Swift redelivered message (technical duplicate), ignored	
Type	Warning	
Meaning	Warning message indicating that an incoming SnF request was detected as duplicated but ignored by Gateway because the message was redelivered by SWIFT.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	None	
132	SNET I ASPCHKOK %1(%2) [%3] ASP Check to validate outgoing request parameters and to get RMA feature successful %4	
Type	Information	
Meaning	Information message indicating outgoing transfer parameters were checked successfully against the service profile, including if RMA filtering is required by ASP for current transfer.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Service profile is requesting RMA filtering or not Possible values: • RMA check required • RMA check required (in trial period) • RMA check not required This parameter is empty if RMA check is not enabled from SWIFTNet configuration.
Action	None	

133	SNET I ASPRMAOK %1(%2) ASP Check to get RMA feature for incoming request with message id %3 successful %4	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating that the ASP check that queries if RMA filtering is required by ASP for current transfer was successful.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Incoming request message ID
	4	Service profile is requesting RMA filtering or not. Possible values: • RMA check required • RMA check required (in trial period) • RMA check not required
<i>Action</i>	None	
134	SNET I ASPRMADEF %1(%2) ASP Check to get RMA feature for incoming request with message id %3 successful, but the default profile was used %4	
<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating that the ASP check that queries if RMA filtering is required by ASP was successful, but the default profile was used to perform the check.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Incoming request message ID
	4	Service profile is requesting RMA filtering or not. Possible values: • RMA check required • RMA check required (in trial period) • RMA check not required

Action	Because the default service profile was used, ensure Gateway is using an updated ASP package. Meanwhile, SWIFT may have added the profile for the service in the latest ASP package.		
135	SNET E ASPRMAERR %1(%2) ASP Check to get RMA feature for incoming request with message id %3 failed. Reason: %4		
Type	Error		
Meaning	Error occurred while checking if the service profile requires RMA filtering for incoming request.		
Parameters	1	Access Point name	
	2	Message partner	
	3	Incoming request message ID	
	4	Message containing failure reason	
Action	Check the failure reason and try to fix the problem. If not possible, contact Axway support.		
136	SNET W ASPNRRERR %1(%2) [%3] ASP Check for non repudiation parameter value used in response failed. Reason: %4		
Type	Warning		
Meaning	Warning message indicating validation of the non repudiation value used in response failed, followed by failure reason.		
Parameters	1	Access Point name	
	2	Message partner	
	3	Local request identifier (number)	
	4	Message containing failure reason	
Action	If the error is related to incorrect non repudiation value, contact Axway support. Gateway is using in response the non repudiation value received in the incoming request, and this normally conforms to service profile.		

137	SNET I ASPNRROK %1(%2) [%3] ASP Check for non repudiation parameter value used in response successful	
Type	Information	
Meaning	Information message indicating the value of non repudiation parameter used in response if conforming to service profile	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	None	
138	SNET I ASPNRRDEF %1(%2) [%3] ASP Check for non repudiation parameter value used in response successful, but the default profile was used	
Type	Information	
Meaning	Information message indicating the value of non repudiation parameter used in response if conforming to default service profile.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	Because the default service profile was used, ensure Gateway is using an updated ASP package. Meanwhile, SWIFT may have added the profile for the service in the latest ASP package.	
140	SNET I ASPDSBLD Swift ASP & RMA disabled in global configuration	
Type	Information	
Meaning	The ASP/RMA check is disabled.	
Action	If you want to enable ASP checking with or without RMA checking, update the SWIFTNet configuration.	

141	SNET I SNFDNRR %1(%2) [%3] received SnF delivery notification from %4 with status %5. Reason: %6	
<i>Type</i>	Information	
<i>Meaning</i>	A Store-and-Forward delivery notification was received.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Remote partner type: • Counter-party • Third-party • Recipient
	5	Delivery notification status
	6	Delivery notification description
<i>Action</i>	None	
142	SNET I SUDEVTRREP %1(%2) [%3] received Undelivered Traffic Report (xsys.005.001.01) with SnFReference %4. Page number: %5; is last page: %6; execution status: %7	
<i>Type</i>	Information	
<i>Meaning</i>	Information message regarding the receipt of an Undelivered Traffic Report system message (xsys.005.001.01)	

Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	The Store-and-Forward reference of the Undelivered Traffic Request message which generated the Report
	5	Page number
	6	True if this is the last page of the report
	7	Success or Failure
Action	None	
143	SNET W UUDEVTRP %1(%2) [%3] received Unsolicited Undelivered Traffic Report (xsys.005.002.01) generated after %4. ReportOption: %5; Page number: %6; is last page: %7; execution status: %8	
Type	Information	
Meaning	Information message regarding the receipt of an Undelivered Traffic Report system message (xsys.005.001.01)	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	"Cold Start" if ReportOption is "CS", "unknown event" otherwise
	5	Report Option ("CS")
	6	Page number
	7	True if this is the last page of the report
	8	Success or Failure
Action	None	

150	SNET E RMACFGE RMA userexit configuration error: %1	
Type	Error	
Meaning	Error occurred while loading the RMA user exit configuration file.	
Parameters	1	Error description
Action	Make sure the full path and file name of the RMA user exit configuration file is specified, the file exists and the content is valid.	
151	SNET E RMALFAIL RMA userexit library load failed: %1	
Type	Error	
Meaning	Error occurred while loading the RMA user exit library.	
Parameters	1	Error description
Action	Make sure the full path and file name of the RMA user exit library file is specified, the library exists and is the correct version.	
152	SNET E RMAIFAIL RMA userexit implementation init failed: %1	
Type	Error	
Meaning	Error occurred while initializing the implementation of RMA user exit library.	
Parameters	1	Error description
Action	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	
153	SNET E RMAUFAIL RMA userexit library unload failed: %1	
Type	Error	
Meaning	Error occurred while unloading the RMA user exit library.	
Parameters	1	Error description
Action	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	

154	SNET I RMACKOKS %1(%2) Check RMA authorization-to-send for incoming request with message id %3: valid authorization found	
<i>Type</i>	Information	
<i>Meaning</i>	RMA authorization check was performed by RMA user exit for the incoming request. Valid authorization was found to send.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Incoming request message ID
<i>Action</i>	None	
155	SNET I RMACKOKR %1(%2) Check RMA authorization-to-receive for incoming request with message id %3: valid authorization found	
<i>Type</i>	Information	
<i>Meaning</i>	RMA authorization check was performed by RMA user exit for the incoming request. Valid authorization was found to receive.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Incoming request message ID
<i>Action</i>	None	
156	SNET E RMACKFALS %1(%2) Check failure for RMA authorization-to-send for incoming request with message id %3. Reason: %4 (%5)	
<i>Type</i>	Error	
<i>Meaning</i>	RMA authorization-to-send check was performed by RMA user exit for the incoming request, but the check failed.	

Parameters	1	Access Point name
	2	Message partner
	3	Incoming request message ID
	4	Error description (reason for the check failure - optional)
	5	Error code message
Action	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	
157	SNET E RMACKFALR %1(%2) Check failure for RMA authorization-to-receive for incoming request with message id %3. Reason: %4 (%5)	
Type	Error	
Meaning	RMA authorization-to-receive check was performed by RMA user exit for the incoming request, but the authorization check failed.	
Parameters	1	Access Point name
	2	Message partner
	3	Incoming request message ID
	4	Error description (reason for the check failure - optional)
	5	Error code message
Action	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	
158	SNET W RMAFAILTP %1(%2) RMA authorization validation failed in trial period, but stop unauthorized traffic in trial is not set, for incoming request with message id %3	
Type	Warning	
Meaning	RMA authorization check was performed by RMA user exit in trial period. The check failed, but the configuration option for stopping the unauthorized traffic in trial period was disabled and the transfer was authorized.	

<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Incoming request message ID
<i>Action</i>	Enable the configuration option for stopping the unauthorized traffic in trial period if you want to reject the transfer requests for which the RMA authorization check failed.	

159	SNET I RMACKOKTS %1(%2) [%3] Check RMA authorization-to-send for outgoing request: valid authorization found	
<i>Type</i>	Information	
<i>Meaning</i>	RMA authorization check was performed by RMA user exit for the outgoing request. Valid authorization to send was found.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
<i>Action</i>	None	

160	SNET I RMACKOKTR %1(%2) [%3] Check RMA authorization-to-receive for outgoing request: valid authorization found	
<i>Type</i>	Information	
<i>Meaning</i>	RMA authorization check was performed by RMA user exit for the outgoing request. Valid authorization to receive was found.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
<i>Action</i>	None	

161	SNET E RMACKFLTS %1(%2) [%3] Check failure for RMA authorization-to-send for outgoing request. Reason: %4 (%5)	
<i>Type</i>	Error	
<i>Meaning</i>	RMA authorization-to-send check was performed by RMA user exit for the outgoing request, but the authorization check failed.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Error description (reason for the check failure - optional)
	5	Error code message
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	
162	SNET E RMACKFLTS %1(%2) [%3] Check failure for RMA authorization-to-receive for outgoing request. Reason: %4 (%5)	
<i>Type</i>	Error	
<i>Meaning</i>	RMA authorization-to-send check was performed by RMA user exit for the outgoing request, but the authorization check failed.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Error description (reason for the check failure - optional)
	5	Error code message
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	
163	SNET W RMAKOTRT %1(%2) [%3] RMA authorization validation failed in trial period, but stop unauthorized traffic in trial is not set; processing will continue	

Type	Warning	
Meaning	RMA authorization check was performed by RMA user exit in trial period. The check failed, but the configuration option for stopping the unauthorized traffic in trial period was disabled and the transfer request was authorized (default behavior).	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	Enable the configuration option for stopping the unauthorized traffic in trial period if you want to reject the transfer requests for which the RMA authorization check failed.	
170	SNET I DELINCHRQ %1(%2) deleting input channel %3	
Type	Information	
Meaning	Input channel was scheduled for deletion	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
Action	None	
171	SNET I DELINCHCF %1(%2) input channel %3 deleted	
Type	Information	
Meaning	Input channel was deleted.	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
Action	None	

172	SNET E DELINCHER %1(%2) error deleting input channel %3 : %4	
Type	Error	
Meaning	Delete of input channel failed.	
Parameters	1	Access Point name
	2	Message partner
	3	Input channel name
	4	Error message
Action	Check whether the channel name is valid and complies with SWIFT specifications. Also, the input channel might be open.	
180	SNET I CREAOCRQ %1(%2) creating output channel %3	
Type	Information	
Meaning	Output channel was scheduled for creation.	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
Action	None	
181	SNET I CREAOCOK %1(%2) output channel %3 successfully created	
Type	Information	
Meaning	Output channel was created.	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
Action	None	

182	SNET E CREAOCRQF %1(%2) error creating output channel %3. Reason: %4	
Type	Error	
Meaning	Output channel was not created.	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Error message
Action	Check whether the channel name is valid and complies with SWIFT specifications. Also, the output channel might be already created.	
183	SNET I DELOCRQ %1(%2) deleting output channel %3	
Type	Information	
Meaning	Output channel was scheduled for deletion	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
Action	None	
184	SNET I DELOCOK %1(%2) output channel %3 successfully deleted	
Type	Information	
Meaning	Output channel was deleted.	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name

<i>Action</i>	None	
185	SNET E DELOCFAIL %1(%2) error deleting output channel %3. Reason: %4	
<i>Type</i>	Error	
<i>Meaning</i>	Output channel was scheduled for creation.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Error message
<i>Action</i>	Check whether the channel name is valid and complies with SWIFT specifications. The output channel might be open or does not exist.	
190	SNET W OPOCDUP %1(%2) Open output channel %3 (queue %4) failed, an output session already exists or broken	
<i>Type</i>	Warning	
<i>Meaning</i>	An open output channel operation failed because an output session already was opened or an error occurred in the output session management mechanism.	
<i>Parameters</i>	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Queue name
<i>Action</i>	Check whether the output channel can be used for Swift communication. Contact Axway support if no solution is found.	
191	SNET W CLOCMISS %1(%2) Close output channel %3 (queue %4) failed, an output session does not exist or broken	

Type	Warning	
Meaning	A close output channel operation failed because an output session was not opened previously or an error occurred in the output session management mechanism	
Parameters	1	Access Point name
	2	Message partner
	3	Output channel name
	4	Queue name
Action	Check whether the output channel is closed. Contact Axway support if no solution is found.	
192	SNET W ACQQDUP %1(%2) Acquire queue %3 failed, an output session already exists or broken	
Type	Warning	
Meaning	An acquire queue operation failed because an output session already was opened or an error occurred in the output session management mechanism	
Parameters	1	Access Point name
	2	Message partner
	3	Queue name
Action	Check whether the queue can be used for Swift communication. Contact Axway support if no solution is found.	
193	SNET W RELQMISS %1(%2) Release queue %3 failed, an output session does not exist or broken	
Type	Warning	
Meaning	A release queue operation failed because an output session was not opened previously or an error occurred in the output session management mechanism	

Parameters	1	Access Point name
	2	Message partner
	3	Queue name
Action	Check whether the queue is released. Contact Axway support if no solution is found.	

194	SNET E MIXQCERR %1(%2) Conflicting declaration for %3. Already declared with %4 at end point %5	
Type	Error	
Meaning	Conflicting situation generated by a double declaration of a queue or output channel.	

Parameters	1	Access Point name
	2	Message partner
	3	Queue or output channel declaration
	4	Conflicting (existing) queue or output channel
	5	Conflicting (initial) declaration remote site alias
Action	Remove the duplicated declaration from one of the remote sites involved in conflict.	

195	SNET E DUPQCERR %1(%2) Double %3 declaration for %4. Already declared with %5	
Type	Error	
Meaning	A queue or output channel was already declared.	

Parameters	1	Access Point name
	2	Message partner
	3	Duplication type (empty if there is a simple duplication or backup if site is part of a chain of backup sites)
	4	Queue or output channel
	5	Conflicting (initial) declaration remote site alias
Action	Remove the duplicated declaration from one of the remote sites involved in conflict.	
196	SNET I DUPQCNOM %1(%2) Main declaration for %3. Already declared with backup site %4	
Type	Information	
Meaning	An output session was already declared in a backup site	
Parameters	1	Access Point name
	2	Message partner
	3	Queue or output channel
	4	Backup site name
Action	None	
197	SNET I DUPQCBCK %1(%2) Backup declaration for %3. Already declared with nominal site %4	
Type	Information	
Meaning	An output session was already declared in the nominal site	
Parameters	1	Access Point name
	2	Message partner
	3	Queue or output channel
	4	Nominal site name

<i>Action</i>	None

198	SNET W QRESET %1 (%2) Output session %3 for queue %4 reset from %5 to %6
<i>Type</i>	Warning
<i>Meaning</i>	The output session status was reset to prevent it remaining in a non-working intermediary state.
<i>Parameters</i>	1 Access Point name
	2 Message partner
	3 SnF session number
	4 Queue name
	5 Initial output session state
	6 New output session state
<i>Action</i>	Check the new output session state. If the output session was reset to DOWN, re-acquiring the corresponding queue might be necessary for continuing functionality of the product.

199	SNET W OCRESET %1 (%2) Output session %3 for output channel %4 (queue %5) reset from %6 to %7
<i>Type</i>	Warning
<i>Meaning</i>	The output session status was reset to prevent it remaining in a non-working intermediary state.

Parameters	1	Access Point name
	2	Message partner
	3	SnF session number
	4	Output channel name
	5	Queue name
	6	Initial output session state
	7	New output session state
Action	Check the new output session state. If the output session was reset to DOWN, re-opening the corresponding output channel might be necessary for continuing functionality of the product.	
200	SNET I ASPLDOK Swift ASP package reload done. Start time %1, end time %2, loaded files %3	
Type	Information	
Meaning	ASP package was loaded or reloaded on ASP helper process.	
Parameters	1	Time when the load or reload operation started
	2	Time when the load or reload operation ended
	3	Number of service profiles loaded or reloaded
Action	None	
210	SNET I GETFSCHK %1(%2) [%3] Checking status for pending FileAct transfer	
Type	Information	
Meaning	A GetFileStatus SWIFTNet query was scheduled for the specified transfer.	

Parameters	1	Access Point name
	2	Message partner
	3	NLocal request identifier (number)
Action	None (the transfer status will be updated automatically). If this message appears often, it might indicate a performance problem.	
211	SNET I GETFSOK %1(%2) [%3] Swift status for pending FileAct transfer is: %4; updating transfer state	
Type	Information	
Meaning	The transfer status will be updated according to its SWIFTNet status.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
	4	Transfer SWIFTNet status
Action	None (the transfer status will be updated automatically).	
212	SNET E GETFSFAIL %1(%2) [%3] Failed to get status for pending FileAct transfer; transfer cancelled	
Type	Error	
Meaning	The GetFileStatus SWIFTNet query failed.	
Parameters	1	Access Point name
	2	Message partner
	3	Local request identifier (number)
Action	Check the messages indicating the cause of the problem. If no solution is found, contact Axway Support.	

230	SNET E ACKLFAIL High Availability userexit library load failed: %1
<i>Type</i>	Error
<i>Meaning</i>	An error occurred while loading the SWIFTNet High Availability library. The reason is specified in parameter %1
<i>Parameters</i>	1 The library load error reason
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.
231	231 SNET E ACKLFAIL High Availability userexit library unload failed: %1
<i>Type</i>	Error
<i>Meaning</i>	An error occurred while loading the SWIFTNet High Availability library. The reason is specified in parameter %1
<i>Parameters</i>	1 The library unload error reason
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.
232	232 SNET E ACKIFAIL High Availability userexit initialization failed: %1
<i>Type</i>	Error
<i>Meaning</i>	An error occurred while initializing the SWIFTNet High Availability library. The reason is specified in parameter %1
<i>Parameters</i>	1 The library initialization error reason
<i>Action</i>	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.
235	235 SNET E ACKUEERR %1(%2) [%3] High Availability userexit %4 failed; error: %5
<i>Type</i>	Error

Meaning	A SWIFTNet High Availability operation failed.	
Parameters	1	Access Point name
	2	Message Partner
	3	Local request identifier (number)
	4	The SWIFTNet High Availability user exit operation which failed
	5	SWIFTNet HA operation error reason
Action	Check the messages indicating the cause of the problem. Contact Axway support if no solution is found.	
236	236 SNET E ACKUEERR License Key is not valid for SWIFTNet High Availability	
Type	Error	
Meaning	SWIFTNet High Availability feature is enabled, but the license key does not allow High Availability functionality for SWIFTNet.	
Parameters	None	
Action	Set a license key which enables the SWIFTNet High Availability feature.	
237	SNET E ACKUEERR License Key is not valid for Gateway RMA	
Type	Error	
Meaning	Gateway RMA feature is enabled, but the license key does not allow RMA functionality for SWIFTNet.	
Parameters	None	
Action	Set a license key which enables the Gateway RMA functionality.	

Axway Gateway: Messages and codes

TCP Server Daemon (TSD)

This topic lists TCP Server Daemon log messages.

001	TSD I CPORT_UP Control port %1 successfully attached	
Type	Information	
Meaning	Indicates that the TSD is ready to communicate with its dynamic servers.	
Parameters	1	Port name comprised of its local address and port number
Action	None	
002	TSD I SPORT_UP Port %2 for service %1 successfully attached	
Type	Information	
Meaning	Indicates that the TSD now listens to incoming connections on port %2 for the service %1.	
Parameters	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
	2	Port name comprised of its local address and port number
Action	None	
003	TSD I PROCS_UP Process %1 successfully launched for service %2	
Type	Information	
Meaning	A new dynamic process has been launched and is ready to accept connections for the service %2.	
Parameters	1	Process name
	2	Service name (HTTP, FTP, PEL, ODT or PHSE)
Action	None	
004	TSD I PROCS_DN Process %1 successfully ended for service %2	
Type	Information	
Meaning	A dynamic process ended as planned and can be launched again later on.	

Parameters	1	Process name
	2	Service name (HTTP, FTP, PEL, ODT or PHSE)
Action	None	
005	TSD I ICONX_UP Incoming connection port %2 for service %1 accepted	
Type	Information	
Meaning	A dynamic process of the service %2 accepted the incoming connection.	
Parameters	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
	2	Port name comprised of its local address and port number
Action	None	
006	TSD I ICONX_DN Incoming connection for service %1 closed	
Type	Information	
Meaning	A dynamic process for the service %1 closed a connection normally.	
Parameters	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
Action	None	
007	TSD E CPORT_ER Can't attach control port %1	
Type	Error	
Meaning	The TSD is not able to communicate with its dynamic servers and cannot go on.	
Parameters	1	Port name comprised of its local address and port number
Action	Check that the port is not already occupied. If other ports where specified, the TSD is able to find them. If there is no remaining free port, the TSD stops.	

008	TSD W SPORT_ER Can't attach port %2 for service %1	
<i>Type</i>	Warning	
<i>Meaning</i>	A listen point was not created for the service %1.	
<i>Parameters</i>	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
	2	Port name comprised of its local address and port number
<i>Action</i>	Check that the port is not already occupied. If the service has other ports, it still may be able to receive incoming connections. Check the TCP/IP settings in the Gateway configuration menu.	
009	TSD E PROCS_NO Process launch failed for service %1	
<i>Type</i>	Error	
<i>Meaning</i>	A dynamic process was not launched for the service %1.	
<i>Parameters</i>	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
<i>Action</i>	Check that no binary file has been deleted, renamed or moved away. Restart Gateway when the problem is fixed.	
010	TSD E PROCS_ER Process %1 abnormally ended for service %2	
<i>Type</i>	Error	
<i>Meaning</i>	A dynamic process of the service %2 unexpectedly stopped.	
<i>Parameters</i>	1	Process name
	2	Service name (HTTP, FTP, PEL, ODT or PHSE)
<i>Action</i>	The process will be automatically restarted if necessary.	
011	TSD E ICONX_NO Incoming connection port %2 for service %1 refused	
<i>Type</i>	Error	

Meaning	The service %2 is not currently able to handle more connections.	
Parameters	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
	2	Port name comprised of its local address and port number
Action	Check the service for the reason.	
012	TSD W SPORT_CO Conflicting port %1 between services %2 and %3	
Type	Warning	
Meaning	Two services are using the same port.	
Parameters	1	Port name comprised of its local address and port number
	2	Service name (HTTP, FTP, PEL, ODT or PHSE)
Action	Check the TCP/IP settings in the Gateway configuration menu. Change conflicting ports.	
014	TSD I MAXC_LOW Maximum connections for service %1 lowered to %2 due to process resource limit	
Type	Information	
Meaning	The number of maximum possible connections has been reduced due to resource limits.	
Parameters	1	Service name (HTTP, FTP, PEL, ODT or PHSE)
	2	New maximum connections limit
Action	Manage connections according to resource limits.	
020	TSD I XSR_CTING [%1] Connecting to XSR agent	
Type	Information	
Meaning	Connecting to the Secure Relay Router Agent.	

Parameters	1	IP address/TCP port
Action	None	
021	I XSR_CTED [%1] Connection with XSR agent established	
Type	Information	
Meaning	A connection with the Secure Relay Router Agent has been established.	
Parameters	1	IP address/TCP port
Action	None	
022	TSD W XSR_CREFU [%1] Connection refused by XSR agent	
Type	Warning	
Meaning	A connection with the Secure Relay Router Agent has been refused.	
Parameters	1	IP address/TCP port
Action	Check the configuration of the connection.	
023	TSD W XSR_CDOWN [%1] Connection with XSR agent broken	
Type	Warning	
Meaning	The connection with the Secure Relay Router Agent is broken.	
Parameters	1	IP address/TCP port
Action	None	
024	TSD I XSR_SAPOK [%1] Sap %2 successfully routed to %3 by XSR agent	
Type	Information	

<i>Meaning</i>	The SAP indicated in the message has been successfully routed by the Secure Relay Router Agent to the address indicated in the message.	
<i>Parameters</i>	1	IP address/TCP port
	2	Local SAP address and access port
	3	Routing destination address and access port
<i>Action</i>	None	

025	TSD E XSR_SAPKO [%1] Sap %2 not routed to %3 by XSR agent: error=%4	
<i>Type</i>	Error	
<i>Meaning</i>	The SAP indicated in the message has not been routed by the Secure Relay Router Agent to the address indicated in the message. The error code is included in the message.	
<i>Parameters</i>	1	IP address/TCP port
	2	Local SAP address and access port
	3	Routing destination address and access port
	4	Secure Relay error code: • 2 = Secure Relay Router Agent could not establish SAP, either because the port number is non-valid or the port is occupied • 4 = The Secure Relay Router Agent is not started
<i>Action</i>	Interpret the message depending on the error code.	

026	TSD E XSR_UNERR [%1] Unexpected error: %2	
<i>Type</i>	Error	
<i>Meaning</i>	Unexpected error.	
<i>Parameters</i>	1	IP address/TCP port
	2	Secure Relay error text message
<i>Action</i>	Contact Axway Support.	

027	TSD E XSR_VMISM [%1] XSR version mismatch, expecting %2-%3 received %4	
Type	Error	
Meaning	Secure Relay version out of range.	
Parameters	1	IP address/TCP port
	2	Minimum version
	3	Current version
	4	Received version
Action	Check version prerequisites between Secure Relay Master Agent and Router Agent.	
028	TSD E XSR_HELKO [%1] Error %2 received in HELLO RESP	
Type	Error	
Meaning	Secure Relay error.	
Parameters	1	IP address/TCP port
	2	Error ID
Action	Check version prerequisites between Secure Relay Master Agent and Router Agent.	
029	TSD E XSR_UNRSP [%1] Unexpected response code=%2 received	
Type	Error	
Meaning	Possible protocol error.	
Parameters	1	IP address/TCP port
	2	Response code
Action	Make sure you are using the right version of Secure Relay.	

030	TSD I XSR_HELOK [%1] Handshake XSR OK, will use XSR protocol version %2	
Type	Information	
Meaning	The protocol handshake completed successfully and the common protocol version was established.	
Parameters	1	IP Address and port
	2	Protocol version
Action	None	
031	TSD I XSR_CFINI [%1] Connection with XSR agent finished	
Type	Information	
Meaning	Connection terminated correctly	
Parameters	1	IP address/TCP port
Action	None	
032	TSD I XSR_VLOWR [%1] HELLO RESP received, lower XFBRELAY version required	
Type	Information	
Meaning	Server version problem (too high for client)	
Parameters	1	IP address/TCP port
Action	Check Secure Relay Router Agent version.	
033	TSD I XSR_LG_OK XSR-MA successfully establishes the log channel	
Type	Information	
Meaning	Secure Relay Master Agent has successfully established the log channel.	
Parameters	None	
Action	None	

034	TSD E XSR_LG_KO XSR-MA is unable to establish the log channel. Cause: %1	
Type	Error	
Meaning	Secure Relay Master Agent cannot establish the log channel.	
Parameters	1	Error cause
Action	Check Secure Relay Router Agent version.	
035	TSD E XSR_NOSPR A security profile should be set for service %1 port %2	
Type	Error	
Meaning	No security profile has been set for the reported service.	
Parameters	1	Service name
	2	Service port
Action	Create a security profile for the reported service.	
036	TSD E XSR_NOSPR Security params encoding error for service %1 port %2 security profile %3	
Type	Error	
Meaning	There was an en encoding error for the parameters for the given service, port and security profile.	
Parameters	1	Service name
	2	Service port
	3	Service security profile
Action	Check that the service name, port and security profile are correctly set.	
037	TSD E XSR_GENER %1	

Type	Error	
Meaning	An error has occurred. This log message is used for miscellaneous errors.	
Parameters	1	Error description
Action	Depends on the error description.	
038	TSD W XSR_VERMI SecureRelay protocol version is %1, it should be %2	
Type	Warning	
Meaning	The Secure Relay protocol version is different from the one used by Gateway.	
Parameters	1	Secure Relay protocol version
	2	Gateway protocol version
Action	Make sure you have the right version of Secure Relay.	
039	TSD E XSR_UNREQ Invalid SecureRelay request %1 received, expected a LOGIN_IND request	
Type	Error	
Meaning	An invalid request was received from Secure Relay instead of a LOGIN_IND request.	
Parameters	1	Request ID
Action	Make sure you are using the right version of Secure Relay.	
040	TSD I XSR_LINRD SecureRelay login service ready on %1/%2	
Type	Information	
Meaning	The Secure Relay login service is ready on the given address and port.	
Parameters	1	Host address
	2	Port number
Action	None	

Time Driver (TDRV)

This topic lists Time Driver log messages.

000	TDRV E DIR_PRB The watch directory %1 does not exist
<i>Type</i>	Error
<i>Meaning</i>	Watch directory does not exist.
<i>Parameters</i>	Name of directory
<i>Action</i>	Check the corresponding watch directory parameters in the Rule Table.
001	TDRV E DIR_PRB The watch directory %1 cannot be opened
<i>Type</i>	Error
<i>Meaning</i>	Impossible to access watch directory.
<i>Parameters</i>	Name of directory
<i>Action</i>	Check the <i>read</i> and <i>write</i> rights for the watch directory.
002	TDRV E DIR_PRB The work directory %1 does not exist
<i>Type</i>	Error
<i>Meaning</i>	Work directory does not exist.
<i>Parameters</i>	Name of directory
<i>Action</i>	Check the corresponding work directory parameters in the Rule Table.
003	TDRV E DIR_PRB The work directory %1 cannot be opened
<i>Type</i>	Error
<i>Meaning</i>	Impossible to access the work directory.

<i>Parameters</i>	Name of directory
<i>Action</i>	Check the <i>read</i> and <i>write</i> rights for the work directory.
570	TDRV E MB_CLEA_E
<i>Type</i>	Error
<i>Meaning</i>	Refer to message number 570 in the Rule Table evaluation log message topic.
579	TDRV I WORK_DONE
<i>Type</i>	Information
<i>Meaning</i>	Refer to message number 579 in the Rule Table evaluation log message topic.

Axway Gateway: Messages and codes

Trade messages (TRADE)

This topic lists TRADE log messages.

101	TRADE I TRACE [%1]
<i>Type</i>	Information
<i>Meaning</i>	Generic message information, details in the message.
<i>Parameters</i>	1 Message
102	TRADE E OPENERR [%1] Couldn't open file %2 (%3)
<i>Type</i>	Error
<i>Meaning</i>	Transfer file could not be opened.
<i>Parameters</i>	1 Transfer identifier
	2 File name
	3 File directory

103	TRADE E READERR [%1] Invalid file format %2 (%3)	
<i>Type</i>	Error	
<i>Meaning</i>	Received file has an unexpected format.	
<i>Parameters</i>	1	Transfer identifier
	2	File name
	3	File directory
104	TRADE E ACK [%1] Invalid MIC returned ('%2') expected '%3' [%4]	
<i>Type</i>	Error	
<i>Meaning</i>	Received acknowledgment does not contain the expected MIC.	
<i>Parameters</i>	1	Transfer identifier
	2	Received MIC
	3	Expected MIC
	4	Acknowledged transfer identifier
105	TRADE E ACK [%1] Invalid signature returned ('%2') expected '%3' [%4]	
<i>Type</i>	Error	
<i>Meaning</i>	Received acknowledgment does not contain the expected signature (S/MIME only).	
<i>Parameters</i>	1	Transfer identifier
	2	Received signature
	3	Expected signature
	4	Acknowledged transfer identifier

106	TRADE E ACK [%1] Invalid content type returned ('%2') expected '%3' [%4]	
<i>Type</i>	Error	
<i>Meaning</i>	Received acknowledgment does not contain the expected Content-Type (S/MIME only).	
<i>Parameters</i>	1	Transfer identifier
	2	Received Content-type
	3	Expected Content-type
	4	Acknowledged transfer identifier
107	TRADE E ACK [%1] Invalid content id returned ('%2') expected '%3' [%4]	
<i>Type</i>	Error	
<i>Meaning</i>	Received acknowledgment does not contain the expected Content-Id (S/MIME only).	
<i>Parameters</i>	1	Transfer identifier
	2	Received Content-Id
	3	Expected Content-Id
	4	Acknowledged transfer identifier
108	TRADE E ACK [%1] Invalid %3 originator '%2'	
<i>Type</i>	Error	
<i>Meaning</i>	Received acknowledgment contains invalid originator.	
<i>Parameters</i>	1	Transfer identifier
	2	Originator
	3	File or receipt
<i>Action</i>	Contact your exchange partner	

109	TRADE E ACK [%1] Invalid %3 destination '%2'	
Type	Error	
Meaning	Received acknowledgment contains invalid destination.	
Parameters	1	Transfer identifier
	2	Destination
	3	File or receipt
Action	Contact your exchange partner	
110	TRADE E RECV_ERR [%1] Transfer has been already received ([%2])	
Type	Error	
Meaning	Duplicated transfer.	
Parameters	1	Transfer identifier
	2	Already received transfer identifier
111	TRADE I ACK [%1] acknowledgment found: [%2]	
Type	Information	
Meaning	An acknowledgment has been found for the received transfer (only logged for AS2 synchronous acknowledgment requests).	
Parameters	1	Transfer identifier
	2	Acknowledged transfer identifier
112	TRADE I NO_ACK [%1] not yet acknowledged	
Type	Information	

Meaning	No acknowledgment has been found for the received transfer (only logged for AS2 synchronous acknowledgment requests).	
Parameters	1	Transfer identifier
113	TRADE I ACK [%1] acknowledges transfer [%2]	
Type	Information	
Meaning	A valid acknowledgment has been received.	
Parameters	1	Transfer identifier
	2	Acknowledged transfer identifier
114	TRADE E ACK [%1] acknowledged transfer not found	
Type	Error	
Meaning	An unknown acknowledgment has been received. Acknowledged transfer could not be found.	
Parameters	1	Transfer identifier
115	TRADE E SECURITY [%1] Invalid acknowledgment URL requested (%2)	
Type	Error	
Meaning	Received transfer has an invalid acknowledgment request.	
Parameters	1	Transfer identifier
	2	URL acknowledgment requested
Action	Check Trading Partner configuration.	
116	TRADE E SECURITY [%1] Received file is not signed	
Type	Error	

Meaning	Received file is not signed.	
Parameters	1	Transfer identifier
Action	Check Trading Partner configuration.	
117	TRADE E SECURITY [%1] Received file is not ciphered	
Type	Error	
Meaning	Received file is not encrypted.	
Parameters	1	Transfer identifier
Action	Check Trading Partner configuration.	
118	TRADE E SECURITY [%1] Requested acknowledgment doesn't match trading partner configuration	
Type	Error	
Meaning	Requested acknowledgment security (signed/unsigned/none) does not match trading partner configuration.	
Parameters	1	Transfer identifier
Action	Check Trading Partner configuration.	
120	TRADE I DECODED [%1] %2 %3 file received from '%4' to '%5'	
Type	Information	
Meaning	Incoming file for local partner has been correctly unpacked	
Parameters	1	Transfer identifier
	2	Packaging format (S/MIME, AS1, AS3, AS3)
	3	File or receipt depending on the file being handled
	4	Originator
	5	Destination

121	TRADE I DECODED [%1] %2 %3 file received from '%4' to remote partner '%5'	
<i>Type</i>	Information	
<i>Meaning</i>	Incoming file for remote partner has correctly been received.	
<i>Parameters</i>	1	Transfer identifier
	2	Packaging format (S/MIME, AS1, AS3, AS3)
	3	File or receipt, depending on the file being handled
	4	Originator
	5	Destination
122	TRADE E ERROR [%1] Decoding error: %2	
<i>Type</i>	Error	
<i>Meaning</i>	A PassPort PS SECS error occurred while decoding.	
<i>Parameters</i>	1	Transfer identifier
	2	Error message
<i>Action</i>	Refer to the error message.	
123	TRADE I ENCODED [%1] %2 %3 file encoded from '%4' to '%5'	
<i>Type</i>	Information	
<i>Meaning</i>	Outgoing file has been packed correctly.	
<i>Parameters</i>	1	Transfer identifier
	2	Packaging format (S/MIME, AS1, AS3, AS3)
	3	File or receipt, depending on the file being handled
	4	Originator
	5	Destination

124	TRADE W ACK [%1] acknowledges transfer [%2] with error '%3'	
Type	Warning	
Meaning	A negative acknowledgment has been received.	
Parameters	1	Transfer identifier
	2	Acknowledged transfer identifier
	3	Error message
Action	Refer to error message received.	
125	TRADE I ENCODED [%1] %2 %3 file to send from '%4' to '%5'	
Type	Information	
Meaning	A routed file is ready to be sent.	
Parameters	1	Transfer identifier
	2	Packaging format (S/MIME, AS1, AS3, AS3)
	3	File or receipt depending on the file being handled
	4	Originator
	5	Destination
126	TRADE E DECODE [%1] Signature cannot be trusted	
Type	Error	
Meaning	This message is displayed when the signature cannot be verified. Usually this message is displayed when the signature certificate does not correspond with the expected certificate.	
Parameters	1	Transfer identifier
127	TRADE I ACK [%1] should acknowledge transfer [%2]	

<i>Type</i>	Information	
<i>Meaning</i>	A new packaging task has been started.	
<i>Parameters</i>	1	Transfer ID of ACK message
	2	Transfer ID of message to be acknowledged
<i>Action</i>	None	

130 TRADE I ENCODE [%1] Preparing %2 to send

<i>Type</i>	Information - Log level 2	
<i>Meaning</i>	A new packaging task has been started.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt depending on the file being handled

131 TRADE I ENCODE [%1] Start encoding %2

<i>Type</i>	Information - Log level 2	
<i>Meaning</i>	File is to be encoded.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt depending on the file being handled

132 TRADE I ENCODE [%1] End encoding %2

<i>Type</i>	Information - Log level 2	
<i>Meaning</i>	File has been encoded correctly.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt depending on the file being handled

133	TRADE E ENCODE [%1] Encoding %2 aborted	
Type	Error - Log level 2	
Meaning	Failed to encode file.	
Parameters	1	Transfer identifier
	2	File or receipt, depending on the file being handled
Action	If present, refer to the corresponding log message 134.	
134	TRADE E ERROR [%1] Encoding error: %2	
Type	Error	
Meaning	A PassPort PS SECS error occurred while encoding.	
Parameters	1	Transfer identifier
	2	Error message
Action	Refer to the error message.	
140	TRADE I ENCODE [%1] Extracting information from encoded %2	
Type	Information - Log level 2	
Meaning	A new task has been started to check a routed file content.	
Parameters	1	Transfer identifier
	2	File or receipt, depending on the file being handled
141	TRADE I INFO [%1] Originator '%2' is partner '%3'	
Type	Information - Log level 1	
Meaning	Trading Partner object found for the originator of the transfer.	

<i>Parameters</i>	1	Transfer identifier
	2	Partner identification name
	3	Matching Trading Partner object
142	TRADE I INFO [%1] Destination '%2' is partner '%3'	
<i>Type</i>	Information - Log level 1	
<i>Meaning</i>	Trading Partner object found for the destination of the transfer.	
<i>Parameters</i>	1	Transfer identifier
	2	Partner identification name
	3	Matching Trading Partner object
143	TRADE W INFO [%1] Destination '%2' does not match partner '%3'	
<i>Type</i>	Warning	
<i>Meaning</i>	The destination of the transfer has only been partially matched by a Trading Partner object. Some of the expected properties do not match (for example, allowed direction for the transfer).	
<i>Parameters</i>	1	Transfer identifier
	2	Partner identification name
	3	Trading Partner object
144	TRADE E INFO [%1] Originator '%2' not found	
<i>Type</i>	Error	
<i>Meaning</i>	No Trading Partner object has been found for the originator of the transfer.	
<i>Parameters</i>	1	Transfer identifier
	2	Partner identification name

145	TRADE E INFO [%1] Destination '%2' not found	
<i>Type</i>	Error	
<i>Meaning</i>	No Trading Partner object has been found for the destination of the transfer.	
<i>Parameters</i>	1	Transfer identifier
	2	Partner identification name
146	TRADE I INFO [%1] Message Id: %2	
<i>Type</i>	Information	
<i>Meaning</i>	Message Id (x_message_id) for outgoing and incoming transfers.	
<i>Parameters</i>	1	Transfer identifier
	2	Message id of the transfer
147	TRADE I INFO [%1] Receipt reply to transfer [%2]	
<i>Type</i>	Information	
<i>Meaning</i>	Transfer id to which the ack is sent (x_reply_to_xfer), for outgoing receipt transfers.	
<i>Parameters</i>	1	Transfer identifier of the receipt
	2	Transfer identifier of the original transfer
150	TRADE I DECODE [%1] Start analyzing received %2	
<i>Type</i>	Information	
<i>Meaning</i>	A new unpacking task has been started.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt, depending on the file being handled

151	TRADE I DECODE [%1] Start decoding %2	
<i>Type</i>	Information - Log level 2	
<i>Meaning</i>	Start unpacking the received file.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt, depending on the file being handled
152	TRADE I DECODE [%1] End decoding %2	
<i>Type</i>	Information - Log level 2	
<i>Meaning</i>	File has been unpacked correctly.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt, depending on the file being handled
153	TRADE E DECODE [%1] Decoding %2 aborted	
<i>Type</i>	Error - Log level 2	
<i>Meaning</i>	Failed to decode.	
<i>Parameters</i>	1	Transfer identifier
	2	File or receipt, depending on the file being handled
<i>Action</i>	If present, refer to the corresponding log message 122.	
154	TRADE W DECODE [%1] Duplicated file %2 has been deleted	
<i>Type</i>	Warning	
<i>Meaning</i>	The duplicate file indicated in the message has been deleted.	
<i>Parameters</i>	1	Transfer identifier
	2	File that was deleted
<i>Action</i>	None	

160	TRADE E SMAPI [%1] Invalid signing certificate for %3 partner '%2'	
<i>Type</i>	Error	
<i>Meaning</i>	Invalid signing certificate for partner. The cryptographic operation cannot continue.	
<i>Parameters</i>	1	Transfer identifier
	2	Trade partner name
	3	Protocol format
161	TRADE E SMAPI [%1] Invalid ciphering certificate for %3 partner '%2'	
<i>Type</i>	Error	
<i>Meaning</i>	Invalid ciphering certificate for partner. The cryptographic operation cannot continue.	
<i>Parameters</i>	1	Transfer identifier
	2	Trade partner name
	3	Protocol format
170	TRADE E ERROR [%1] Couldn't read dynamic information for '%2'	
<i>Type</i>	Error	
<i>Meaning</i>	Gateway could not find the EDI originator and destination. A timeout may have occurred when retrieving the information from PassPort.	
<i>Parameters</i>	1	Transfer identifier
	2	Identifier

<i>Action</i>	Check the value of the <code>vsite_query_timeout</code> parameter in the <code>pelsetup.def</code> file. This parameter specifies the waiting timeout value (in seconds) when retrieving information from PassPort. The default value is 20 seconds. If necessary, increase the value of the <code>vsite_query_timeout</code> parameter. For information about editing the <code>pelsetup.def</code> file, refer to Modifying conffile parameters.	
201	TRADE E TRACE [%1]	
<i>Type</i>	Error	
<i>Meaning</i>	Generic error message, details in the message.	
<i>Parameters</i>	1	Message

Axway Gateway: Messages and codes

Trading Partner Management (TPM)

This topic lists PassPort (Trading Partner) log messages.

101	TPM E INITERR Connector initialisation error %1	
<i>Type</i>	Error	
<i>Meaning</i>	Error on PassPort connector initialization.	
<i>Parameters</i>	1	Externally generated error message
<i>Action</i>	Correct configuration according to displayed message.	
102	TPM W IRESERR Initiator resolution error - dea_id=%2 protocol=%3 msg="%1"	
<i>Type</i>	Warning	
<i>Meaning</i>	Error resolving DEA ID (Gateway in Initiator mode).	

Parameters	1	Externally generated error message
	2	PassPort DEA ID
	3	Protocol
Action	Correct configuration according to displayed message.	
103	TPM W RRESERR Responder resolution error - user_name=%2 protocol=%3 msg="%1"	
Type	Warning	
Meaning	Error resolving DEA ID (Gateway in Responder mode).	
Parameters	1	Externally generated error message
	2	Remote partner user name
	3	Protocol
Action	Correct configuration according to displayed message.	
104	TPM W PMISMAT Protocol mismatch - dea_id=%1 expecting protocol %2 got %3	
Type	Warning	
Meaning	Protocol mismatch prevents connection.	
Parameters	1	PassPort DEA ID
	2	Protocol expected
	3	Protocol received
Action	Correct the remote Site protocol ID.	
105	TPM W DMISMAT Direction mismatch - dea_id=%1 expecting direction %2 got %3	
Type	Warning	

Meaning	Transfer direction mismatch prevents connection.	
Parameters	1	PassPort DEA ID
	2	Direction expected
	3	Direction received
Action	Correct Transfer Request direction attribute and resubmit Transfer Request, or use an alternate DEA ID via the <code>peltrans tpm_dea_id</code> command.	
106	TPM E APIERR %1 error - retcod=%2 errcode=%4 errmsg="%3"	
Type	Error	
Meaning	PassPort related API error.	
Parameters	1	One of the following API actions: • TpmConnect • TPMGetDeaById • TpmGetDeaByAttr • TpmDisconnect
	2	TPM Error code - generated externally to Gateway
	3	TPM Return code - generated externally Gateway
	4	TPM Message string - generated externally to Gateway
Action	Contact Axway Support.	
107	TPM E NOPARAMF No parameters file configured	
Type	Error	
Meaning	The PassPort parameter file name is not indicated in the Gateway configuration file.	
Parameters	None	
Action	Use the configuration menu to set the correct PassPort configuration parameters.	

108	TPM I INITDONE Initialisations done - paramfile="%1"	
Type	Information	
Meaning	Initialization is completed. Parameters are loaded to memory.	
Parameters	1	PassPort parameter file name
Action	None	
300	TPM E TPM entry not found	
Type	Error	
Meaning	The requested PassPort entry cannot be found.	
Parameters	None	
Action	Verify PassPort database definition.	
301	TPM E TPM initialization error	
Type	Error	
Meaning	Gateway cannot initialize the PassPort connector, because it has encountered an initialization error.	
Parameters	None	
Action	Verify that PassPort is installed on the platform and all prerequisites are met.	
302	TPM E TPM access error	
Type	Error	
Meaning	[message currently not used]	
Parameters	None	
Action	None	

303	TPM E Invalid request to TPM
<i>Type</i>	Error
<i>Meaning</i>	A mandatory parameter is missing, for example DEA_ID.
<i>Parameters</i>	None
<i>Action</i>	Correct the Transfer Request and re-submit.
304	TPM E TPM data mismatches request
<i>Type</i>	Error
<i>Meaning</i>	The data retrieved from PassPort has an unexpected value.
<i>Parameters</i>	None
<i>Action</i>	Check the data definition for the relevant PassPort entry.
305	TPM E TPM API error
<i>Type</i>	Error
<i>Meaning</i>	PassPort API returns an error.
<i>Parameters</i>	None
<i>Action</i>	Refer to the Gateway log file for more detailed information.
306	TPM E Invalid response from TPM
<i>Type</i>	Error
<i>Meaning</i>	The data retrieved from PassPort has an invalid or missing value, for example no DEA_ID, MODE or DIRECTION.
<i>Parameters</i>	None
<i>Action</i>	Check the data definition for the relevant PassPort entry.

Axway Gateway: Messages and codes

User Request (USR)

This topic lists User Request log messages.

000	USR S SUCCESS %1
001	USR I INFO %1
002	USR W WARNING %1
003	USR E ERROR %1
004	USR F FATAL %1
<i>Type</i>	• 000: Success • 001: Information • 002: Warning • 003: Error • 004: Fatal
<i>Meaning</i>	These messages are printed on a user exit request, using a log utility function that is supplied with the monitor (named LogPrintf). This function can be called from a user exit to print user messages in the log file. The message identification number (integer in the first column) is used by the utility function to select the user message type to print. This integer number is the first argument of the LogPrintf function: • 000: successful operation in the user exit function • 001: information message • 002: warning condition detected in the user exit function • 003: error condition detected in the user exit function • 004: fatal error condition detected in the user exit function
<i>Parameters</i>	1 Test of the message whose content and format are defined by the user
<i>Action</i>	None

Axway Gateway: Messages and codes

VFD errors

The error messages in this topic are not logged directly, but are used as parts of other log messages.

100	memory allocation error

Meaning	The VFD encountered a system error while allocating memory.	
Parameters	None	
Action	Check the operating system configuration and memory usage.	
101	memory allocation error	
Meaning	The VFD encountered a system error while reallocating memory.	
Parameters	None	
Action	Check the operating system configuration and memory usage.	
102	configuration parameter name %1 is invalid	
Meaning	The VFD received an invalid configuration parameter	
Parameters	1	Name of the unrecognized parameter
Action	Internal error. Contact Axway Support.	
103	invalid session id	
Meaning	The VFD has been called with an invalid session identifier	
Parameters	None	
Action	Internal error. Contact Axway Support.	
104	invalid root directory %1	
Meaning	A VFD session has been initiated with an invalid root directory	
Parameters	1	Root directory that caused the error
Action	Check the root directory parameter of the selected CGate and of its CGateGroup ancestors.	

105	invalid home directory %1	
<i>Meaning</i>	A VFD session has been initiated with an invalid home directory	
<i>Parameters</i>	1	Home directory that caused the error
<i>Action</i>	Check the home directory parameter of the selected CGate and of its CGateGroup ancestors.	
106	cd up error, directory %1 position %2	
<i>Meaning</i>	The VFD encountered an error while going one level up in a directory (cd ..)	
<i>Parameters</i>	1	Directory being moved up
	2	Parsing position when the error occurred
<i>Action</i>	Internal error. Contact Axway Support.	
107	parsing error, directory head type invalid	
<i>Meaning</i>	The VFD encountered an error while parsing a directory name	
<i>Parameters</i>	None	
<i>Action</i>	Internal error. Contact Axway Support.	
108	get path name error, directory %1 should be absolute	
<i>Meaning</i>	The VFD encountered an error while parsing a directory name.	
<i>Parameters</i>	1	Directory that caused the error
<i>Action</i>	Internal error. Contact Axway Support.	
109	get last name error	
<i>Meaning</i>	The VFD encountered an error while parsing a directory name	
<i>Parameters</i>	1	Directory that caused the error

<i>Action</i>	Internal error. Contact Axway Support.
110	node ident generation error
<i>Meaning</i>	The VFD encountered an error when generating a new node ident
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
111	index record not found
<i>Meaning</i>	The VFD encountered an error while reading in its database
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
112	index record not found
<i>Meaning</i>	The VFD encountered an error while adding a new index entry
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
113	absolute directory expected
<i>Meaning</i>	The VFD encountered an error while parsing a directory name.
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
114	home directory %1 is not under root directory
<i>Meaning</i>	A VFD session has been initiated with inconsistent root and home directories.

Parameters	1	Home directory that caused the error
Action	Check the root and home directories parameters of the selected CGate and of its CGateGroup ancestors. The home directory must be under or of the same level as equal to the root directory. Example: If /public/users is the root directory, the home directory can be /public/users/donald but it cannot be /public/our private stuff.	
115	invalid option %d	
Meaning	The VFD encountered an invalid option	
Parameters	None	
Action	Internal error. Contact Axway Support.	
116	root directory overstep	
Meaning	The VFD failed to change current directory to an invalid location. This may happen when a command like "cd .." is supplied in the "/" directory.	
Parameters	None	
Action	Internal error. Contact Axway Support.	
117	invalid directory %1	
Meaning	The VFD encountered an error while parsing a directory.	
Parameters	1	Directory that caused the error
Action	Internal error. Contact Axway Support.	
118	current directory invalid %1	
Meaning	The VFD encountered an error while parsing a directory	
Parameters	1	Directory that caused the error
Action	Internal error. Contact Axway Support.	

119	directory %1 already exists
<i>Meaning</i>	The VFD could not create a directory because it already exists
<i>Parameters</i>	1 Directory that could not be created
<i>Action</i>	Use existing directory or choose another name for the new directory
120	directory %1 creation error, parent bad type
<i>Meaning</i>	The VFD encountered an error while creating a directory
<i>Parameters</i>	1 Directory that could not be created
<i>Action</i>	A directory can be created only if its parent is a virtual directory. It is not possible to create a subdirectory in a mounted or real directory.
121	this type of directory can't be mounted
<i>Meaning</i>	The VFD encountered an error while mounting a directory
<i>Parameters</i>	None
<i>Action</i>	A directory can be mounted only if it is of the type virtual and is empty
122	mount directory %1 does not exist
<i>Meaning</i>	The VFD encountered an error while mounting a directory
<i>Parameters</i>	Directory that was not found on the file system
<i>Action</i>	An attempt was made to mount a virtual directory on a real directory (file system) that does not exist. Check the directory to be mounted.
123	only an empty directory can be mounted
<i>Meaning</i>	The VFD encountered an error while mounting a directory

<i>Parameters</i>	None
<i>Action</i>	An attempt was made to mount a virtual directory that is not empty. This operation is forbidden.
124	real entry name %1 is too long
<i>Meaning</i>	The VFD found an entry (file or directory) whose name exceeds 127 characters.
<i>Parameters</i>	File or directory name that exceeds the limit
<i>Action</i>	This may happen in real directories. The current limit is 127 characters for a file or directory name and 255 characters for a complete path name. This error prevents you from seeing the long file name but does not damage the VFD integrity.
125	unknown entry type (st_mode=%1)
<i>Meaning</i>	The VFD encountered a real entry whose type is unknown
<i>Parameters</i>	1 Numerical value of the unrecognized entry type
<i>Action</i>	st_mode is part of the system data type struct stat. Normally, entries have the type S_ISDIR or S_ISREG on UNIX and _S_IFDIR or _S_IFDIR on Windows. Contact Axway Support.
126	directory bad type, can not be unmounted
<i>Meaning</i>	The VFD encountered an error while deleting a directory
<i>Parameters</i>	None
<i>Action</i>	An attempt was made to unmount a directory that is not a mounted directory.
127	directory bad type, can not be deleted
<i>Meaning</i>	The VFD encountered an error while unmounting a directory.
<i>Parameters</i>	None

<i>Action</i>	Only a virtual empty directory can be deleted. A real directory can not be deleted. A mounted directory has to be unmounted (so that it becomes virtual) before deletion.
128	directory deletion not allowed
<i>Meaning</i>	The VFD encountered an error while deleting a directory.
<i>Parameters</i>	None
<i>Action</i>	This may happen if you try to delete the VFD root directory. The root directory must always exist and there should never be any need to delete it.
129	only an empty directory can be deleted
<i>Meaning</i>	An attempt was made to delete a directory that is not empty. This operation is not allowed.
<i>Parameters</i>	None
<i>Action</i>	Empty the directory and try again
130	max number of directories has been reached
<i>Meaning</i>	The VFD encountered an error while creating a directory. The maximum number of directories in the VFD is presently limited to 8192 (this includes virtual and mounted directories only, not real ones).
<i>Parameters</i>	None
<i>Action</i>	Delete any unused directories and try again.
131	directory motion not allowed
<i>Meaning</i>	The VFD encountered an error while moving a directory.
<i>Parameters</i>	None

<i>Action</i>	Make sure that you are not trying to move: • The root directory • The directory to a location that already contains a directory with the same name
132	directory bad type, can not be moved
<i>Meaning</i>	The VFD encountered an error while moving a real directory. You can only move virtual and mounted directories (not real directories).
<i>Parameters</i>	None
<i>Action</i>	Check that the directory is not a real directory and that the target location is not a virtual directory.
133	directory already exists
<i>Meaning</i>	The VFD encountered an error while creating a directory. The directory already exists.
<i>Parameters</i>	None
<i>Action</i>	This is rather an internal message and error number 119 (see above) is most likely to be reported.
134	directory bad type %1, can not register item
<i>Meaning</i>	The VFD encountered an error while recording an item in a directory that is not virtual. Only virtual directories can contain items.
<i>Parameters</i>	1 <i>Type of directory that caused the error</i>
<i>Action</i>	Contact Axway Support.
135	item already exists
<i>Meaning</i>	The VFD encountered an error while recording an item. The item already exists in the selected directory.

<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
136	item not found
<i>Meaning</i>	The VFD encountered an error while reading an item.
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
137	directory resolution error, bad input parameter
<i>Meaning</i>	The VFD encountered an error while reading an item.
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
138	directory resolution error, not found
<i>Meaning</i>	The VFD encountered an error while resolving a directory node ident.
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
139	not implemented
<i>Meaning</i>	The VFD was requested for a non-implemented service.
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
140	initialization option invalid

Meaning	The VFD was initialized with an unrecognized parameter.	
Parameters	None	
Action	Internal error. Contact Axway Support.	
142	initialization error, shared index is down	
Meaning	The VFD encountered an error while accessing its index table.	
Parameters	None	
Action	Internal error. Contact Axway Support.	
143	directory bad type, attributes cannot be set	
Meaning	The VFD encountered an error while setting directory attributes. You can only set attributes for virtual and mounted directories, not real directories.	
Parameters	None	
Action	Check that the directory is not a real directory.	
144	translate name bad type %1	
Meaning	The VFD encountered an error while translating a name to the file system.	
Parameters	1	Type of directory that caused the error
Action	Internal error. Contact Axway Support.	
145	data record already exists	
Meaning	The VFD encountered an error while writing in its database. The data record already exists.	
Parameters	None	
Action	Internal error. Contact Axway Support.	

146	data record not found
<i>Meaning</i>	The VFD encountered an error while reading in its database. Cannot find the data record.
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
147	directory alias %1 is duplicated
<i>Meaning</i>	The VFD encountered an error while setting a directory alias. The directory alias already exists in the VFD.
<i>Parameters</i>	1 Alias that is duplicated
<i>Action</i>	Aliases must be unique in the VFD. Set a different alias for the directory.
148	directory name %1 is reserved
<i>Meaning</i>	The VFD encountered an error while creating a directory. Directory names beginning with "_item_" are reserved for internal needs (for item storage).
<i>Parameters</i>	1 The reserved name that caused the error
<i>Action</i>	Do not use "_item_" at the beginning of the directory name.
149	mounted directory %1 is missing
<i>Meaning</i>	The VFD encountered an error while accessing a mounted directory.
<i>Parameters</i>	1 File system directory that caused the error
<i>Action</i>	A VFD directory has been mounted on a file system directory that is no longer present or reachable. Check the directory on your file system or unmount the VFD directory that points to it.
150	root directory %1 should be absolute
<i>Meaning</i>	A VFD session was initiated with an invalid root directory.

<i>Parameters</i>	1	Root directory that caused the error
<i>Action</i>	The root directory must be absolute (that is, beginning with a slash). Check the root directory parameter of the selected CGate and of its parent CGateGroup.	
151	home directory %1 should be absolute	
<i>Meaning</i>	A VFD session was initiated with an invalid home directory.	
<i>Parameters</i>	1	Home directory that caused the error
<i>Action</i>	The home directory must be absolute (that is, beginning with a slash). Check the home directory parameter of the selected CGate and of its parentCGateGroup.	
152	file deletion is impossible in a virtual directory	
<i>Meaning</i>	You cannot delete a file from a virtual directory.	
<i>Parameters</i>	None	
<i>Action</i>	None	
153	invalid file %s	
<i>Meaning</i>	The file you selected is not valid.	
<i>Parameters</i>	None	
<i>Action</i>	Select a valid file	
154	permission denied	
<i>Meaning</i>	You do not have permission for this action.	
<i>Parameters</i>	None	
<i>Action</i>	None	

155	directory name too long (%s)
<i>Meaning</i>	The directory name you entered is too long.
<i>Parameters</i>	None
<i>Action</i>	The maximum length of the directory name cannot exceed 128 characters.
156	directory path too long
<i>Meaning</i>	The directory path you entered is too long.
<i>Parameters</i>	None
<i>Action</i>	The maximum length of the directory path cannot exceed 255 characters.
157	drives list overflow
<i>Meaning</i>	If you use a Gateway GUI client, either remotely, or on the same machine as the Gateway server, when you mount a VFD, you select the real file directory to which you link the VFD from a <i>Browse</i> window. The number of drives that Gateway can display in the <i>Browse</i> window cannot exceed 255. Refer to Working with Virtual File Directories.
<i>Parameters</i>	None
<i>Action</i>	Reduce the number of devices accessible from your supporting machine.
158	no space left on device
<i>Meaning</i>	No remaining disk space.
<i>Parameters</i>	None
<i>Action</i>	Reconfigure drive space.
200	system error not implemented
<i>Meaning</i>	The VFD encountered an error while using system services.

<i>Parameters</i>	None				
<i>Action</i>	Internal error. Contact Axway Support.				
201	system error mkdir %1, errno=%2				
<i>Meaning</i>	The VFD encountered an error in <code>mkdir</code> system call.				
<i>Parameters</i>	<table> <tr> <td>1</td><td>Directory to be created</td></tr> <tr> <td>2</td><td>System errno value</td></tr> </table>	1	Directory to be created	2	System errno value
1	Directory to be created				
2	System errno value				
<i>Action</i>	Internal error. Contact Axway Support.				
202	system error creat %1, errno=%2				
<i>Meaning</i>	The VFD encountered an error in <code>creat</code> system call.				
<i>Parameters</i>	<table> <tr> <td>1</td><td>Path name to be created</td></tr> <tr> <td>2</td><td>System errno value</td></tr> </table>	1	Path name to be created	2	System errno value
1	Path name to be created				
2	System errno value				
<i>Action</i>	Internal error. Contact Axway Support.				
203	system error open %1, errno=%2				
<i>Meaning</i>	The VFD encountered an error in <code>open</code> system call.				
<i>Parameters</i>	<table> <tr> <td>1</td><td>File to be opened</td></tr> <tr> <td>2</td><td>System errno value</td></tr> </table>	1	File to be opened	2	System errno value
1	File to be opened				
2	System errno value				
<i>Action</i>	Internal error. Contact Axway Support.				
204	system error close, errno=%1				
<i>Meaning</i>	The VFD encountered an error in <code>close</code> system call.				
<i>Parameters</i>	<table> <tr> <td>1</td><td>System errno value</td></tr> </table>	1	System errno value		
1	System errno value				
<i>Action</i>	Internal error. Contact Axway Support.				

205	system error stat %1, errno=%2	
<i>Meaning</i>	The VFD encountered an error in stat system call.	
<i>Parameters</i>	1	File to be stated
	2	System errno value
<i>Action</i>	Internal error. Contact Axway Support.	
206	system error read, errno=%1	
<i>Meaning</i>	The VFD encountered an error in read system call.	
<i>Parameters</i>	1	System errno value
<i>Action</i>	Internal error. Contact Axway Support.	
207	system error read count, %1 out of %2	
<i>Meaning</i>	The VFD encountered an error in read system call.	
<i>Parameters</i>	1	Number of bytes already read
	2	Number of bytes expected
<i>Action</i>	Internal error. Contact Axway Support.	
208	system error write, errno=%1	
<i>Meaning</i>	The VFD encountered an error in write system call.	
<i>Parameters</i>	1	System errno value
<i>Action</i>	Internal error. Contact Axway Support.	
209	system error write count, %1 out of %2	

Meaning	The VFD encountered an error in write system call.	
Parameters	1	Number of bytes already written
	2	Number of bytes expected
Action	Internal error. Contact Axway Support.	
210	system error lseek set, errno=%1	
Meaning	The VFD encountered an error in lseek system call	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
211	system error lseek end, errno=%1	
Meaning	The VFD encountered an error in lseek system call.	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
212	system error lseek cur, errno=%1	
Meaning	The VFD encountered an error in lseek system call.	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
213	system error opendir %1, errno=%2	
Meaning	The VFD encountered an error in opendir system call.	
Parameters	1	Directory to be opened
	2	System errno value
Action	Internal error. Contact Axway Support.	

214	system error readdir, errno=%1	
Meaning	The VFD encountered an error in readdir system call.	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
215	system error closedir, errno=%1	
Meaning	The VFD encountered an error in closedir system call.	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
216	system error unlink %1, errno=%2	
Meaning	The VFD encountered an error in unlink system call.	
Parameters	1	File to be unlinked
	2	System errno value
Action	Internal error. Contact Axway Support.	
217	system error shmget, errno=%1	
Meaning	The VFD encountered an error in shmget system call.	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
218	system error shmat, errno=%1	
Meaning	The VFD encountered an error in shmat system call	

Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
219	system error shmdt, errno=%1	
Meaning	The VFD encountered an error in shmdt system call	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
220	system error shmctl, errno=%1	
Meaning	The VFD encountered an error in shmctl system call	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
221	system error ftok, errno=%1	
Meaning	The VFD encountered an error in ftok system call	
Parameters	1	System errno value
Action	Internal error. Contact Axway Support.	
222	system error rmdir %1, errno=%2	
Meaning	Error on removing directory	
Parameters	1	Directory name
	2	System error number
Action	Internal error. Contact Axway Support.	

223	system error rename %1 to %2, errno=%3	
<i>Meaning</i>	Error on renaming directory	
<i>Parameters</i>	1	Old directory name
	2	New directory name
	3	System error number
<i>Action</i>	Internal error. Contact Axway Support.	
224	system error fsync, errno=%1	
<i>Meaning</i>	System error	
<i>Parameters</i>	1	System error number
<i>Action</i>	Internal error. Contact Axway Support.	
225	system error getDrives, errno=%1	
<i>Meaning</i>	On Windows platforms only, error in getting list of drive devices	
<i>Parameters</i>	1	System error number
<i>Action</i>	Internal error. Contact Axway Support.	
301	internal error 001	
<i>Meaning</i>	An internal error occurred in the VFD	
<i>Parameters</i>	None	
<i>Action</i>	Internal error. Contact Axway Support.	
302	internal error 002	
<i>Meaning</i>	An internal error occurred in the VFD	

<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
303	internal error 003
<i>Meaning</i>	An internal error occurred in the VFD
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
304	internal error 004
<i>Meaning</i>	An internal error occurred in the VFD
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
305	internal error 005
<i>Meaning</i>	An internal error occurred in the VFD
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.
306	internal error 006
<i>Meaning</i>	An internal error occurred in the VFD
<i>Parameters</i>	None
<i>Action</i>	Internal error. Contact Axway Support.

Axway Gateway: Messages and codes

X.400 (X420) log messages

This topic lists X.400 log messages.

001	X420 S TRACE TRACE from %1: %2	
Type	Success	
Meaning	Generic trace.	
Parameters	1	Location in the program
	2	Trace message
Action	None	
002	X420 I INITING X420 initializing (Port=%1, Host=%2, Password=%3)	
Type	Information	
Meaning	Information message indicating the connection of the X.420 message transfer connector to the local X.420 UA.	
Parameters	1	TCP port
	2	TCP address
	3	Password
Action	None	
003	X420 I INITOK X420 initialization OK	
Type	Information	
Meaning	Information message indicating the successful initialization of the X.420 connector.	
Parameters	None	
Action	None	

011	X420 I MSGRCVD MSG received - originator_alias=%1 recipient_alias=%2 content_id=%3 ipm_urid=%4 ua_rn_id=%5	
Type	Information	
Meaning	Information message indicating the reception of a message from the local UA.	
Parameters	1	Originator
	2	Recipient
	3	Message content identifier
	4	IPM User ID
	5	UA receipt notification
Action	None	
012	X420 I MSGSENDING Sending MSG - originator_alias=%1 recipient_alias=%2 content_id=%3 ipm_urid=%4	
Type	Information	
Meaning	Information message indicating the transmission of a message to the local UA.	
Parameters	1	Originator
	2	Recipient
	3	Message content identifier
	4	IPM User ID
Action	None	
014	X420 I MSGSENT MSG sent - content_id=%1 size=%2 total time=%3	
Type	Information	
Meaning	Information message indicating the of transmission of a message to the local UA.	

Parameters	1	Message content identifier
	2	Number of byte transferred
	3	Total transfer time in hundreds of seconds
Action	None	

015	X420 I DNRCVD DN received - recipient_alias=%1 content_id=%2	
Type	Information	
Meaning	Information message indicating the reception of a delivery notification.	
Parameters	1	Recipient
	2	Message content identifier
Action	None	

016	X420 I NDNRCD NDN received - recipient_alias=%1 content_id=%2	
Type	Information	
Meaning	Information message indicating the reception of a non-delivery notification.	
Parameters	1	Recipient
	2	Message content identifier
Action	None	

017	X420 I RNRCVD RN received - originator_alias=%1 recipient_alias=%2 ipm_urid=%3	
Type	Information	
Meaning	Information message indicating the reception of a receipt notification.	

Parameters	1	Originator
	2	Recipient
	3	IPM User ID
Action	None	
018	X420 I NNRRCVD NRN received - originator_alias=%1 recipient_alias=%2 ipm_urid=%3	
Type	Information	
Meaning	Information message indicating the reception of a non-receipt notification.	
Parameters	1	Originator
	2	Recipient
	3	IPM User ID
Action	None	
019	X420 I RNSNDING Sending RN - content_id=%1 ua_rn_id=%2	
Type	Information	
Meaning	Information message indicating the transmission of a receipt notification.	
Parameters	1	Message content identifier
	2	UA receipt notification ID
Action	None	
020	X420 I NRNSNDING Sending NRN - content_id=%1 ua_rn_id=%2	
Type	Information	
Meaning	Information message indicating the transmission of a non-receipt notification.	

Parameters	1	Message content identifier
	2	UA receipt notification ID
Action	None	
021	X420 I NORSENDING Sending NOR - ua_rn_id=%1	
Type	Information	
Meaning	Information message indicating the transmission of no receipt notification.	
Parameters	1	UA receipt notification ID
Action	None	
051	X420 I XFERCREA %2 [%1] Inbound transfer created - destination=%3 file=%4	
Type	Information	
Meaning	Information message indicating the creation of an incoming transfer.	
Parameters	1	Local request identifier
	2	Identification of the sender
	3	Identification of the receiver
	4	Local file name
Action	None	
052	X420 I XFERSTART %2 [%1] Outbound transfer starting - originator=%3 file=%4	
Type	Information	
Meaning	Information message indicating the creation of an outgoing transfer.	

<i>Parameters</i>	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender
	4	Local file name
<i>Action</i>	None	

054 X420 I XFERSENT %2 [%1] Outbound transfer sent - originator=%3 file=%4

<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the successful end of an outgoing transfer	
<i>Parameters</i>	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender
	4	Local file name
<i>Action</i>	None	

057 X420 I XRNCREA %2 [%1] Inbound RN created - destination=%3

<i>Type</i>	Information	
<i>Meaning</i>	Information message indicating the creation of an incoming receipt notification.	
<i>Parameters</i>	1	Local request identifier
	2	Identification of the sender
	3	Identification of the receiver
<i>Action</i>	None	

058	X420 I XNRNCREA %2 [%1] Inbound NRN created - destination=%3	
Type	Information	
Meaning	Information message indicating the creation of an incoming non-receipt notification.	
Parameters	1	Local request identifier
	2	Identification of the sender
	3	Identification of the receiver
Action	None	
059	X420 I XRNSTART %2 [%1] Outbound RN starting - originator=%3	
Type	Information	
Meaning	Information message indicating the creation of an outgoing receipt notification.	
Parameters	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender
Action	None	
060	X420 I XNRNSTART %2 [%1] Outbound NRN starting - originator=%3	
Type	Information	
Meaning	Information message indicating the creation of an outgoing non-receipt notification.	
Parameters	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender

Action	None	
061	X420 I XNORSTART %2 [%1] Outbound NORN starting - originator=%3	
Type	Information	
Meaning	Information message indicating the creation of an outgoing no receipt notification.	
Parameters	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender
Action	None	
062	X420 I XRNSENT %2 [%1] Outbound RN sent - originator=%3	
Type	Information	
Meaning	Information message indicating the successful end of an outgoing receipt notification.	
Parameters	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender
Action	None	
063	X420 I XNRNSENT %2 [%1] Outbound NRN sent - originator=%3	
Type	Information	
Meaning	Information message indicating the successful end of an outgoing non-receipt notification.	

Parameters	1	Local request identifier
	2	Identification of the receiver
	3	Identification of the sender
Action	None	
101	X420 E ERROR ERROR from %1: %2	
Type	Error	
Meaning	Generic error message.	
Parameters	1	Location in the program
	2	Error message
Action	The error message, further log message and trace should hint about the issue. If you cannot resolve the problem, contact Axway Support and provide them with the information from the log message.	
102	X420 E INITFAIL X420 initialization failed	
Type	Error	
Meaning	The X.420 connector failed to initialize.	
Parameters	None	
Action	Check associated log messages, traces and configuration.	
201	X400 S TRACE TRACE from %1: %2	
Type	Success	
Meaning	Generic product trace.	
Parameters	1	Location in the program
	2	Trace message
Action	None	

202	X400 I INITING X400 Configuration Updater initializing...	
<i>Type</i>	Information	
<i>Meaning</i>	The configuration update process is being initialized.	
<i>Parameters</i>	None	
<i>Action</i>	None	
203	X400 I INITOK X400 Configuration Updater initialization OK	
<i>Type</i>	Information	
<i>Meaning</i>	The configuration update process has been initialized.	
<i>Parameters</i>	None	
<i>Action</i>	None	
204	X400 S CNFGENER X400 Configuration generation success to file : %1	
<i>Type</i>	Success	
<i>Meaning</i>	A specific configuration file has been generated successfully.	
<i>Parameters</i>	1	Local configuration file
<i>Action</i>	None	
205	X400 I CNFGENER X400 Configuration generation success	
<i>Type</i>	Information	
<i>Meaning</i>	A new configuration has been generated successfully.	
<i>Parameters</i>	None	
<i>Action</i>	None	

302	X400 E INITFAIL X400 Configuration Updater initialization failed	
Type	Error	
Meaning	The configuration update process failed to initialize.	
Parameters	None	
Action	Check associated log messages, traces and configuration.	
303	X400 W CNFMSIGN [%1] %2 : Ignored, pointed by no local user	
Type	Warning	
Meaning	The MSP7 entry is pointed to by no local user. Therefore it is ignored.	
Parameters	1	Item type: MSP7
	2	Item name
Action	Create or update a local user entry to point to that item. Delete the MSP7 entry if it is not used.	
304	X400 E CNFGEERR X400 Configuration generation failed to file : %1	
Type	Error	
Meaning	An X.400 configuration file could not be generated.	
Parameters	1	Local configuration file
Action	Check file permissions.	
305	X400 W CNFUSMTA [%1] %2 : Pointing to non existing MTA %3	
Type	Warning	
Meaning	The MTA entry set in the remote user definition does not exist.	

Parameters	1	Item type: Remote User
	2	Item name
	3	MTA name
Action	Change the MTA value in the remote user settings or create an MTA entry whose name matches the setting.	
306	X400 E CNFUSMS7 [%1] %2 : Pointing to non existing MSP7 %3	
Type	Error	
Meaning	The MSP7 entry set in the local user definition does not exist.	
Parameters	1	Item type : Remote User
	2	Item name
	3	MSP7 name
Action	Change the MSP7 value in the local user settings or create an MSP7 entry whose name matches the setting.	
307	X400 E CNFCLMTA [%1] %2 : Local MTA is not configured	
Type	Error	
Meaning	The local MTA is not set in the configuration.	
Parameters	1	Item type: Configuration
	2	Item name
Action	Configure the local MTA.	
308	X400 E CNFCDMTA [%1] %2 : Invalid DEFAULT_MTA_NAME = %3	
Type	Error	

<i>Meaning</i>	The default MTA value located in the configuration does not point to an existing MTA.	
<i>Parameters</i>	1	Item type: Configuration
	2	Item name
	3	MTA name
<i>Action</i>	Change the default MTA value in the configuration or create an MTA entry whose name matches the setting.	

309	X400 E CNFUSLOC [%1] %2 : No local X420 user defined	
<i>Type</i>	Error	
<i>Meaning</i>	No local X.420 user has been found.	
<i>Parameters</i>	1	Item type: User
	2	Item name
<i>Action</i>	Create an X.420 local user or turn a remote user into local.	

311	X400 E CHKINVEN [%1] %2 : Check failed. Invalid entry %3 = %4	
<i>Type</i>	Error	
<i>Meaning</i>	An item contains an entry which value has been checked as not being valid.	
<i>Parameters</i>	1	Item type: Configuration, MTA, MSP7 or User
	2	Item name
	3	Entry
	4	Entry value
<i>Action</i>	Fix the specific value.	

Axway Gateway: Connectors

Axway Gateway connectors

You can link Gateway to other Axway products and to external products. The set of elements that software applications use to communicate with each other are known as *connectors*. Below are the links to the sections detailing those connectors:

[Axway Messaging connector](#)

[Axway Sentinel connector](#)

[Axway Integrator connector](#)

[Axway PassPort PM connector](#)

[Axway PassPort PS connector](#)

[Axway PassPort AM connector](#)

[SWIFTNet connector](#)

[JMS connector](#)

[X.400 connector](#)

Axway Gateway: Connectors

Axway Messaging connector

[Axway Gateway and Axway Messaging](#)

[Operational overview](#)

[Axway Messaging connector](#)

[Linking Axway Gateway to Axway Messaging: Overview](#)

Axway Gateway and Axway Messaging

This section introduces Axway Messaging (formerly XMS) and the Axway Messaging connector in Gateway.

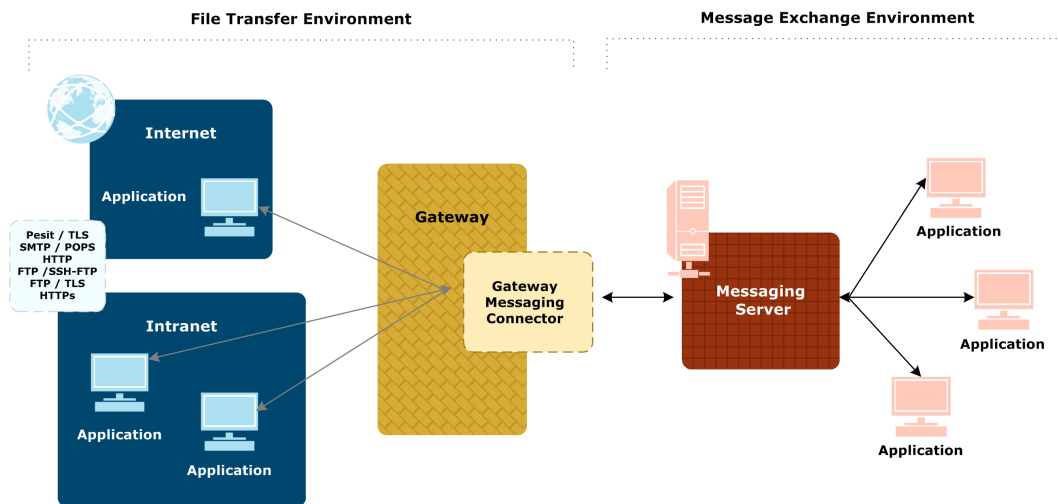
Gateway and the components that support Messaging were originally designed to work in different environments.

- **Gateway** operates in a file transfer environment, handling file exchanges via multiple standardized protocols across local networks, extranets and the Internet.

- **Messaging** is Message-Oriented Middleware. The Messaging components (servers, clients and administrative tools) provide an environment in which applications can exchange messages without directly communicating with each other.

The **Axway Messaging connector** provides you with the means to link these two environments, enabling you to:

- Transmit Messaging formatted application messages across your network (or across the Internet)
- Redirect any file received at the Gateway to a Messaging client in the form of an Axway Messaging message



Operational overview

From Gateway to Messaging

When Gateway is operating in *server/receiver* mode or *client/receiver* mode, it can transform and route any incoming file to one or more Messaging servers.

To do this, Gateway processes the file and the associated technical (transfer) parameters and uses the information to generate a Messaging format envelope. Gateway then redirects the enveloped file to a Messaging Server message queue.

From Messaging to Gateway

For Messaging formatted messages deposited to an Axway Messaging message queue, one or more Messaging servers can direct the messages to Gateway through the connector for routing via one of the protocols supported by Gateway.

To do this, Gateway:

- Acts as an application that *consumes* a message deposited to the Axway Messaging message queue
- Extracts and writes the message content to disk, and extracts the information necessary for the message transfer
- Forwards the reformatted message to the target application via a Gateway Transfer Request

Axway Messaging connector

Connectors

An *Axway Messaging connector* is a group of technical components that enables Gateway and Messaging to exchange data. You define connectors using either the GUI or the command line interface. Each connector operates in a single direction either from the Gateway to a Messaging server queue or vice versa. For each Messaging server it is typical (but not mandatory) to create three connectors, one each for:

- Messages from the Messaging server to Gateway
- Messages from Gateway to the Messaging server message queue
- Messages from Gateway to the Messaging server error message queue

Property Lists

In the Axway Messaging environment, each message contains the following parts:

- *Body*: contains the information that the sending application sends to the receiving application
- *Header*: contains attributes that provide details about the message. Sending applications specify the values of these predefined attributes. Among the details in the header are the following:
 - Names of the sending and receiving applications
 - Names of the sending and receiving servers
- *Header extension*: contains properties that provide additional details about the message. Properties *extend* a header with application-defined properties. Sending applications define both the names and the values of properties. Because properties are optional, header extensions can be empty

A Gateway **Property List** contains a description of the *header extension* part of the Axway Messaging message. For each field of the *header extension* the Property List contains a name and data type.

For files received at the Gateway, and routed to a Messaging queue, Gateway uses the Property List to provide information about the current transfer to the application that is the destination of the Axway Messaging message.

For files transferred from a Messaging server queue, and handled by the Gateway for routing via a standardized protocol, the Property List provides the technical parameters that enable Gateway to reroute the message contents.

Define Property Lists using either the GUI or the command line interface.

XMS-type Decision Rules

Once you have created an Axway Messaging connector, Gateway provides a special category of **Decision Rules** that can link an Axway Messaging message reception event to a file transfer via the Gateway.

Define *XMS-type Decision Rules* using either the GUI or the command line interface.

XMS Models

After creating a Property List, you can create an **XMS-type Model** that uses attributes defined in the list. XMS Models can be one of two types:

- From Gateway
- To Gateway

To use the XMS Model, create a Decision Rule and specify the appropriate XMS-type Model in the rule.

Linking Axway Gateway to Axway Messaging: Overview

To link Gateway to Axway Messaging:

1. In the configuration menu, [activate the Axway Messaging connector](#).
2. Create a Property List
3. Create an XMS Model of the type *To* or *From* Gateway
4. Create an IN or OUT type connector
5. Create a Decision Rule: XMS-type for IN connectors and XFER_CHANGE_STATE for OUT connectors
6. Create a Rule Table: XMS-type for IN connectors and XFER_CHANGE_STATE for OUT connectors
7. Set the `p_libxmsrapi` environment variable to `fullpath` in the `xmsrapi` library. On UNIX it is called `libxmsrapi.so` and on Windows `xmsrapi.dll`

[Linking Axway Gateway to Axway Messaging](#)

[Linking Axway Gateway to Axway Messaging \(command line\)](#)

[Working with Property Lists](#)

Axway Messaging connector

[Axway Gateway and Axway Messaging](#)

[Operational overview](#)

[Axway Messaging connector](#)

[Linking Axway Gateway to Axway Messaging: Overview](#)

Axway Gateway and Axway Messaging

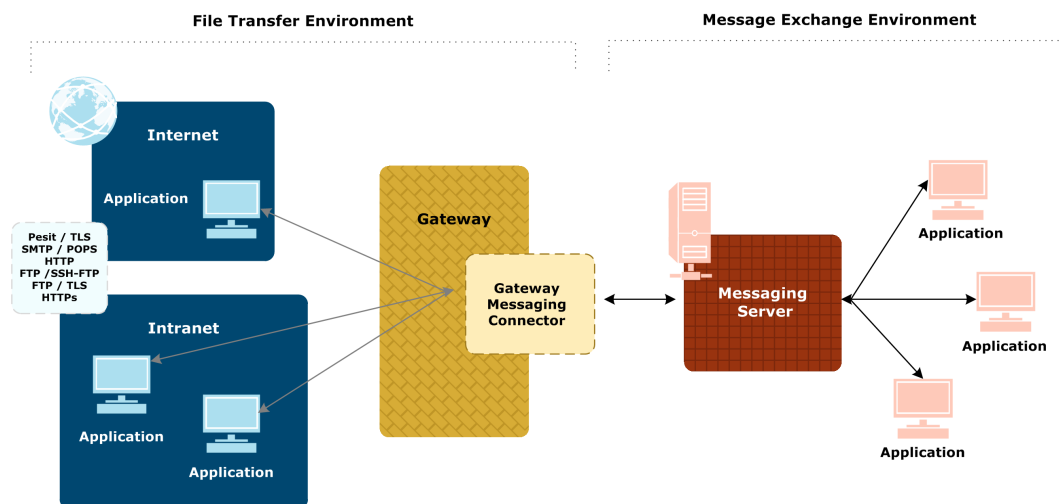
This section introduces Axway Messaging (formerly XMS) and the Axway Messaging connector in Gateway.

Gateway and the components that support Messaging were originally designed to work in different environments.

- **Gateway** operates in a file transfer environment, handling file exchanges via multiple standardized protocols across local networks, extranets and the Internet.
- **Messaging** is Message-Oriented Middleware. The Messaging components (servers, clients and administrative tools) provide an environment in which applications can exchange messages without directly communicating with each other.

The **Axway Messaging connector** provides you with the means to link these two environments, enabling you to:

- Transmit Messaging formatted application messages across your network (or across the Internet)
- Redirect any file received at the Gateway to a Messaging client in the form of an Axway Messaging message



Operational overview

From Gateway to Messaging

When Gateway is operating in *server/receiver* mode or *client/receiver* mode, it can transform and route any incoming file to one or more Messaging servers.

To do this, Gateway processes the file and the associated technical (transfer) parameters and uses the information to generate a Messaging format envelope. Gateway then redirects the enveloped file to a Messaging Server message queue.

From Messaging to Gateway

For Messaging formatted messages deposited to an Axway Messaging message queue, one or more Messaging servers can direct the messages to Gateway through the connector for routing via one of the protocols supported by Gateway.

To do this, Gateway:

- Acts as an application that *consumes* a message deposited to the Axway Messaging message queue
- Extracts and writes the message content to disk, and extracts the information necessary for the message transfer
- Forwards the reformatted message to the target application via a Gateway Transfer Request

Axway Messaging connector

Connectors

An *Axway Messaging connector* is a group of technical components that enables Gateway and Messaging to exchange data. You define connectors using either the GUI or the command line interface. Each connector operates in a single direction either from the Gateway to a Messaging server queue or vice versa. For each Messaging server it is typical (but not mandatory) to create three connectors, one each for:

- Messages from the Messaging server to Gateway
- Messages from Gateway to the Messaging server message queue
- Messages from Gateway to the Messaging server error message queue

Property Lists

In the Axway Messaging environment, each message contains the following parts:

- *Body*: contains the information that the sending application sends to the receiving application
- *Header*: contains attributes that provide details about the message. Sending applications specify the values of these predefined attributes. Among the details in the header are the following:
 - Names of the sending and receiving applications
 - Names of the sending and receiving servers
- *Header extension*: contains properties that provide additional details about the message. Properties *extend* a header with application-defined properties. Sending applications define both the names and the values of properties. Because properties are optional, header extensions can be empty

A Gateway **Property List** contains a description of the *header extension* part of the Axway Messaging message. For each field of the *header extension* the Property List contains a name and data type.

For files received at the Gateway, and routed to a Messaging queue, Gateway uses the Property List to provide information about the current transfer to the application that is the destination of the Axway Messaging message.

For files transferred from a Messaging server queue, and handled by the Gateway for routing via a standardized protocol, the Property List provides the technical parameters that enable Gateway to reroute the message contents.

Define Property Lists using either the GUI or the command line interface.

XMS-type Decision Rules

Once you have created an Axway Messaging connector, Gateway provides a special category of **Decision Rules** that can link an Axway Messaging message reception event to a file transfer via the Gateway.

Define *XMS-type Decision Rules* using either the GUI or the command line interface.

XMS Models

After creating a Property List, you can create an **XMS-type Model** that uses attributes defined in the list. XMS Models can be one of two types:

- From Gateway
- To Gateway

To use the XMS Model, create a Decision Rule and specify the appropriate XMS-type Model in the rule.

Linking Axway Gateway to Axway Messaging: Overview

To link Gateway to Axway Messaging:

1. In the configuration menu, [activate the Axway Messaging connector](#).
2. Create a Property List

3. Create an XMS Model of the type *To* or *From* Gateway
4. Create an IN or OUT type connector
5. Create a Decision Rule: XMS-type for IN connectors and XFER_CHANGE_STATE for OUT connectors
6. Create a Rule Table: XMS-type for IN connectors and XFER_CHANGE_STATE for OUT connectors
7. Set the `p_libxmsrapi` environment variable to `fullpath` in the `xmsrapi` library. On UNIX it is called `libxmsrapi.so` and on Windows `xmsrapi.dll`

[Linking Axway Gateway to Axway Messaging](#)

[Linking Axway Gateway to Axway Messaging \(command line\)](#)

[Working with Property Lists](#)

Axway Gateway: Connectors

Linking Axway Gateway to Axway Messaging

[How to link Axway Gateway and Axway Messaging](#)

[How to create a link from an Axway Messaging server to Axway Gateway](#)

[How to create a link from Axway Gateway to an Axway Messaging server](#)

How to link Axway Gateway and Axway Messaging

The Gateway GUI enables you to create and manage links between Gateway and Axway Messaging.

You can create Gateway/Messaging links that operate in either of two directions:

- [From an Axway Messaging server to a Gateway](#) (*IN* connector) outgoing Axway Messaging messages originating from a specified Axway Messaging server and queue and going to Gateway
- [From a Gateway to an Axway Messaging server](#) (*OUT* connector) incoming messages to be transmitted to a specified Axway Messaging server message queue from Gateway

How to create a link from an Axway Messaging server to Axway Gateway

To create a link from an Axway Messaging server to a Gateway:

1. Create a Property List that describes the messages received from the Axway Messaging server:
 - In the left pane of the GUI main window, click  to expand the nodes: **Transfer Management > Parameters > Model > XMS Model**.
 - Right-click **Property list** and select **New** from the context menu. The *New Property List* window is displayed.

- Complete the fields in the *New Property List* window.
- Click **Add** after each property that you define to include that property in the Property List.
- Click **OK** to confirm and create the new Property List.

Refer to [Creating a Property List](#).

2. Create an XMS Model for Transfers to the Gateway:
 - Select the Property List that you created in step 1.
 - Complete the **Name** field for the Model.
 - In the **Direction** field, select **To Gateway**.
 - Click **OK**. The second Model window is displayed.
 - Add the Model arguments and click **OK** to confirm.

Refer to [Working with XMS Models](#).

3. Create an *In* type Axway Messaging connector:
 - In the left pane of the GUI main window, click  to expand the nodes: **Event Management > XMS**.
 - Right-click **Connector In** and select **New**.
 - Complete the *XMS Connector In* window and click **OK** to confirm.

Refer to [Defining an In Connector](#).

4. Create an XMS-type Decision Rule:
 - In the left pane of the GUI main window, click  to expand the nodes: **Event Management > Scheduling**.
 - Right-click **New** and select **Decision Rule**.
The *New Decision Rule* window is displayed.
 - In the **Execution type** field, select **Model**.
 - In **Model**, link to the XMS Model created in Step 2.

Refer to [Working with Decision Rules](#).

5. Create an *XMS-type* Rule Table.

Refer to [Working with Rule Tables](#).

How to create a link from Axway Gateway to an Axway Messaging server

To create a connector from a Gateway to an Axway Messaging server:

1. Create a Property List that describes the messages received from the Gateway:

- In the left pane of the GUI main window, click  to expand the nodes: **Transfer Management > Parameters > Model > XMS Model**.
- Right-click **Property list** and select **New** from the context menu.
The *New Property List* window is displayed.
- Complete the fields in the *New Property List* window.
- Click **Add** after each property that you define to include that property in the Property List.
- Click **OK** to confirm and create the new Property List.

Refer to [Creating a Property List](#).

2. Create a Model for Transfers to the Axway Messaging server:

- Select the Property List that you created in step 1.
- Complete the **Name** field for the Model.
- In the **Direction** field, select **From Gateway**.
- Click **OK**. The second Model window is displayed.
- Add the Model arguments and click **OK** to confirm.

Refer to [Working with XMS Models](#).

3. Create an *Out* type Axway Messaging connector:

- In the left pane of the GUI main window, click  to expand the nodes: **Event Management > XMS**.
- Right-click **Connector Out** and select **New**.
- Complete the *XMS Connector Out* window and click **OK** to confirm.

Refer to [Defining an Out connector](#).

4. Create an XMS-type Decision Rule:

- In the left pane of the GUI main window, click  to expand the nodes: **Event Management > Scheduling**.
- Right-click **New** and select **Decision Rule**.
The *New Decision Rule* window is displayed.
- In the **Execution type** field, select **To XMS**.
- In **Model**, link to the XMS Model created in Step 2.
- In **XMS Connector out**, select the Axway Messaging connector created in Step 3.

Refer to [Working with Decision Rules](#).

5. Create a *Transfer Change State* Rule Table.

Refer to [Working with Rule Tables](#).

Linking Axway Gateway to Axway Messaging (command line)

[How to link Axway Gateway and Axway Messaging](#)

[Details of Axway Gateway/Axway Messaging link management commands](#)

[Creating an Axway Messaging connector](#)

[Creating an Axway Messaging connector Property List](#)

[Updating an Axway Messaging connector Property List](#)

[Deleting an Axway Messaging connector](#)

[Displaying an Axway Messaging connector Property List](#)

[Displaying all Axway Messaging connector Property Lists](#)

How to link Axway Gateway and Axway Messaging

You can create Gateway/Axway Messaging links that operate in either of two directions:

- From an Axway Messaging server to Gateway (*IN* connector)
- From Gateway to an Axway Messaging server (*OUT* connector)

How to create a link from an Axway Messaging server to Gateway

To create a link from an Axway Messaging server to Gateway:

1. Create a proplist (Property List) that describes the messages received from the Axway Messaging server. Use the `peladm create_proplist` command.

Example:

```
peladm create_proplist -pl list1 -ap "xms_ori:string;xms_dest:string"
```

2. Create a Model for Transfers to Gateway with the following characteristics:

- `TO_GTW` value for the protocol (`-pr`) attribute
- Specify a Property list in the `prop_list_name` (`-pln`) attribute

Use the [peladm create_model](#) command.

Example:

```
peladm create_model -ml mod_to_gtw -pr TO_GTW -pln list1 -oa "&(xms_ori)"  
-da "&(xms_dest)" -ty TRANS -dir O -m I -ap DEFAULT_B
```

3. Create an *IN* type Axway Messaging connector. Use the `peladm create_xmsctor` command.

Example:

```
peladm create_xmsctor -n xmsCtor1 -t IN -a Y -host mypc -p 4569 -i USER1  
-pwd USER1 -rqn Q1 -mt MSG -rd c:\temp\recpXMS
```

4. Create a Decision Rule with the following characteristics:

- *XMS* value for type_of_decision rule type (-type) attribute
- *MODEL* value for the execution_type (-et) attribute
- Specify an XMS-type Model for file routing via Gateway
- Specify a Property list in the prop_list_name (-pln) attribute

Use the [peladm create decisionrule](#) command.

Example:

```
peladm create_decisionrule -n DT_TO_GTW -type XMS -pln list1 -ca "xms_  
dest=SitDest" -et MODEL -ml mod_to_gtw
```

5. Create an XMS-type Rule Table. Use the [peladm create ruletable](#) command.

Example:

```
peladm create_ruletable -n RT_TO_GTW -type XMS -pln list1 xcn xmsCtor1 -a  
"DR_TO_GTW"
```

How to create a link from Gateway to an Axway Messaging server

To create a link from Gateway to an Axway Messaging server:

1. Create a proplist (Property List) that describes the messages received from Gateway. Use the [peladm create_proplist](#) command.

Example:

```
peladm create_proplist -pl list1 -ap "xms_ori:string;xms_dest:string"
```

2. Create a Model for Transfers to the Axway Messaging server.

- *FROM_GTW* value for the protocol (-pr) attribute
- Specify a Property list in the prop_list_name (-pln) attribute

Use the [peladm create model](#) command.

Example:

```
peladm create_model -ml mod_from_gtw -pr FROM_GTW -pln list1 -ma "xms_  
ori=&(org_alias); xms_dest=&(dest_alias); DestinationQmgr=QMGR412;  
DestiationQueue=FromGtw"
```

3. Create an *OUT* type Axway Messaging connector. Use the [peladm create_xmsctor](#) command.

Example:

```
peladm create_xmsctor -n routeCtor1 -t OUT -a Y -hos mypc -p 4569 -i  
USER1 -pwd USER1
```

4. Create an XMS-type Decision Rule, with the following characteristics:

- ***XFER_CHANGE_STATE*** value for the **type** attribute
- *TO_XMS* value for the execution type (-et) value
- specifies an XMS-type Model for routing to an Axway Messaging message queue.

Use the [peladm create decisionrule](#) command.

Example:

```
peladm create_decisionrule -n DR_FROM_GTW -type XFER_CHANGE_STATE -xcn ROUTECTOR1 -et TO_XMS -ml mod_from_gtw -ca \"direction=I;protocol=PEL\"
```

5. Create a Transfer Change State-type Rule Table. Use the [peladm create ruletable](#) command.

Example:

```
peladm create_ruletable -n RT_FROM_GTW -type XFER_CHANGE_STATE -a "DR_FROM_GTW"
```

Details of Axway Gateway / Axway Messaging link management commands

The following online commands are available for managing links between Gateway and an Axway Messaging server:

- [peladm create xmsctor](#)
- [peladm update xmsctor](#)
- [peladm delete xmsctor](#)
- [peladm create proplist](#)
- [peladm update proplist](#)
- [peladm delete proplist](#)
- [peldsp display xmsctor](#)
- [peldsp select xmsctor](#)
- [peldsp display proplist](#)
- [peldsp select proplist](#)

Creating an Axway Messaging connector: peladm create_xmsctor

Syntax	<code>peladm create_xmsctor [parameters]</code>

Description	Use this command to create an Axway Messaging connector. This command creates a connector for either: • outgoing Axway Messaging messages originating from a specified Axway Messaging server and queue • incoming messages to be transmitted to a specified Axway Messaging server message queue Repeat the command for each required connector creation.

Parameters	<code>-name (-n)</code> : Enter a name for the Axway Messaging connector.
	<code>-type (-t)</code> : Enter one of the following values to indicate the Axway Messaging connector type: • IN: for transmitting incoming file transfers to an Axway Messaging server message queue • OUT: for transmitting the messages of an Axway Messaging server message queue to a file transfer application via a Gateway Transfer Request
	<code>-active (-a)</code> : Enter one of the following values: • Y (yes) = activate the connector • N (no): = (default) deactivate the connector
	<code>-hostname (-host)</code> : Enter the name or the IP address of the Axway Messaging host machine to which Gateway is attached.
	<code>-port (-p)</code> : Enter the TCP access port of the Axway Messaging host machine.
	<code>-ident (-i)</code> : Enter the Gateway login identity for Axway Messaging host login.
	<code>-password (-pwd)</code> : Enter the password for Axway Messaging host login.
	<code>-log_level (-ll)</code> : Enter the required logging level: • 0 = no logging • 1 = short logs • 2 = full logs
	<code>-backup_messages (-bm)</code> : For incoming messages, to save a copy of messages filtered and not handled by Decision Rules and Rule Tables in the Axway Messaging file [<i>XMSReceptionQueue</i>] <i>bck</i> , enter one of the following values: • Y = The message is saved in the backup file • N = The message is not saved
	<code>-reception_queue_name (-rqn)</code> : For outgoing messages, enter the name of the Axway Messaging queue in which Gateway reads messages.
	<code>-message_type (-mt)</code> : For outgoing messages, enter one of the following values: • MSG = Axway Messaging message handling • ERROR_ACK = error and acknowledgment handling
	<code>-reception_directory (-rd)</code> : For outgoing messages, enter the directory in which Gateway stores Axway Messaging data content to be sent.
	<code>-message_size_max (-msm)</code> : Enter the maximum message size to be handled.

	<code>-compression (-c)</code>: For incoming files transferred to Axway Messaging, enter one of the following values: • Y = The message is compressed • N = (default) The message is not compressed <code>-delivery_mode (-dm)</code>: For incoming files transferred to Axway Messaging, enter one of the following values: • PERSISTENT = Persistent mode. Until the receiving application consumes the message, the receiving server stores a copy of the message in the Axway Messaging Log. If the connection between the receiving application and the receiving server fails, the receiving server does not lose the message. • NO_PERSISTENT = Non-persistent mode. The receiving server does not store a copy of the message in the Axway Messaging Log. If the connection between the receiving application and the receiving server fails, the receiving server loses the message. <code>-priority (-pr)</code>: For incoming files, enter one of the following values to indicate the priority an application can use as a criteria for getting messages from a source: • LOW = XMS_LOW_PRIORITY: The message has low priority • MEDIUM = XMS_MEDIUM_PRIORITY: The message has intermediate priority • HIGH = XMS_HIGH_PRIORITY: The message has high priority
Example	Enter the command: <pre>peladm create_xmsctor -n routeCtor1 -t OUT -a Y -hos mypc -p 4569 -i USER1 -pwd USER1</pre> Gateway creates and activates an OUT Axway Messaging connector with the properties: • name = <i>routeCtor1</i> • host name = <i>mypc</i> • host TCP port = <i>4569</i> • user ID / User password = <i>USER1 / USER1</i>

Updating an Axway Messaging connector: `peladm update_xmsctor`

Syntax	<code>peladm update_xmsctor [parameters]</code>
Description	Use this command to update parameter values of an existing Axway Messaging connector.

Parameters	<i>Mandatory</i>
	-name (-n): Enter the name of the Axway Messaging connector for which you want to update parameter values.
	-active (-a): Enter one of the following values: • Y (yes) = activate the connector • N (no): = deactivate the connector
	-hostname (-host): Enter the name or the IP address of the Axway Messaging host machine to which Gateway is attached.
	-port (-p): Enter the TCP access port of the Axway Messaging host machine.
	-ident (-i): Enter the Gateway login identity for Axway Messaging host login.
	-password (-pwd): Enter the password for Axway Messaging host login.
	-log_level (-ll): Enter the required logging level: • 0 = no logging • 1 = short logs • 2 = full logs
	-backup_messages (-bm): For outgoing messages, to save a copy of messages selected via Decision Rules and Rule Tables in the Axway Messaging file <i>XMSReceptionQueue bck</i> , enter one of the following values: • Y = The message is saved in the backup file • N = The message is not saved
	-reception_queue_name (-rqn): For outgoing messages, enter the name of the Axway Messaging queue in which Gateway reads messages.
	-message_type (-mt): For outgoing messages, enter one of the following values: • MSG = Axway Messaging message handling • ERROR_ACK = error and acknowledgement handling
	-reception_directory (-rd): For outgoing messages, enter the directory in which Gateway stores Axway Messaging data content to be sent.
	-message_size_max (-msm): Enter the maximum message size to be handled.
	-compression (-c): For incoming files transferred to Axway Messaging, enter one of the following values: • Y = The message is compressed • N = The message is not compressed

	-delivery_mode (-dm): For incoming files transferred to Axway Messaging, enter one of the following values: • PERSISTENT = Persistent mode. Until the receiving application consumes the message, the receiving server stores a copy of the message in the Axway Messaging Log. If the connection between the receiving application and the receiving server fails, the receiving server does not lose the message. • NO_PERSISTENT = Non-persistent mode. The receiving server does not store a copy of the message in the Axway Messaging Log. If the connection between the receiving application and the receiving server fails, the receiving server loses the message. -priority (-pr): For incoming files, enter one of the following values to indicate the priority an application can use as a criteria for getting messages from a source: • LOW = XMS_LOW_PRIORITY: The message has low priority • MEDIUM = XMS_MEDIUM_PRIORITY: The message has intermediate priority • HIGH = XMS_HIGH_PRIORITY: The message has high priority
Example	Enter the command: <pre>peladm update_xmsctor -n xmsCtor1 -rd c:\temp\recepXMS_2</pre> Gateway sets the reception directory attribute of the Axway Messaging connector named <i>xmsCtor1</i> to <i>c:\temp\recepXMS_2</i>.

Deleting an Axway Messaging connector: *peladm delete_xmsctor*

Syntax	<code>peladm delete_xmsctor [-name]</code>
Description	Use this command to delete an Axway Messaging connector. Repeat the command for each connector you want to delete.
Parameter	<i>Mandatory</i> <code>-name (-n)</code>: Enter the name of the Axway Messaging connector you want to delete.
Example	Enter the command: <pre>peladm delete_xmsctor -n xmsCtor1</pre> Gateway deletes the Axway Messaging connector named <i>xmsCtor1</i>.

Creating an Axway Messaging connector Property List: *peladm create_proplist*

Syntax	<code>peladm create_proplist [-properties_list] [-comments] [-add_prop]</code>
Description	Use this command to create a new Property List. Repeat the command for each Property List that you want to create.
Parameters	<code>-properties_list (-pl)</code> : Enter the name of the Axway Messaging connector Property List you want to create.
	<code>-comments (-cts)</code> : Enter a free text description. Maximum: 80 alphanumeric characters.
	<code>-add_prop (-ap)</code> : Add one or more properties to the list, in the form <code>prop1:type1;prop2:type2;propN:typeN</code>
Example	Enter the command: <pre>peladm create_proplist -pl list1 -ap "xms_ ori:int;xms_dest:ext"</pre> Gateway creates the Property List named <i>list1</i> .

Updating an Axway Messaging connector Property List: *peladm update_proplist*

Syntax	<code>peladm update_proplist [-properties_list] [-comments] [-add_prop] [-add_prop] [-update_prop] [-delete_prop]</code>
Description	Use this command to update an existing Axway Messaging connector Property List. Repeat the command for each Property List you want to update.
Parameters	<code>-properties_list (-pl)</code> : Enter the name of the Axway Messaging connector Property List you want to update.
	<code>-comments (-cts)</code> : Enter a free text description. Maximum: 80 alphanumeric characters.
	<code>-add_prop (-ap)</code> : Add one or more properties to the list, using the form: <code>prop1:type1;prop2:type2;propN:typeN</code>
	<code>-update_prop (-up)</code> : Update one or more properties in an existing list, using the form: <code>prop1:type1;prop2:type2;propN:typeN</code>
	<code>-delete_prop (-dp)</code> : Delete one or more properties from an existing list, using the form: <code>prop1;prop2;propN</code>

Example	Enter the command: <pre>peladm update_proplist -pl list1 -up "xms_ori:int"</pre> Gateway sets the <code>xms_ori</code> attribute of the <code>list1</code> Property List to an <code>int</code> data type value.

Deleting an Axway Messaging connector Property List: `peladm delete_proplist`

Syntax	<code>peladm delete_proplist [-properties_list]</code>
Description	Use this command to delete an Axway Messaging connector Property List. Repeat the command for each Property List that you want to delete.
Parameters	<code>-properties_list (-pl)</code> : Enter the name of the existing Axway Messaging connector Property List you want to delete.
Example	Enter the command: <pre>peladm delete_proplist -pl list1</pre> Gateway deletes the Property List named <code>list1</code> .

Displaying an Axway Messaging connector: `peldsp display_xmsctor`

Syntax	<code>peldsp display_xmsctor [-name]</code>
Description	Use this command to display an Axway Messaging connector. Repeat the command for each connector for which you want to view properties.
Parameters	<code>-name (-n)</code> : Enter the name of the existing Axway Messaging connector you want to display.
Example	Enter the command: <pre>peldsp display_xmsctor -n xmsCtor1</pre> Gateway displays the properties of the Axway Messaging connector named <code>xmsCtor1</code> .

Selecting an Axway Messaging connector: `peldsp select_xmsctor`

Syntax	<code>peldsp select_xmsctor [-name]</code>
Description	Use this command to select an Axway Messaging connector. Repeat the command for each connector you want to select.

Parameters	<code>-name (-n)</code> : Enter the name of the existing Axway Messaging connector for which you want to display characteristics.
	<code>-type (-t)</code> : Enter one of the following connector types as a selection criteria: • IN: incoming file transfers to an Axway Messaging server message queue • OUT: outgoing file transfers from an Axway Messaging server
	<code>-active (-a)</code> : Enter one of the following activity states as a selection criteria: • Y (yes) = connector in active state • N (no): = connector in inactivate state
	<code>-reception_queue_name (-rqn)</code> : Enter the name of an Axway Messaging server reception queue.
Example	Enter the command: <pre>peldsp select_xmsctor</pre> Gateway displays a list of all Axway Messaging connectors.

Displaying an Axway Messaging connector Property List: peldsp display_proplist

Syntax	<code>peldsp display_proplist [-properties_list]</code>
Description	Use this command to display a Property List.
Parameters	<code>-properties_list (-pl)</code> : Enter the name of the Property List whose properties you want to display.
Example	Enter the command: <pre>peldsp display_proplist -pl list1</pre> Gateway displays the properties of the Property List named <i>list1</i> .

Displaying all Axway Messaging connector Property Lists: peldsp select_proplist

Syntax	<code>peldsp select_proplist [-properties_list]</code>
Description	Use this command to select one or more Property Lists for display.
Parameters	<code>-properties_list (-pl)</code> : Enter the names of the Property Lists you want to display.
Example	Enter the command: <pre>peldsp select_proplist</pre> Gateway displays a list of the names of all Property Lists.

Axway Sentinel

Axway Sentinel is a Business Activity Monitoring (BAM) product, part of the Axway Platform. Sentinel provides you with a set of tools that enable you to view and react to events that occur in your network. You can use Sentinel to collect information about targeted platforms and business and technical process events. Sentinel enables you to interpret and react to this information, applying adapted solutions both in real-time and according to a pre-programmed schedule.

If your Axway product package includes both Gateway and Sentinel, you can configure Sentinel to monitor Gateway transfer activities:

- Transfer state changes
- Logs

Before using Sentinel, it is recommended that you become familiar with the main concepts and the user interfaces of Sentinel.

[Configuring Gateway for Sentinel monitoring](#)

[XFBTransfer Tracked Object](#)

[XFBTransfer Sentinel/Gateway correspondence](#)

[XFBLog Tracked Object](#)

[Monitoring Transfers](#)

[Managing .pmq files with the pmqctl command](#)

More information

For more information on Sentinel architecture and concepts, refer to the *Sentinel* documentation.

Axway Gateway: Connectors

Configuring Gateway for Sentinel monitoring

[Prerequisites](#)

[Configuring Gateway for use with Sentinel](#)

Prerequisites

The Axway Platform installation DVD includes three files that contain descriptions of the Sentinel objects that you use for monitoring Gateway from Sentinel. These files are located on the installation DVD in the `/extras/sentinel` folder:

- XFB_ALL_EN.xml (for non-High Availability solutions)
- XFB_ALL_EN_HA.xml (for SWIFTNet High Availability solutions)
- XFB_ALL_PB_EN.xml

The two files XFB_ALL_EN.xml and XFB_ALL_PB_EN.xml are identical except for some request objects that use a view request scope set to "Private" in the first of these two files and "Public" in the second one.

The view request scope defines the way Sentinel manages the access to requests, either public or based on a username/password pair using profiles created in Sentinel. If PassPort AM is used together with Sentinel then the profiles used will be those from PassPort AM.

Before you can use Sentinel to monitor Gateway events, you must import one of these description files into Sentinel. If you use PassPort AM with Sentinel, use the second file. Otherwise use the first file.

Note: Sentinel does not currently support "Private" view request method for its objects. This is planned for a future release.

The two files: XFB_ALL_EN.xml and XFB_ALL_EN_HA.xml are identical except for the TrackingObject option "nullMngt", which has the value "Last" (the updates are done relative to the last update) for SWIFTNet High Availability. Therefore, if you are not using the file provided in the delivery and generate the Tracking Object, do not forget to enable the "null management" option when using it in the context of SWIFTNet High Availability.

Configuring Gateway for use with Sentinel

To configure Gateway for use with Sentinel, set up the Sentinel connector in the Gateway [configuration menu](#):

- In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
- Select **Connectivity > Axway Connectivity > Sentinel**.
- Complete the [Sentinel parameters](#).
- Click **Activate** to enable the connector.
- Click **OK**.

After you have activated the Sentinel connector in the configuration menu, Gateway automatically launches Sentinel notification at startup.

[About Axway Sentinel](#)

[Managing .pmq files with the pmqctl command](#)

Sentinel: XFBTransfer Tracked Object

[About XFBTransfer](#)

[XFBTransfer: Attributes](#)

[XFBTransfer: Predefined Requests](#)

About XFBTransfer

When you configure Sentinel to monitor Gateway, you can import the *XFBTransfer* Tracked Object. XFBTransfer stores and describes Tracked Instances of Gateway transfers. Each Tracked Instance describes a single step of a single transfer process. Taken together, the Tracked Instances in each Processing Cycle describe the transfer of a file or, alternatively, a message between two Gateway applications.

In this documentation, a Gateway file transfer is referred to as a *transfer*.

XFBTransfer Attributes

XFBTransfer includes attributes that you can use to identify:

- [Each step in the transfer process](#)
- [Transfer dates and times](#)
- [The roles of transfer partners](#)
- [Transfer protocols](#)
- [Senders and Receivers](#)
- [Transfer options](#)
- [Transfer monitors](#)
- [The size of transfers](#)
- [Transfer users](#)
- [The structure and content of transfers](#)
- [Transfers](#)
- [Details about SMTP/POP3 transfers](#)
- [Transfer validity periods](#)
- [Details about SWIFTNet transfers](#)

Identifying each step in the transfer process

For each step of each Gateway transfer, Gateway generates one Tracked Instance. In each Tracked Instance, the **State** attribute identifies the relevant step of the transfer process. The following table lists and describes the possible values of the **State** attribute for the Tracked Instances of a single Gateway transfer.

State	Details
Created	The Requester prepared a Transfer Request and updated the transfer catalog or, alternatively the transfer Mailbox.

State	Details
To_Execute	The Requester sent the Transfer Request to the Server .
Sending	The Sender is sending data.
Receiving	The Receiver is receiving data.
Sent	The Sender sent all of the transfer data.
Received	The Receiver received all of the transfer data.
Terminated	The Sender sent all of the transfer data, and the Receiver received all of the transfer data. Both partners successfully completed the transfer.

Taken together, the set of Tracked Instances that concern a single Gateway transfer are referred to as a *Processing Cycle*. To identify the Tracked Instances that belong to the same Processing Cycle, use the **CycleId** attribute.

Note: Not all States in Gateway have equivalent States in Axway Sentinel.

Identifying the roles of transfer partners

Although each Gateway transfer occurs between only two transfer partners, each transfer partner plays two roles:

- **Sender** or **Receiver**
- **Requester** or **Server**

The [preceding table](#) underscores these roles. Note that transfer partners are referred to as Requesters, Senders, and Receivers.

For a given transfer, only the following combinations of partner roles are possible:

- **Sender/Requester** and **Receiver/Server**
 The partner that **sent** the transfer **requested** the transfer.
 The partner that **received** the transfer **listened** for the Transfer Request.
- **Receiver/Requester** and **Sender/Server**
 The partner that **received** the transfer **requested** the transfer.
 The partner that **sent** the transfer **listened** for the Transfer Request.

In each Tracked Instance, the following attributes identify the roles of the transfer partner that generated the Tracked Instance.

Attribute	Data Type	Value
Direction	String	One of: • E: The Sender generated the Tracked Instance. Equivalent to Output (O) on Gateway. • R: The Receiver generated the Tracked Instance. Equivalent to Input (I) on Gateway.
IsServer	String	One of: • 1: The Sender or, alternatively the Receiver is a Server. • 0: The Sender or, alternatively the Receiver is a Requester.

Examples

When a transfer occurs between a **Sender/Requester** and a **Receiver/Server**, Sentinel receives the following Tracked Instances.

State	Direction	IsServer
Created	E (Sender)	0 (Requester)
To_Execute	E (Sender)	0 (Requester)
Sending	E (Sender)	0 (Requester)
Receiving	R (Receiver)	1 (Server)
Sent	E (Sender)	0 (Requester)
Received	R (Receiver)	1 (Server)

However, when a transfer occurs between a **Receiver/Requester** and a **Sender/Server**, Sentinel receives the following Tracked Instances.

State	Direction	IsServer
Created	R (Receiver)	0 (Requester)
To_Execute	R (Receiver)	0 (Requester)

State	Direction	IsServer
Sending	E (Sender)	1 (Server)
Receiving	R (Receiver)	0 (Requester)
Sent	E (Sender)	1 (Server)
Received	R (Receiver)	0 (Requester)

Identifying Senders and Receivers

Together, the [Direction](#) and [IsServer](#) attributes identify the [role](#) of a transfer partner in the transfer process. However, they do not explicitly identify partners. To explicitly identify transfer partners, use the following attributes:

Attribute	Data Type	Value
Site	String	Partner alias that the local partner uses to identify the parameter settings for the remote partner. When the value of the Direction attribute is: • R (Receiver), the Receiver is the local partner. The Receiver used the value of this attribute to identify the parameter settings for the Sender. • E (Sender), the Sender is the local partner. The Sender used the value of this attribute to identify the parameter settings for the Receiver.
ReceiverId	String	Network identifier of the Receiver
SenderId	String	Network identifier of the Sender
RAppl	String	Network identifier of the Receiver application
SAppl	String	Network identifier of the Sender application

Attribute	Data Type	Value
FinalReceiverId	String	If the Processing Cycle: - Describes a stored and forwarded transfer, the value of this attribute is the Network Identifier of the final Receiver of the transfer - Does not describe a stored and forwarded transfer, the value of this attribute is empty
OriginalSenderId	String	If the Processing Cycle: - Describes a stored and forwarded transfer, the value of this attribute is the Network Identifier of the initial Sender of the transfer - Does not describe a stored and forwarded transfer, the value of this attribute is empty

Identifying transfer monitors

Each Sender and Receiver is a logical entity that operates from a physical machine. Each machine is referred to as a *transfer monitor*. To identify details about transfer monitors, use the attributes that are listed in the following table. If the value of the [Direction](#) attribute is:

- E** (Sender), these attributes identify the Sender's transfer monitor
- R** (Receiver), these attributes identify the Receiver's transfer monitor

Attribute	Data Type	Value
Location	String	Unique and logical identifier of the transfer monitor.
Machine	String	Name of the Gateway application and platform that run on the transfer monitor.
Monitor	String	Application name: "GTW"
MonitorVersion	String	Version of the Gateway application that runs on the transfer monitor.

Identifying transfer users

Although Senders and Receivers perform all actions in the transfer process, they do not initiate these actions. Transfer users initiate transfer actions. Transfer users include both people and software applications. To monitor details about transfer users, use the following attributes.

Attribute	Data Type	Value
UserId	String	Local identifier of the user who owns the transferred file or message
GroupId	String	Local identifier of the group to which the transfer owner belongs
RequestUserId	String	Local identifier of the user who requested the transfer. If the user is: - also the transfer owner, the values of UserId and RequestUserId match - not the transfer owner, the values of UserId and RequestUserId do not match
RequestGroupId	String	Local identifier of the group to which the requesting user belongs
Trustee	String	If User Access Control is: - activated for the transfer, the value of this attribute is the local identifier of the transfer owner (UserId) - not activated for the transfer, the value of this attribute is the network identifier of the user who requested the transfer (RequestUserId)
RUser	String	Local identifier of the user who received the transfer
SUser	String	Local identifier of the user who sent the transfer

Identifying transfers

To identify a transfer, use the following attributes.

Attribute	Data Type	Value
IdAppl	String	If the value of the IsServer attribute is: • 0, the value of this attribute is the local identifier of the Requester • 1, the value of this attribute is empty
Application	String	Local Application.
FileName	String	If the value of the CommandType attribute is: • Message, this attribute is empty • File and the value of the Direction attribute is E (Sender), this attribute identifies the file from which the Sender retrieved the transfer data • File and the value of the Direction attribute is R (Receiver), this attribute identifies the file in which the Receiver recorded the transfer data
LocalId	String	If the value of the Direction attribute is: • E (Sender), the value of this attribute is the local transfer identifier for the Sender • R (Receiver), the value of this attribute is the local transfer identifier for the Receiver Senders and Receivers use local transfer identifiers to manage transfers (cancel, delete, and so on).

Attribute	Data Type	Value
ProtocolFileName	String	Network transfer identifier. The Sender sent this identifier to the Receiver.
ProtocolFileLabel	String	Network transfer name. The Sender sent this name to the Receiver. The Receiver can use this name to locally name the transfer.
ProtocolId	String	Network transfer identifier. The Sender generated this identifier. The Receiver acknowledged this identifier.
ProtocolMessage	String	If the value of the CommandType attribute is: • Message, the value of this attribute is the content of the transferred message • File, this attribute is empty
PositionNumber	Integer	Catalog transfer identifier. If the value of the Direction attribute is: • E (Sender), the value of this attribute identifies the transfer in the Sender's transfer catalog or transfer Mailbox • R (Receiver), the value of this attribute identifies the transfer in the Receiver's transfer catalog or transfer Mailbox
ProtocolParameter	String	Network transfer description. The Sender sent this description to the Receiver.

Attribute	Data Type	Value
UserParameter	String	Local transfer description. When the Requester recorded the transfer in the transfer catalog or transfer Mailbox, the Requester generated this description. The value of this attribute includes two parts: • The first part contains the first forty characters of the description • The second part contains the remaining characters of the description

Identifying transfer validity periods

Senders and Receivers can perform specific transfers only during the specified validity periods. To monitor the validity periods for transfers, use the following attributes.

Attribute	Data Type	Value
EarliestDate	Date	Date on which the validity period begins
EarliestTime	String	Time at which the validity period begins
LatestDate	Date	Date on which the validity period ends
LatestTime	String	Time at which the validity period ends

Identifying transfer dates and times

During the transfer process, Senders and Receivers record the time and date of certain actions that they perform. To monitor these details, use the following attributes.

Attribute	Date Type	Value
CreationDate	Date	By default, the system date on which the Sender sent the transfer. The Sender can set this date. The Receiver can filter transfers based on this date.
CreationTime	String	By default, the system time at which the Sender sent the transfer. The Sender can set this time. The Receiver can filter transfers based on this time.
SendDate	Date	If the value of the Direction attribute is: • E (Sender), the value of this attribute is the date on which the Sender recorded the transfer in the transfer catalog or transfer Mailbox • R (Receiver), the value of this attribute is the date on which the Receiver recorded the transfer in the transfer catalog or transfer Mailbox The Sender and the Receiver record each transfer only once.
SendTime	String	If the value of the Direction attribute is: • E (Sender), the value of this attribute is the local time at which the Sender recorded the transfer in the transfer catalog or transfer Mailbox • R (Receiver), the value of this attribute is the local time at which the Receiver recorded the transfer in the transfer catalog or transfer Mailbox The Sender and the Receiver record each transfer only once.

Attribute	Date Type	Value
AckDate	Date	Date on which the Receiver acknowledged the transfer. If the Receiver did not acknowledge the transfer, this attribute is empty.
AckTime	String	Time at which the Receiver acknowledged the transfer. If the Receiver did not acknowledge the transfer, this attribute is empty.
StartDate	Date	If the value of the State attribute is: • SENT, the value of this attribute is the date on which the Sender began sending the transfer • RECEIVED, the value of this attribute is the date on which the Receiver began receiving the transfer These dates are expressed in <i>dd.mm.yyyy</i> format.
StartTime	String	If the value of the State attribute is: • SENT, the value of this attribute is the local time at which the Sender began sending the transfer • RECEIVED, the value of this attribute is the local time at which the Receiver began receiving the transfer These times are expressed in <i>hh:mm:ss</i> format.

Attribute	Date Type	Value
EndDate	Date	If the value of the State attribute is: • SENT, the value of this attribute is the date on which the Sender stopped sending the transfer • RECEIVED, the value of this attribute is the date on which the Receiver stopped receiving the transfer These dates are expressed in <i>dd.mm.yyyy</i> format.
EndTime	String	If the value of the State attribute is: • SENT, the value of this attribute is the local time at which the Sender stopped sending the transfer • RECEIVED, the value of this attribute is the local time at which the Receiver stopped receiving the transfer These times are expressed in <i>hh:mm:ss</i> format.
TransmissionDuration	String	If the value of the State attribute is: • SENT, the value of this attribute is the time that was required to send the transfer • RECEIVED, the value of this attribute is the time that was required to receive the transfer These times are expressed in seconds.

Identifying transfer protocols

Most transfers are associated with three protocol layers:

- Network Layer: rules that manage transfer media
- SSL/TLS Layer (Security Sockets Layer / Transport Layer Security): rules that manage transfer security
- Protocol Layer: rules that manage transfer communication

To monitor details about the Protocol and SSL/TLS Layers, use the following attributes:

Attribute	Data Type	Value
Protocol	String	Name of the protocol that operates at the Protocol Layer of the transfer. Possible values: • PSIT_HS_E (<i>PESIT, version E</i>) • PSIT_HS_D (<i>PESIT, version D</i>) • PEL • ODT (<i>Odette File Transfer Protocol</i>) • FTP • HTTP • FTP_HTTP • SFTP • POP3 • SMTP • AS1 • AS2 • AS3 • RN_POP3 • RN_SMTP • RN_HTTP • SWIFTNET • JMS
IsSSL	String	One of: • 1: SSL/TLS protected the transfer. • 0: SSL/TLS did not protect the transfer.

Attribute	Data Type	Value
SSLAuth	String	One of: • S: The Server sent X.509 certificates to the Requester. • B: <i>Both</i> the Server and the Requester sent X.509 certificates to each other. • N: <i>Neither</i> the Server nor the Requester sent X.509 certificates.
SSLCypher	String	The cipher suite that the Server and the Requester used during the SSL/TLS session. The cipher suite identifies the following: • Authentication method • Encryption algorithm • Hash algorithm for MAC calculation

Identifying transfer options

To monitor transfer options, use the following attributes.

Attribute	Data Type	Value
Compression	String	One of the following compression options: • 0: Undefined • 1: Horizontal • 2: Vertical • 3: Both horizontal and vertical • 4: Not compressed

Attribute	Data Type	Value
EOTProcedure	String	When the value of the State attribute is: • SENT, the value of this attribute is the name of the EOT (End of Transfer Procedure) that the Sender executed • RECEIVED, the value of this attribute is the name of the EOT (End of Transfer Procedure) that the Receiver executed
Priority	Integer	Transfer priority. Receivers process transfers in the order of their priority. The range of possible values for this attribute is zero to 255. The lowest priority is zero. The highest priority is 255.
RetryMaxNumber	Integer	Maximum number of times that the Sender can attempt to send transfers.
RetryNumber	Integer	Number of times that the Sender attempted to send the transfer. Each time the Sender established a connection with the Receiver, the Sender counted one attempt.
RequestType	String	One of: • S: The Sender sent a <i>single</i> transfer to a single Receiver. • G: The Sender sent a <i>group</i> of transfers to a single Receiver. For each transfer in the group, Gateway generated one Processing Cycle. • D: The Sender sent a single transfer to a group of Receivers (<i>diffusion</i>). For each Receiver in the group, Gateway generated one Processing Cycle.

Attribute	Data Type	Value
TransferType	String	One of: • G: The transfer belongs to a <i>group</i> of transfers that the Sender sent to a single Receiver. For each transfer in the group, Gateway generated one Processing Cycle. • D: The Receiver belongs to a group of Receivers to whom the Sender sent the transfer (<i>diffusion</i>). For each Receiver in the group, Gateway generated one Processing Cycle.

Identifying the size of transfers

To monitor the size of transferred files and messages, use the following attributes:

Attribute	Data Type	Value
FileSize	Integer	Size of the transferred file or message on the Sender's monitor, before compression. This size is expressed in bytes.
TransmittedBytes	Integer	Size of the transferred file or message on the Receiver's monitor, after decompression. This size is expressed in bytes.

Identifying the structure and content of transfers

To monitor the structure and content of transfers, use the following attributes:

Attribute	Data Type	Value
CommandType	String	One of: • Message (M): The transferred data is a message. Denotes InterAct for SWIFTNet transfers. • File (F): The transferred data is a file. Denotes FileAct for SWIFTNet transfers. • Receipt (R) • Directory (D): FTP list.
BlockSize	Integer	If the value of the CommandType attribute is: • Message, this attribute is empty • File and the Receiver's monitor is a main frame, the value of this attribute identifies where the Receiver stores the transferred data • File and the Receiver's monitor is not a main frame, this attribute is empty
FileOrganization	String	If the value of the CommandType attribute is: • Message, the value of this attribute is org_undefined • File, the value of this attribute is one of the following: ◦ org_sequential: The transferred data is not indexed. ◦ indexed: The transferred data is indexed. ◦ direct: The transferred data is assigned relative access.

Attribute	Data Type	Value
FileType	String	If the value of the CommandType attribute is: • Message, this attribute is empty • File, the value of this attribute is one of the following: ◦ fixed: The transferred data contains <i>fixed-length</i> records. ◦ variable: The transferred data contains <i>variable-length</i> records. ◦ undefined: The structure of the transferred data is unknown.
RecordNumber	Integer	If the value of the CommandType attribute is: • Message, this attribute is empty • File, the value of this attribute is the number of records in the transferred file
RecordSize	Integer	If the value of the CommandType attribute is: • Message, this attribute is empty • File and the value of the FileType attribute is fixed, the value of this attribute is the size of all records in the transferred file, expressed in bytes • File and the value of the FileType attribute is variable or undefined, the value of this attribute is the size of the largest record in the transferred file, expressed in bytes

Attribute	Data Type	Value
Transcoding	String	Character code of the transferred data: • A: ASCII • B: Binary • E: EBCDIC • T: A transcoding table modified the character code of the transfer • U: Undefined
TranslationTableId	String	Translation table identifier. When a transfer partner uses a translation table that is delivered with Gateway, the value of this parameter is INTERNAL .

Identifying details about SMTP/POP3 transfers

To identify details about the emails that Gateway transfers, use the following attributes.

Attribute	Data Type	Value
SmtpFrom	String	Email address in the <i>From</i> field (Sender)
SmtpTo	String	Email addresses in the <i>To</i> field (Receiver)
SmtpCc	String	Email addresses in the <i>CC</i> field (Copy)
SmtpBcc	String	Email addresses in the <i>BCC</i> field (Blind copy)
SmtpSubject	String	Contents of the <i>Subject</i> field
SmtpAckTo	String	Contents of the <i>Ack</i> field

Attribute	Data Type	Value
SmtmAef	String	One of the following: • 0: If Gateway: ◦ sent the relevant transfer to an SMTP server, Gateway did not format the associated email. The transfer was already formatted as an email. However, Gateway attached all related files. ◦ received the relevant transfer from a POP3 server, Gateway did not decode the associated email. However, Gateway extracted all file attachments • 1: If Gateway: ◦ sent the relevant transfer to an SMTP server, Gateway formatted the associated email and attached all related files ◦ received the relevant transfer from a POP3 server, Gateway decoded the associated email and extracted all file attachments

Identifying details about SWIFTNet transfers

To identify details about SWIFTNet transfers, use the following attributes.

Attribute	Data Type	Value
TradeCompressing	String	• Y • N

Attribute	Data Type	Value
TradeCompressionAlgo	String	Example values: • ZIP • ZLIB • GZIP
TradeDeliveryMode	String	• RT (RealTime) • SNF (Store-and-Forward)
TradeDestination	String	Distinguished Name of the Originator of the Transfer. For example, ou=Paris, o=Bank, o=swift.
TradeDestinationAlias	String	Gateway Trading Partner name. For example, SWIFT_REAL.
TradeDestinationRemote	String	• Y • N
TradeEncrypting	String	• Y • N
TradeFileEncoded	String	• Y • N
TradeFormat	String	• "SWIFTNET" (for SWIFTNet Protocol).
TradeKeyEncryptionAlgo	String	• RSA
TradeMessageId	String	SNL-defined (incoming) or Gateway-defined (outgoing) technical transfer ID.
TradeNonRepudiation	String	• Y • N
TradeOriginator	String	Distinguished Name of the Originator of the Transfer. For example, ou=Lyon, o=Bank, o=swift.

Attribute	Data Type	Value
TradeOriginatorAlias	String	Gateway Trading Partner name. For example, SWIFT_REAL_LOCAL.
TradeReceiptMicAlgo	String	Example values: • ZIP • ZLIB • GZIP
TradeRequestType	String	SWIFTNet RequestType. This is related to the name of Message Structure used. For example, "camt.006.001.03" refers to the "ReturnTransaction" message, as defined in ISO20022.
TradeService	String	SWIFTNet service used. For example, "swift.generic.fa".
TradeSigning	String	• Y • N
TradeSigningAlgo	String	
TradeState	String	• ACKED • TERMINATED • CANCELED • ROUTED • TO_EXECUTE • SENT • ENDED_TO_ACK • AVAILABLE

The following attributes have specific values for SWIFTNet:

Attribute	Data Type	Value
Protocol	String	• SWIFTNET
CommandType	String	• F (FileAct) • M (Interact/Message) • R (Receipt)
ProtocolId	String	Value of the SWIFTNet Protocol Sw:MsgId. Note: You can use the <code>peltrans -sw_message_id</code> command to modify this value.

XFBTransfer Predefined Requests

When you configure Sentinel, you can import a set of predefined XFBTransfer Requests into the Monitoring Interface. The following table describes these Requests. All of these Requests retrieve Tracked Instances from the Current Table of XFBTransfer.

This Request	Retrieves...
CurrentTransfers	All Tracked Instances from the Current Table of XFBTransfer.
CurrentTransfersToday	Tracked Instances from the Current Table of XFBTransfer that correspond to the current date.
CurrentTransfersAlert	Tracked Instances from the Current Table of XFBTransfer that correspond to an Alert.

Axway Gateway: Connectors

XFBTransfer Sentinel/Gateway correspondence

The table in this section identifies the correspondence between Sentinel XFBTransfer object attributes and Gateway transfer properties.

In Axway Sentinel, use XFBTransfer to monitor and to respond to Gateway file transfers. Each attribute of this Tracked Object corresponds, in most cases, to a single Gateway transfer property.

To use the names of Axway Sentinel attributes to identify the corresponding Gateway properties, refer to the table below.

- *Axway Sentinel*: This column displays the names of the XFBTransfer attributes in alphabetical order.
- *Axway Gateway*: For the attributes in the *Axway Sentinel* column, this column displays the names of the corresponding Gateway properties.

The XFBTransfer object describes all types of Axway Managed File Transfer. Therefore some XFBTransfer attributes do not correspond to any Gateway properties. Such XFBTransfer attributes are not listed in this table.

Axway Sentinel	Axway Gateway
Application	appli
BlockSize	block_size
CGateUserParameter1	cgate_user_param1
CGateUserParameter2	chanteuse
CiphEntityObjectId	ciphEntity.object_id
CiphEntityObjectType	ciphEntity.object_type
CommandType	Depends on type
Compression	compression
CreationDate	Date from hist_create_date
CreationTime	Time from hist_create_date
CycleId	this_cycle_id
Direction	direction
EarliestDate	Date from date_to_begin
EarliestTime	Time from date_to_begin
EndDate	Date from date_to_end
EndTime	Time from date_to_end
EotProcedure	end_xfer_scriptpath
Etb5AuthType	auth_type
Etb5CiphType	ciph_type
Etb5IdBank	id_banque

Axway Sentinel	Axway Gateway
Etb5IdBankSec	id_banque_sec
Etb5IdClient	id_client
Etb5IdClientSec	id_client_sec
Etb5MemoType	memo_type
Etb5Paraf_used	paraf_used
Etb5RSign1Used	rsign1_used
Etb5RSign2Used	rsign2_used
Etb5SealType	seal_type
Etb5SignType	sign_type
Etb5UserArea	userarea
FileName	Built from dir_path and file_component
FileOrganisation	file_org
FileSize	file_size
FileType	file_type
FinalReceiverId	destination
GroupName	group_name
InternalCycleId	this_extended_cycle_id
IsEnd	• 1 if state is ACKED or ENDED • 0 otherwise
IsPermanent	permanent
IsServer	mode
IsSSL	• 1 if tls_sprof is set • 0 otherwise
LatestDate	Date from date_to_end

Axway Sentinel	Axway Gateway
LatestTime	Time from <code>date_to_end</code>
LocalId	<code>local_ident</code>
Location	<code>local_site_alias</code>
Machine	"GTW-UNIX" or "GTW-NT"
Monitor	"GTW"
MonitorVersion	Version, plus SP number where applicable.
OriginalSenderId	<code>originator</code>
PositionNumber	<code>sequence</code>
Priority	<code>priority</code>
ProductIpAddr	<code>sentinel_product_ip_addr</code> (configuration parameter)
ProductName	<code>local_site_alias</code>
Protocol	<code>protocol</code>
ProtocolFileLabel	<code>file_label</code>
ProtocolFileName	<code>file_name</code>
ProtocolId	<code>xfer_ident</code>
ProtocolMessage	<code>rcv_msg</code> or <code>snd_msg</code> (if the transfer is of type message) Note: on PeSIT, only the message sent in transfer-related phases is visible in the tracked object updates sent to Sentinel
ProtocolParameter	<code>rcv_msg</code> or <code>snd_msg</code> (if the transfer is of type transfer). See note for ProtocolMessage above.
ProtocolSelectParameter	<code>rcv_msg</code> (if the transfer is: outgoing, PeSit or SWIFTNet, and of type transfer). See note for ProtocolMessage above.
RAppl	<code>rcv_appli</code>

Axway Sentinel	Axway Gateway
ReceiverId	remote_agent_ident or local_agent_ident
RecordFormat	xfer_rec_fmt
RecordNumber	rec_count
RecordSize	rec_len
RequestCreationDate	Date from date_create
RequestCreationTime	Time from date_create
RequestType	Depends on type
RequestUserId	username
RetryMaxNumber	retry_count_max
RetryNumber	retry_count
ReturnCode	last_end_reason, last_end_diag, and last_end_err
ReturnMessage	Message corresponding to last_end_reason
RUser	rcv_user
SAPpl	snd_appli
SendDate	date_create
SenderId	remote_agent_ident or local_agent_ident
SessionTag	session_tag
SignEntityObjectId	signEntity.object_id
SignEntityObjectType	signEntity.object_type
Site	remote_agent
SmtpAckTo	receipt_req_to
SmtpAef	attach_extract_file

Axway Sentinel	Axway Gateway
SmtpBcc	smtp_bcc
SmtpCc	smtp_cc
SmtpFrom	smtp_from
SmtpSubject	smtp_subject
SmtpTo	smtp_to
SSLAuth	• "Serveur" if tls_server_auth is set • "Both" if tls_server_auth and tls_client_auth are set • "None" otherwise
SSLCypher	tls_cipher_suite
StartDate	Date from date_to_begin
StartTime	Time from date_to_begin
State	Combination of: state,direction, mode,user_processed and route_state.
SUser	snd_user
SwPossibleDuplicateInd	sw_pd_indication
TpmDeaId	dea_id
TradeCompressing	trade_compress
TradeCompressionAlgo	trade_compress_algo
TradeDeliveryMode	trade_delivery_mode
TradeDestination	trade_dest
TradeDestinationAlias	trade_dest_alias
TradeDestinationRemote	trade_remote_destination
TradeEncrypting	trade_encrypt

Axway Sentinel	Axway Gateway
TradeEncryptionAlgo	trade_ciphering_algo
TradeFileEncoded	trade_file_encoded
TradeFormat	trade_format
TradeKeyEncryptionAlgo	trade_key_ciphering_algo
TradeMessageId	message_id
TradeNonRepudiation	trade_non_repudiation
TradeNotifQueue	trade_notif_queue
TradeOriginator	trade_org
TradeOriginatorAlias	trade_org_alias
TradeReceiptMessageId	trade_receipt_message_id
TradeReceiptMic	trade_receipt_mic
TradeReceiptMicAlgo	trade_compression_algo
TradeReceiptRecipient	trade_receipt_recipient
TradeReceiptSigningAlgo	trade_receipt_sign_algo
TradeReceiptSmimeContentType	trade_receipt_content_type
TradeReceiptSmimeSignature	trade_receipt_signature
TradeRequestType	trade_request_type
TradeService	trade_service
TradeSigning	trade_sign
TradeSigningAlgo	trade_sign_algo
TradeState	trade_state
TransferTag	local_ident
Transcoding	data_code, plus "T" if xfer_data_code is not equal to data_code, in other words when there is transcoding

Axway Sentinel	Axway Gateway
TransferType	Depends on type
TransmissionDuration	$f(date\_end - date\_begin)$
TransmittedBytes	x_bytes
UserId	username
UserParameter1	param1
UserParameter2	param2
UserProcessed	user_processed
UserState	user_state
VirtualDirName	dir_name

Axway Gateway: Connectors

Sentinel: XFBLLog Tracked Object

About XFBLLog

When you configure Sentinel to monitor Gateway, you can import the *XFBLLog* Tracked Object. XFBLLog describes the contents of a Gateway log file. This log stores traces of the events and errors that occur on a Gateway monitor.

XFBLLog Attributes

The XFBLLog includes the following attributes.

Attribute	Data Type	Value
ApplicationName	String	Name of the relevant parameter file. This name is defined in the Gateway configuration file.
Product	String	Name of a Gateway product . This name depends on the operating system that you use.
IdentMsg	String	String that identifies a message in the Gateway log file. The format of this identifier depends on the application that sends the relevant message.

RecDate	Date	Date on which Gateway recorded a message in the Gateway log file
RecTime	String	Time at which Gateway recorded a message in the Gateway log file
SeverityClass	String	Content and severity of an error message in the Gateway log file. The possible values of this attribute are: • ES: System Error • EC: Component Error • EM: Message Error • AV: Error and Warning • IG: General Information • IP: Program Information

XFBLog Predefined Request

When you configure Sentinel to monitor Gateway, you can import a predefined Request for XFBTransfer: *TransferLog*. This Request retrieves Tracked Instances that describe the contents of a Gateway log.

Axway Gateway: Connectors

Managing .pmq files with the pmqctl command

The `pmqctl` command is only intended for system administrators, preferably with the supervision of Axway Support.

Introduction

The `pmqctl` command enables you to manage `.pmq` files used to buffer data to send to Axway Sentinel. In certain cases you may need to optimize the space allocated for these files. This may be necessary if, for example, Sentinel is not fed fast enough or if events are not working. You can also use this command to see if the router queue is full.

The Router process also uses an internal `router.pmq` file but this is handled internally by the supervisor process.

pmqctl usage

Enter the `pmqctl` command, without arguments, to display the help text:

Usage :

```
pmqctl read pathname [[nb_to_read] [nb_per_sync]]

pmqctl enlarge pathname new_size

pmqctl create pathname [size]

pmqctl info pathname

pmqctl list pathname

pmqctl dump pathname
```

pmqctl enlarge pathname new_size

Use this command to specify the useful data space of the file. You can use the command when Gateway is running.

For example, enter the following command to increase the space to 10 000 000 bytes:

```
pmqctl enlarge $p_home_dir/run_time/data/tracker.pmq 10000000
```

The resulting file size will be 10 000 256 bytes (256 bytes of meta-data are added to the file).

pmqctl info pathname

Use this command to print out information about the file. You can use the command when Gateway is running.

Enter:

```
pmqctl info pathname
```

Example of output:

```
--
    signature : a1b2c3d4
version_major : 1
version_minor : 0
    status : 1
    reader_pid : 25945
    sequence : 1001
    sync_seqn : 1000
    read_index : 2450
    write_index : 9852
    end_index : 8000000
--
5 records total, 7401 bytes total
```

Other commands

The following commands are supplied for diagnostic purposes. Only use them if Axway Support advises you to use them.

```
pmqctl create pathname [size]
```

```
pmqctl read pathname [[nb_to_read] [nb_per_sync]]
```

```
pmqctl list pathname
```

```
pmqctl dump pathname
```

Warning: Never run the `pmqctl read` command against an instance of the `tracker.pmq` file on a production system as it can destroy data (information is read and removed from the file).

[About Axway Sentinel](#)

Axway Gateway: Connectors

Axway Integrator / AMTriX

About Axway Integrator

Axway Integrator (formerly XIB) is a suite of Axway products that provides a business-to-business (B2B) integration broker environment. It enables organizations to create and run dynamic online trading communities with multiple trading partners and high transaction volume.

Axway Integrator provides:

- Trading Partner connectivity
- Supply web visibility
- Business process modeling

About AMTriX

Axway AMTriX is an EDI (Electronic Data Interchange) integration broker. EDI enables businesses to place orders, send invoices, publish catalogues and carry out other operations without exchanging letters, phone calls or faxes. Business operations are carried out automatically by means of structured information exchanges between computers in compliance with international standards.

AMTriX enables you to:

- Exchange data with your suppliers, clients and commercial partners in compliance with the various EDI standards (EDIFACT, Ansi X12, GENCOD, Tradacoms, etc.)

- Use multiple data transport modes at your disposal (file transfer, email, value-added networks, etc.)
- Integrate your external exchanges with your ERP or internal applications

[Axway Integrator connector](#)

Axway Gateway: Connectors

Axway Integrator connector

[Axway Gateway and Axway Integrator / AMTrix](#)

[Required versions](#)

[Deployment](#)

[Configuration](#)

[Purging for transfers routed from Gateway to Integrator](#)

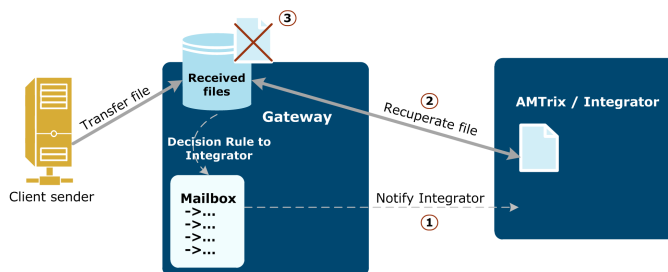
Axway Gateway and Axway Integrator / AMTrix

The Axway Integrator connector in Gateway enables you to:

- Route files received on Gateway to Integrator (or AMTrix)
- Send files from Integrator (or AMTrix) using any of the Gateway supported protocols

Routing files received on Gateway to Integrator

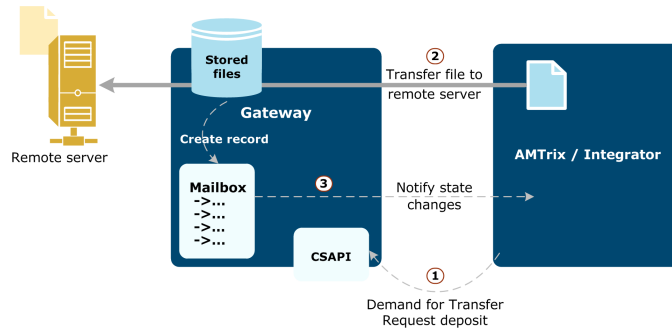
After configuring the connector, Gateway handles file routing to Integrator (or AMTrix) as illustrated in the following figure.



1. On reception of a file from a Gateway client, Gateway sends a notification to the destination Integrator server indicating that a file is available for retrieval by Integrator.
If Sentinel monitoring is activated on Gateway, the notification includes information that enables Gateway and Integrator to share a CycleID link on the transfer.
2. Integrator retrieves the file identified in the Gateway notification.
3. Integrator deletes the copy of the retrieved file in the Gateway directory.

Sending files from Integrator to remote servers using any Gateway supported protocol

Once you have configured your connector, Integrator (or AMTrix) sends files to remote servers via Gateway, as illustrated in the following figure.



1. Integrator initiates a request to Gateway for a Transfer Request of a file to a remote server. This request from Integrator is handled by a Gateway API. The initiation request includes information that enables Gateway to access the file on the Axway Integrator platform.
2. Gateway uses information contained in the initial Integrator request to create a Transfer Request sending a copy of the indicated file from Integrator to the target remote server. Gateway derives the `file_component` attribute of the Transfer Request (indicating the location of the file to be transferred from the Axway Integrator machine) from information provided by Integrator.
3. Gateway notifies Integrator of each change of transfer state (ToBegin, Processing, Ended, Acked...).

If Sentinel monitoring is activated, Gateway generates Sentinel state change messages and CycleID for the transfer.

Required versions

You can connect Gateway to:

- Axway Integrator 2.0.1 (or later)
- Axway AMTrix 4.4.0 (or later)

Deployment

Two types of configuration are possible:

- Gateway and Integrator (or AMTrix) on the same machine
- Gateway and Integrator (or AMTrix) on two different machines running the same operating system

If you deploy Gateway and Integrator (or AMTrix) on different machines, you must create a folder on each machine to which the other machine has access.

Configuration

To configure Gateway for use with Integrator (or AMTrix), set up the Integrator connector in the Gateway [configuration menu](#):

1. In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
2. Select **Connectivity > Axway Connectivity > Integrator**.
3. Complete the [Integrator parameters](#).
4. Click **Activate** to enable the connector.
5. Click **OK**.

From Gateway to Integrator

Integrator (or AMTrix) listens to the port that you define for it in the Gateway configuration menu. Gateway connects via this port, and sends the transfer parameters to Integrator. By default, the Gateway routing file is located in `run_time\tmp`.

If there is a problem with the connection, Gateway creates a new file that lists the transfer parameters. This file is placed in the *shared directory* which is accessible to both Gateway and Integrator. When Integrator is reconnected, it can recover this file. Therefore, the shared directory is used only to record off-line functioning.

From Integrator to Gateway

Integrator (or AMTrix) uses the API-C standards for Gateway to create a transfer. By default, the file coming from the source Integrator is placed in the `xib/data/tmp` directory.

Configuration limitations

- You must never configure Gateway to have more than one action (`route-to-XIB`) for a transfer. This means the only decision rule for a transfer routed to Integrator should be in the transfer's final state: `ENDED`, `ENDED-TO-ACK` (for business ack only), `ACKED`.
- Routing `ENDED_TO_ACK` transfers to Integrator (FileAct RealTime) generates a business ACK from Integrator. If the receipt option is set to Monitor (Automatic), Gateway automatically sends its own ACK message which results in error for the second ack send. The recommendation is to update the decision rule by adding the value `MANUAL` for **receipt send mode**. In such cases there will be no ACK duplication by assuring that only transfers which have set the manual ACK (`ACK_BY_USER`) are routed to Integrator. An alternative would be to set the receipt option to `ACK_BY_USER`.
- For SWIFTNet FileAct RealTime, the final state should be `ENDED` or `ENDED_TO_ACK` and Integrator must send the business ACK (receipt option: ACK By User).
- For SWIFTNet Store-and-Forward flows, the final state should be `ACKED`, and Gateway must send the protocol ACK (receipt option: ACK By Monitor).

Purging for transfers routed from Gateway to Integrator

To prevent Gateway's purge from deleting the payload file after a transfer is routed (via a decision rule) from Gateway to Integrator, even if Integrator was offline or was not fast enough to process the payload:

- The transfers routed to Integrator are marked by a `routed_to_XIB` flag. This is done automatically after the routing and can be seen using `peldsp display_trans` command line.
- The purge operation, when **Purge the file sent or received** is selected, will not delete the files associated with the `routed_to_XIB` transfers (including also the temporary files). The following protocols do not obey this rule and will have their temporary files (only) removed during purge:
 - AS1 (SMTP and POP3)
 - RosettaNet (SMTP, POP3 and HTTP)
 - AS2
 - Odette

Limitations

Note that this approach has certain limitations:

- You must ensure that a purge operation is not launched when routing to XIB is in progress. The transfer will have to be marked during the routing action itself so purge would later know to delete the transfer from the mailbox but not its payload.
- You must never configure Gateway to have more than one action (route-to-XIB) for a transfer. This means the only decision rule for a transfer routed to Integrator should be in the transfer's final state: ENDED, ENDED-TO-ACK (for business ack only), ACKED.
- Multiple post-processing must never be performed on the same transfer or it may cause undesirable results like in the following case:
 - If a script, for example, is launched by a decision rule prior to the decision rule routing the transfer to XIB, a race condition occurs: if the user script modifies the data, Integrator may read the file in an inconsistent state. If a script is then launched by a decision rule after routing the transfer to XIB, the script can access the payload file while it is still being modified by Integrator.

Note: Gateway does not modify or delete the payload file after a transfer has been routed to XIB.

[About Axway Integrator / AMTriX](#)

Axway PassPort PM connector

About PassPort PM

PassPort PM (Partner Management) enables you to manage partners on the Axway Platform.

Activating the PassPort PM connector

To configure Gateway for use with PassPort PM, set up the PassPort PM connector in the Gateway [configuration menu](#):

- In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
- Select **Connectivity > Axway Connectivity > PassPort**.
- Complete the [PassPort parameters](#).
- Click **OK**.

Enter the following information to configure the PassPort PM connection:

- PassPort Version
- Select the **Use partner management** option
- The PassPort server host name or IP address
- The PassPort HTTP port
- The Component Id
- The login user name
Note: you need to set the component password by running the `pelencpass` command locally on the Gateway server.
- Also you can select the **Use TLS** option to secure the connection between Gateway and PassPort PM. If you do that, you need to enter the following parameters:
 - The PKI SSL port
 - The PassPort CA location is pre-defined, but you can modify it.

PassPort PM cache

After modifying a partner from PassPort PM, it may take some time until Gateway can use the updated values (based on the value of the `vsite_age_max` parameter). There are two `pelctl` commands you can use to invalidate the PassPort PM cache, and view the changes earlier:

- To invalidate all PassPort's PM cache:
`pelctl invalidate_tpm_cache`
- To invalidate only the `<comprofile>` entry in the cache:
`pelctl invalidate_tpm_cache -tpmdea <comprofile>`

Optimizing performance between Gateway and PassPort PM

When using PassPort PM, Gateway saves the results of queries in specific Site objects called *Virtual Sites*. These Virtual Sites save PassPort from being queried all the time by Gateway.

Virtual Sites have a tunable lifetime set by the `vsite_age_max` parameter. Setting a long lifetime reduces the number of queries to PassPort resulting in shorter processing times. However, during that time any change made in the related PassPort entry is not refreshed. On the other hand, having a short `vsite_age_max` value allows quicker updates of PassPort modifications within Gateway, but with poorer performance.

Modify the `vsite_age_max` parameter in the [Gateway advanced options](#).

[Using PassPort PM with OFTP](#)

[Managing X.400 partners](#)

Axway Gateway: Connectors

Axway PassPort PM connector

About PassPort PM

PassPort PM (Partner Management) enables you to manage partners on the Axway Platform.

Activating the PassPort PM connector

To configure Gateway for use with PassPort PM, set up the PassPort PM connector in the Gateway [configuration menu](#):

- In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
- Select **Connectivity > Axway Connectivity > PassPort**.
- Complete the [PassPort parameters](#).
- Click **OK**.

Enter the following information to configure the PassPort PM connection:

- PassPort Version
- Select the **Use partner management** option
- The PassPort server host name or IP address
- The PassPort HTTP port

- The Component Id
- The login user name
Note: you need to set the component password by running the `pelencpass` command locally on the Gateway server.
- Also you can select the **Use TLS** option to secure the connection between Gateway and PassPort PM. If you do that, you need to enter the following parameters:
 - The PKI SSL port
 - The PassPort CA location is pre-defined, but you can modify it.

PassPort PM cache

After modifying a partner from PassPort PM, it may take some time until Gateway can use the updated values (based on the value of the `vsite_age_max` parameter). There are two `pelctl` commands you can use to invalidate the PassPort PM cache, and view the changes earlier:

- To invalidate all PassPort's PM cache:
`pelctl invalidate_tpm_cache`
- To invalidate only the <comprofile> entry in the cache:
`pelctl invalidate_tpm_cache -tpmdea <comprofile>`

Optimizing performance between Gateway and PassPort PM

When using PassPort PM, Gateway saves the results of queries in specific Site objects called *Virtual Sites*. These Virtual Sites save PassPort from being queried all the time by Gateway.

Virtual Sites have a tunable lifetime set by the `vsite_age_max` parameter. Setting a long lifetime reduces the number of queries to PassPort resulting in shorter processing times. However, during that time any change made in the related PassPort entry is not refreshed. On the other hand, having a short `vsite_age_max` value allows quicker updates of PassPort modifications within Gateway, but with poorer performance.

Modify the `vsite_age_max` parameter in the [Gateway advanced options](#).

[Using PassPort PM with OFTP](#)

[Managing X.400 partners](#)

Axway PassPort PS connector

About PassPort PS

PassPort PS (PKI Server) is a stand-alone tool that you can use in a Public Key Infrastructure (PKI) with Gateway when Gateway communicates with partners via a secure protocol (SSL/TLS or SSH).

Using Gateway with PassPort PS

When you use Gateway with PassPort PS, PassPort PS assures certain security functions in place of Gateway:

- Certificate selection for exchanges with remote partners
- Data encryption and signing

Activating the PassPort PS connector

To configure Gateway for use with PassPort PS, set up the PassPort PS connector in the Gateway [configuration menu](#):

- In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
- Select **Connectivity > Axway Connectivity > PassPort**.
- Complete the [PassPort parameters](#).
- Click **OK**.

Enter the following information to configure the PassPort AM connection:

- PassPort Version
- Select the **Use PKI Services** option
- The PassPort server host name or IP address
- The PassPort PKI port
- The Component Id
- The login user name
Note: you need to set the component password by running the `pelencpass` command locally on the Gateway server
- Also you can select the **Use TLS** option to secure the connection between Gateway and PassPort PS. If you do that, you need to enter the following parameters:
 - The PKI SSL port
 - The PassPort CA location is pre-defined, but you can modify it

Axway Gateway: Connectors

Axway PassPort AM connector

About PassPort AM

PassPort AM (Access Management) centralizes the management of users for all products of the Axway Platform. If you have installed PassPort AM, you can use it to define all your users and profiles. Gateway can then connect to PassPort AM to check user permissions.

Activating the PassPort AM connector

To configure Gateway for use with PassPort AM, set up the PassPort AM connector in the Gateway [configuration menu](#):

- In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
- Select **Connectivity > Axway Connectivity > PassPort**.
- Complete the [PassPort parameters](#).
- Click **OK**.

Enter the following information to configure the PassPort AM connection:

- PassPort Version
 - Select the **Use access management** option
 - The PassPort server host name or IP address
 - The PassPort HTTP port
 - The Component Id
 - The login user name
- Note:** you need to set the component password by running the `pelencpass` command locally on the Gatewayserver.
- Also you can select the **Use TLS** option to secure the connection between Gateway and PassPort AM. If you do that, you need to enter the following parameters:
 - The HTTPS port
 - The PassPort CA location is pre-defined, but you can modify it.
 - The Client certificate location is also pre-defined, but you can modify it.

Axway Gateway: Connectors

SWIFTNet

SWIFTNet Introduction

SWIFTNet is the SWIFT organization's IP-based network. SWIFTNet services enable the secure and reliable transfer of financial information and transactional data.

Many institutions frequently need to manage separate interfaces, standards and security models for each market infrastructure used, and for various types of communication with correspondents and customers. With SWIFTNet, you connect only once. Single window connectivity delivers a single communications infrastructure to access multiple service providers, correspondents and customers. Other advantages are automation and straight-through processing within a financial organization.

The Gateway/SWIFTNet connector supports the following SWIFTNet services:

- InterAct messaging
- FileAct file transfer

SWIFTNet message types

Originally SWIFT used FIN messaging. More recently, with SWIFTNet, the XML-based InterAct and FileAct message types have been introduced.

InterAct

InterAct is SWIFT's interactive messaging service that supports the exchange of messages between two parties.

With InterAct, institutions and communities can exchange messages in an automated and interactive way — an application sends a request message to another application and receives an immediate response message.

Each message exchange consists of a request and a reply message. Request messages usually contain either information to be sent from a sender to a receiver or a request for information which is sent from sender to receiver. Reply messages contain either the confirmation that the receiver has received the sender's message or a reply on a sender's request.

InterAct is used to exchange time-critical and short to medium length messages (such as securities orders, payment instructions) between parties.

Each InterAct message consists of two parts:

- The *payload* that contains the business contents, such as an investment funds order
- The *envelope* that contains the technical information which is required for the SWIFT network to send the message (such as requester and receiver, PKI information, SnF mode, etc.)

InterAct messages can be validated, for example for XML compliance and correct semantics, within SWIFTNet according to the message specification.

FileAct

FileAct allows the secure and reliable transfer of files between parties via SWIFTNet.

FileAct supports tailored solutions for market infrastructure communities, closed user groups and financial institutions. FileAct is particularly suitable for bulk payments, batches of structured financial messages and large reports.

Files are usually created in batch processes and should not be time critical. Files are typically large. There is no validation of FileAct messages. They can be free formatted, even binary data can be transmitted.

SWIFTNet delivery modes

SWIFTNet provides two different delivery modes:

- *Real Time* (RT) mode exchanges messages and files between parties in real time. In this case both sender and receiver have to be connected to the SWIFT network at the same time.
- *Store-and-Forward* (SnF) mode allows the sender's message to be stored within the SWIFT network until the receiver is able to receive the message. The sender and receiver do not need to be connected at the same time. SnF relies on queues managed by SWIFT.

SWIFTNet software

SWIFT provides the following software to connect to SWIFTNet:

- SWIFTNet Link (SNL). This is the low-level connection software that manages access to SWIFTNet. A client may have more than one instance of SNL for reasons of performance and reliability. SNL connects gateway applications such as SAG to SWIFTNet.
- SWIFT Alliance Gateway (SAG). SAG acts as concentrator and allows multiple applications to be connected to SWIFTNet. Examples are: automated file transfer, distant applications using SWIFTNet RA (Remote API) and applications using MQ-series queues. Flows coming from SWIFTNet to these applications are routed according to configurable criteria.

SWIFTNet message and file limitations

Gateway fully supports the SWIFTNet InterAct and FileAct protocols within the size limits established by SWIFT. For the current limits, refer to the official SWIFTNet documentation.

More information

Gateway is just part of the Axway Financial Exchange solution. Using Axway products to deploy a full SWIFTNet solution is outside the scope of this document. For more information on Axway products, go to www.axway.com.

For more information on SWIFTNet, refer to:

- The SWIFT (Society for Worldwide Interbank Financial Telecommunication) website:
www.swift.com
- The SWIFTNet documentation delivered with the product

Axway Gateway: Connectors

SWIFTNet connector

SWIFTNet and Gateway

The Gateway SWIFTNet connector enables you to perform the following roles:

- Transport and route SWIFTNet messages between partners and platforms
- Manage SWIFTNet file/message transfers

Functionality

The Gateway SWIFTNet connector offers the following functions:

- SWIFTNet data exchange, with full support of:
 - FileAct/InterAct services
 - Real Time/Store-and-Forward delivery modes
- Transfer management:
 - Parallel processing (inbound/outbound)
 - Priority handling
 - Reliability (if a transfer is not successful, Gateway retries)
- Routing:
 - Automated transfer routing
- Monitoring:
 - Retry, restart on failure
 - Event cycle linking between applications

Required Gateway objects for SWIFTNet

You need to create the following Gateway objects to be able to exchange and/or route SWIFTNet files and messages.

Remote Site object

In the Remote Site object you specify the characteristics of the SWIFTNet access point. If you have several access points, create a Remote Site object for each one.

Local Trading Partner object

In the Local Trading Partner object you specify your local SWIFTNet identification information.

Remote Trading Partner object

In the Remote Trading Partner object you specify your business partner's SWIFTNet identification information. In addition, you specify SWIFTNet services and delivery mode.

Transfer Request object

In the Transfer Request object you describe how a file is sent or received (using SWIFTNet FileAct) or how a message is sent or received (using SWIFTNet InterAct). Amongst other parameters, it specifies:

- The SWIFTNet Remote Site
- The SWIFTNet Local Trading Partner object
- The SWIFTNet Remote Trading Partner object
- ...

Acknowledgment Message object

In the Acknowledgement Message object you specify information concerning receipts using SWIFTNet FileAct in Real Time.

Decision Rule object and Model object

If you want to route SWIFTNet files and/or messages to another business partner, use the Decision Rule object and Model object.

Axway Gateway: Connectors

Configuring the Axway Gateway/SWIFTNet environment

Prerequisites

Before you can use Gateway to send and receive transfers via SWIFTNet, you must:

1. Install Gateway and SWIFTNet RA 7 (Remote API) on the same machine. This machine must comply with SWIFTNet 7 prerequisites. For the minimum patch level and details of supported platforms, go to www.swift.com and select **Support**.
2. Configure SWIFTNet RA 7 to access SAG 7 (SWIFT Alliance Gateway). RA contains the libraries that are required at runtime.
3. In SAG, when defining a Message Partner to be used with Gateway:
 - a. In the **Type** field, select *ClientServer* as connection mode (Gateway uses both Server and Client, for instance to receive acknowledgments in Real Time)
 - b. In the **Host Adapter** field, select *Remote API Host Adapter* as message partner.
 - c. In the **Default Message Format for Emission** field, select *Strict SNL Format*.
 - d. In the **Supported Message Formats** pane, select *Strict SNL Format*. Do **not** use *Relaxed SNL Format*!
4. Ensure that the user has the necessary rights to access RA.
5. Ensure that any DN you use in Gateway is related to a certificate defined with **strict** mode. Gateway always enforces the strict mode for any certificates it uses.

For supported platforms, refer to the *SWIFTNet documentation*.

Configuring the SWIFTNet connector

To configure Gateway for use with SWIFTNet, set up the SWIFTNet connector in the Gateway [configuration menu](#):

1. In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
2. Select **Connectivity > Axway Connectivity > SWIFTNet**.
3. Complete the [SWIFTNet parameters](#) (**Default RA installation pathname**).
4. Click **Activate** to enable the connector.
5. Click **OK**.

Required objects

To carry out a SWIFTNet transfer to or from a remote business partner, you require as a minimum the following Gateway objects:

- Remote Site object
- Local Trading Partner object
- Remote Trading Partner object

Remote Site object

In the Remote Site object you specify the characteristics of the SWIFTNet access point. You can also define a [backup Remote Site](#) to ensure service continuity.

Local Trading Partner object

In the Local Trading Partner object you specify local identification information.

Remote Trading Partner object

In the Remote Trading Partner object you specify the following parameters for exchanges with your business partner:

- Trading partner identification
- Transfer and file descriptions
- Requested SWIFTNet services (for example, signature or encryption)
- Delivery mode

SYSTEM_MESSAGES remote Trading Partner

System messages are InterAct messages generated by SWIFT. The RequestHeader of system messages always identifies SWIFT by the DN "cn=system,o=swift,o=swift". The Service is "swift.snf.system" or "swift.snf.system!p". The RequestType starts with the business area "xsys" indicating system messages.

When you install Gateway, it automatically creates the SYSTEM_MESSAGES remote Trading Partner. This Trading Partner is created with unknown request type "AnyRequestType" so that it accepts all system messages. If required, you can create separate remote Trading Partners for each request type.

The default value for trade service name (**Service name**) is "swift.snf.system!x". You should update this parameter according to your needs (for example: for transfers made on the SWIFT test or live environment).

Creating objects for SWIFTNet transfers

This section describes how to create the following objects for SWIFTNet transfers:

- [Remote Site objects](#)
- [Local Trading Partner objects](#)
- [Remote Trading Partner objects](#)

Creating a Remote Site object for SWIFTNet transfers

In the Remote Site object you specify the characteristics of the SWIFTNet access point. If you have several access points, create a Remote Site object for each one.

To create a Remote Site object:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Right-click the **Remote Site** sub-node, then select **New** to display the *New Remote Site* window.
3. Complete the fields on the **General** tab as follows:

Field	Description
Alias	Enter the Remote Site alias. This is a local name, only used in Gateway. Maximum: 31 alphanumeric characters.
Protocol	Select SWIFTNet . Gateway then disables some fields in the Options tab and creates the SWIFTNet tab.
Comments	<i>Optional</i> Enter a description. Maximum: 80 alphanumeric characters.
Role	Select all of the following options: • Initiator/Sender • Initiator/Receiver • Responder/Sender • Responder/Receiver Because of the architecture of the SWIFTNet protocol, a communication node is at the same time both Initiator and Responder as well as being both Sender and Receiver. Examples Initiator: When sending a FileAct transfer/ when sending a receipt. Responder: When receiving FileEvent notifications from the RA on the initiated transfer and when receiving the receipt/ receiving a FileAct transfer.
Initialization mode	Select both of the following initialization modes: • Initiating Site • Responding Site Optionally, activate the Trace check-box. Note: The dynamic settings for the Initiator and Responder modes must also both be on (default value) for a SWIFTNet Site to be active. Access these settings from the Remote Site's context menu. Alternatively use the peladm update_site [-init_mode (-im) { Y N }] [-resp_mode (-rm) { Y N }] command to modify these settings.

4. Optionally, complete the Sentinel settings on the [Options](#) tab.
5. Complete the fields on the [SWIFTNet](#) tab.
6. Click **OK**.

Gateway creates a new Remote Site object with the specified characteristics.

Using command line

Alternatively, you can create a Remote Site object using the [peladm create site](#) online command.

Example:

```
peladm create_site -alias SAG1
  -protocol SWIFTNET
  -comments "Testing Remote Site"
  -login_ident test
  -login_password test1234
  -ra_instance Ra1
  -message_partner TESTmp
  -event_end_point GTWleep
  -snf_queues "testxxx_tst01iast!x testxxx_tst 01iadm!x
testxxx_tst 01fa!x testxxx_tst01fadn!x"
  -sep testxxx_tst
```

Creating a Local Trading Partner object for SWIFTNet transfers

The Local Trading Partner object provides Gateway with information about local identification.

You can create Local Trading Partner objects either using the GUI or command line.

Using the GUI

To create a Local Trading Partner object using the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.
Gateway displays a context menu.
3. Select **New**.
Gateway displays a context menu.
4. Select the type of partner: **SWIFTNet partner**
Gateway displays the *Trading Partner* window.
5. Complete the fields of the *Trading Partner* window as follows:

Local Trading Partner – SWIFTNet

Field	Description
Identification criteria	
Alias	Name of the Trading Partner object to use for local identification operations. The name must be unique. Maximum: 31 alphanumeric characters. Gateway stores this alias as part of the data of each Transfer Request. You must use the alias when submitting Transfer Requests.
Format	<i>You cannot edit this field</i> Protocol format: SWIFTNet .
Local Partner	Select this option (<i>local</i> /Trading Partner object).
Name	Name of the local Trading Partner object. This name represents the DN (Distinguished Name) of: • The sender in Initiator mode • The receiver in Responder mode Maximum: 100 alphanumeric characters.
Comments	<i>Optional</i> Free-text description of the Trading Partner object. This description is included in the displayed list of available Trading partners. Maximum: 80 alphanumeric characters.

6. Click **OK**.

Gateway creates a new Trading Partner object with the specified characteristics.

Using command line

To create a Local Trading Partner object via command line, use the `peladm create_tradepart` command.

Example:

```
peladm create_tradepart  -alias LSWDEVFO
                        -name "o=bankabc,o=swift"
                        -is_local Y
                        -format SWIFTNET
```

Creating a Remote Trading Partner object for SWIFTNet transfers

The Remote Trading Partner object provides Gateway with information about the trading partner and the requested SWIFTNet services.

You can create Remote Trading Partner objects either using the GUI or command line.

Using the GUI

To create a Remote Trading Partner object using the GUI:

1. In the left pane of the GUI main window, click  to expand the **Partner Management** node.
2. Right-click the **Trading Partners** sub-node.
Gateway displays a context menu.
3. Select **New**.
Gateway displays a context menu.
4. Select the type of partner: **SWIFTNet partner**
Gateway displays the *Trading Partner* window.
5. Complete the fields of the *Trading Partner* window as follows:

Remote Trading Partner – SWIFTNet

Field	Description
Identification criteria	
Alias	Name of the Trading Partner object to use for local identification operations. The name must be unique. Maximum: 31 alphanumeric characters. Gateway stores this alias as part of the data of each Transfer Request. You must use the alias when submitting Transfer Requests.
Format	<i>You cannot edit this field.</i> Protocol format: SWIFTNet .
Local Partner	Do not select this option when you are creating a <i>Remote</i> Trading Partner object.

Field	Description
Name	Name of the remote Trading Partner object. This name represents the DN (Distinguished Name) of: - The receiver in Initiator mode - The sender in Responder mode Maximum: 100 alphanumeric characters.
Comments	<i>Optional</i> Free-text description of the Trading Partner object. This description is included in the displayed list of available Trading partners. Maximum: 80 alphanumeric characters.
Service name	SWIFT business service name (as specified in the XML header). Maximum: 30 alphanumeric characters.
Request type	Type of request (as specified in the XML header). Maximum: 30 alphanumeric characters. Also refer to Unknown request type.
Gateway specific parameters	
Associated local partner	Either: - Enter a local partner alias, or - Select a partner from the drop-down list
Direction allowed	Authorized directions for transfers using this partner: Receive Send Both (default)
Default outgoing site	Either: - Enter a default outgoing Site name, or - Select a Site name from the drop-down list The Site must be an existing Site on your local Gateway. When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.

Field	Description
Mode allowed	Authorized modes for transfers using this partner: • Initiator • Responder • Both (default)
General SWIFT parameters	
Delivery mode	Select one of the following values: • Real time = immediate (default) • Store-and-Forward = send and then store in the queue in the SWIFTNet domain
Encrypt	<i>InterAct only</i> Select this option to activate message encryption.
Sign	Select this option to activate message signing. The option is activated by default if you have selected Non repudiation .
Compress	Select this option to activate file or message compression.
Non repudiation	Select this option to request SWIFT to keep proof of sending (availability depends on the service used).
Signature list	Reference to file on disk containing a Signature List. This file must respect a specific structure .
Transfer description	<i>Supplied by the initiator</i> Free-text description of the transfer. Maximum: 256 alphanumeric characters.
File description	<i>Supplied by the sender</i> Free-text description of the file. Maximum: 256 alphanumeric characters.
Transfer information	<i>Supplied by the initiator</i> Structured information about the transfer. For more details, refer to the SWIFT documentation.

Field	Description
File information	<i>Supplied by the sender</i> Structured information concerning the transfer. For more details, refer to the SWIFT documentation.
Customize file info parser	Select whether or not to customize the SWIFT grammar rules for the Sw:FileInfo field for this partner. Warning: In normal circumstances you should not override the default values provided in Gateway: • Separator: ; [semi-colon] • Equals: = [equals sign] • Quotes: " [quote mark] These default values correspond to the grammar for the Sw:FileInfo field as defined in the SWIFT documentation. In some specific cases, for example for backward compatibility with older applications that are not SWIFT compliant, you may need to modify one or more of these three special characters. The Trading Partner parameter values override the global values specified in the configuration menu. You should modify the Trading Partner values rather than the configuration menu values if you have a small number of partners that do not use the standard SWIFT grammar.
InterAct only parameters	
Response payload	<i>Real time only.</i> Enter the name of the local file containing the default reply message.

Field	Description
Payload type	Select the payload format: • Text (default) • Embedded: XML format, enclosed in SWIFT payload tags. Use the following format: <pre><SwInt:RequestPayload> request message </SwInt:RequestPayload></pre> or <pre><SwInt:ResponsePayload> reply message </SwInt:ResponsePayload></pre>
Sign DN	Only if you have selected Non repudiation or Sign If specified, name used for signing in Initiator mode. Otherwise, the <i>securityDN</i> (supplied by SNL) is used.
Encrypt DN	Only if you have selected Non repudiation or Encrypt If specified, name used for encryption in Initiator mode.
Notification parameters	
Receipt option	Determines how acknowledgments are handled. For FileAct/Real time, this refers to the business real-time acknowledgement. For FileAct/Store-and-Forward and InterAct/Store-and-Forward, this refers to the delivery notification. Specify one of the following values: • Ack by monitor: automatically send an acknowledgement • Ack by user (default): wait for acknowledgement sent by user
Receipt request synchronized	<i>InterAct, Real time only.</i> Select this option to activate the waiting period for a Transfer receipt containing the reply message.

Field	Description
Receipt request type	<i>FileAct, Real time only, if you have selected Request receipt</i> Type of acknowledgement request.
Request receipt	Select this option to receive a receipt when the file is received (recommended for Store-and-Forward mode).
Receipt DN	<i>FileAct, Real time only, if you have selected Request receipt</i> DN = Distinguished Name Name of the receipt reception address. This is the address to which the receiving partner sends the receipt. By default, equal to local Name .
Notif queue	<i>Store-and-Forward mode only</i> Reception queue for Store-and-Forward delivery notifications.
Advanced parameters: Specifying destination folder and file name pattern for received files Click the Advanced... button to display the Directory path and File name pattern fields. These fields enable you to specify the directory and name format of the incoming files.	
Directory path	Enter the local physical path of the directory where received files are to be stored. For example: <code>d:\swiftnet</code> Gateway uses the same directory for certain file operations, such as decompression (<code>. dcm</code> files). If you do not specify a directory, Gateway creates the files in <code><Gateway installation directory>\runtime\tmp</code> The directory specified here takes precedence over any directory specified in the Remote Site.
File name pattern	Enter a file name format to enable you to identify the file in the specified directory. The file name pattern can contain symbolic variables that are replaced with their corresponding values at run-time. For example: <code>&(x_local_ident)</code>

Field	Description
Decompression	Enables you to deactivate the decompression of incoming archived InterAct and FileAct traffic. Specify one of the following values: • by Monitor (on): Decompression activated • by User (off): Decompression deactivated. Use this setting if for example the SWIFTNet flow uses a compression algorithm that is not supported by Gateway (supported algorithms: ZIP, ZLIB and GZIP).
Receipt via system message	Only in Store-and-Forward mode: Requests delivery of notification via system message. Note For copy and distributed transfers, Gateway will <i>always</i> request delivery notification via system message, regardless of this setting.

6. Click **OK**.

Gateway creates a new Trading Partner object with the specified characteristics.

Using command line

To create a Remote Trading Partner object via command line, use the `peladm create_tradepart` command.

Example:

```
peladm create_tradepart -alias SWDEVQUQL01
                        -name "o=bankxyz,o=swift"
                        -is_local N
                        -format SWIFTNET
                        -service "swift.fa.generic"
                        -request_type abcd.008.001.01
```

The following `peladm create_tradepart` parameters are specific to SWIFTNet transfers.

Parameter	SWIFTNet-related value
-compress (-c) Compress	Specify Y to activate file or message compression. • Y • N (default)
-decompression_option (-dcmo) Decompression	Enables you to activate or deactivate the decompression of incoming archived FileAct and InterAct traffic. Specify one of the following values: • on = Activate decompression (default) • off = Deactivate decompression
-delivery_mode (-dmo) Delivery mode	Specify one of the following values: • RT = immediate (default) • SNF = send and then store in the queue in the SWIFTNet domain
-dir_name (-dn) File name pattern	<i>For received files only</i> Specify a file name format to enable you to identify the received file in the specified directory. The file name pattern can contain symbolic variables that are replaced with their corresponding values at run-time. For example: &(x_local_ident)
-dir_path (-dp) Directory path	<i>For received files only</i> Specify the local physical path of the directory where received files are to be stored. For example: d:\swiftnet Gateway uses the same directory for certain file operations, such as decompression (.dcm files). If you do not specify a directory, Gateway creates the files in <Gateway installation directory>\runtime\tmp The directory specified here takes precedence over any directory specified in the Remote Site.
-direction_allowed (-dir_all) Direction allowed	Authorized directions for transfers using this partner: • I = In • O = Out • B = Both (default)

Parameter	SWIFTNet-related value
-encrypt (-e) Encrypt	<i>InterAct only</i> Specify y to activate message encryption. • y • n (default)
-encrypt_dn (-endn) Encrypt DN	<i>InterAct only</i> If specified, name used for encryption in initiator mode.
-file_component (-fc) Response payload	<i>InterAct, Real time only.</i> Filename of the local file containing the default reply message.
-file_description (-fdesc) File description	<i>Supplied by the sender</i> Free-text description of the file. Maximum: 256 alphanumeric characters.
-file_info (-finf) File information	<i>Supplied by the sender</i> Structured information concerning the transfer. For more details, refer to the SWIFT documentation.
-mode_allowed (-mod_all) Mode allowed	Authorized modes for transfers using this partner: • I = Initiator • R = Responder • B = Both (default)
-name (-n) Name	<i>Mandatory</i> Enter a unique name for the Trading Partner object. This name represents the DN (Distinguished Name) of: • The receiver in Initiator mode • The sender in Responder mode
-non_repudiation (-nr) Non repudiation	Specify y to request SWIFT to keep proof of sending (availability depends on the service used). • y • n
-notif_queue (-nq) Notif queue	<i>Store-and-Forward mode only</i> Reception queue for Store-and-Forward delivery notifications.

Parameter	SWIFTNet-related value
-payload_type_option (-pto) Payload type	<i>InterAct only.</i> Specify the payload format: text (default) embedded: XML format, enclosed in SWIFT payload tags. Use the following format: <SwInt:RequestPayload> request message </SwInt:RequestPayload> or <SwInt:ResponsePayload> reply message </SwInt:ResponsePayload>
-receipt_option (-ro) Receipt option	<i>FileAct, Real time only.</i> Determines how acknowledgments are handled. Specify one of the following values: ACK_BY_MONITOR: automatically send an acknowledgement ACK_BY_USER (default): wait for acknowledgement sent by user
-receipt_request (-rr) Request receipt	Specify signed or unsigned if you want to receive a receipt when the file is received (recommended for Store-and-Forward mode). none (default) signed unsigned
-receipt_request_synchronized (-rrs) Receipt request synchronized	<i>InterAct, Real time only.</i> Specify Y to activate the waiting period for a Transfer receipt containing the reply message. Y N
-receipt_request_to (-rrto) Receipt DN	<i>FileAct, Real time only</i> Enter the name of the receipt reception address. This is the address to which the receiving partner sends the receipt. By default, equal to local name .
-receipt_request_type (-rrty) Receipt request type	<i>FileAct, Real time only</i> Type of acknowledgement request.

Parameter	SWIFTNet-related value
-remote_site (-rs) Default outgoing site	Default outgoing Site name. The Site must be an existing Site on your local Gateway. When you send a file to a remote exchange partner, the remote partner uses this value to process the sent message.
-request_type (-rqt) Request type	Type of request (as specified in the XML header). Maximum: 30 alphanumeric characters.
-service (-sc) Service name	SWIFT business service name (as specified in the XML header). Maximum: 30 alphanumeric characters.
-sign (-s) Sign	Specify Y to activate message signing. • Y • N (default)
-sign_dn (-sidn) Sign DN	<i>InterAct only</i> If specified, name used for signing in initiator mode. Otherwise the securityDN (supplied by SNL) is used.
-sign_list (-signl) Signature List	Reference to file on disk containing a Signature List. This file must respect a specific structure.
-sw_copy_indicator (-ci)	SWIFTNet File or Message Copy service indicator. Possible values: • Y = yes • N = no Specify Y to request a copy of the file to be sent to a third party defined at the service level.
-sw_add_params (-swaddpar) Additional params	Reference to file on disk containing additional SWIFTNet structures (for example, Header Info and/or SignatureList). This file must respect a specific structure.

Parameter	SWIFTNet-related value
<code>-sw_auth_ indicator (-ai)</code>	SWIFTNet Authorization notification indicator. Possible values: • Y = yes • N = no Specify Y to receive Authorization system messages for files sent over Y-copy services.
<code>-sw_public_ recipient_list_ ind</code>	For use with a SWIFTNet Distribution List (distribution to multiple recipients). Use this parameter to specify whether or not the responder can receive the list of recipients.
<code>-sw_use_file_ info_parser_ options (-ufipo)</code> Customize file info parser	<i>In normal circumstances you should not need to use this parameter. If you do, use it in combination with the <code>sw_file_info_parser_options</code> parameter.</i> Select whether or not to customize the SWIFT grammar rules for the Sw:FileInfo field for this partner. Possible values: • Y = yes • N = no (default) Note: If you select "Y" you must also specify values for the <code>sw_file_info_parser_options</code> parameter.

Parameter	SWIFTNet-related value
<code>-sw_file_info_parser_options (-fipo)</code> Separator Equals Quotes	<i>In normal circumstances you should not need to use this parameter. If you do, use it in combination with the <code>sw_use_file_info_parser_options</code> parameter.</i> Modify one or more of the three special characters used in the grammar for the Sw:FileInfo field. Warning: Normally you should not override the default values provided in Gateway: • Separator: ; [semi-colon] • Equals: = [equals sign] • Quotes: " [quote mark] These default values correspond to the grammar for the Sw:FileInfo field as defined in the SWIFT documentation. In some specific cases, for example for backward compatibility with older applications that are not SWIFT compliant, you may need to modify one or more of these three special characters. The Trading Partner parameter values override the global values specified in the configuration menu. You should modify the Trading Partner values rather than the configuration menu values if you have a small number of partners that do not use the standard SWIFT grammar. Special characters Note that you may have to "escape" some special characters, such as quotes, when entering information on the command line. Example: To use a comma instead of semicolon for the separator but keep the equals sign and the quotes, enter the parameter as follows: <pre>-sw_file_info_parser_options ",= ""</pre> In this example, the complete string of characters is enclosed in quote marks and the quote mark character itself is repeated because it is a special character for the command line.
<code>-trade_org_alias (-tradeoa)</code> Associated local partner	Local partner alias.

Parameter	SWIFTNet-related value
-xfer_description (-xdesc) Transfer description	<i>Supplied by the initiator</i> Free-text description of the transfer. Maximum: 256 alphanumeric characters.
-xfer_info (-xinf) Transfer information	<i>Supplied by the initiator</i> Structured information concerning the transfer. For more details, refer to the SWIFT documentation.
-sw_delivery_notif_via_system_message (-swdnviasm) Delivery notification via system message	Requests delivery of notification via system message. Note For copy and distributed transfers, Gateway will <i>always</i> request delivery notification via system message, regardless of this setting.

Working with signed or encrypted messages

When handling signed and/or encrypted messages you need to check and complete the following encryption/decryption parameters:

- Remote Site:
 - **Sign ident List** (or [sign ident list](#) if you use online commands)
 - **Decrypt ident list** (or [decrypt ident list](#) if you use online commands)
 - Trading Partner:
 - **Sign DN** (or [sign dn](#) if you use online commands)
 - **Encrypt DN** (or [encrypt dn](#) if you use online commands)

Before making a Transfer Request, make sure that these parameters are coherent.

A certificate is associated with each user in SAG (Swift Alliance Gateway). This certificate is identified by a Distinguished Name (DN) and stored by SAG. SAG validates the access to these certificates via the user and password.

The Remote Site parameters **Sign ident List** and **Decrypt ident list** contain the names and passwords of users defined in SAG. These parameters enable the signing of sent messages and the decrypting of received messages.

In the case of a signature and/or encryption, at least one of the users defined in the **Sign ident List** parameter of the Remote Site must correspond to the DN defined in the **Sign DN** parameter of the SWIFTNet Trading Partner object.

The **Encrypt DN** parameter of the SWIFTNet Trading Partner object must be used in accordance with the distant application, which must have initialized its decrypting certificate(s).

The **Decrypt ident list** parameter of the Remote Site allows the distant application to encrypt messages using one of the initialized certificates.

Unknown request type

You can configure the Gateway SWIFTNet Connector to accept requests having an unknown request type for a Distinguished Name-Service couple:

1. Create a new remote Trading Partner containing the incoming Distinguished Name-Service couple.
2. For the **Request type** parameter, enter the reserved string AnyRequestType.

Configuring connection retry timeout

When you start Gateway, it tries to connect to all Remote Sites. For SWIFTNet this means initializing the access point and starting all server and client processes.

The `swift_job_restart_delay` parameter in the `conf` file stores the timeout (in seconds) between two successive connections attempts to SAG. The default timeout is 30 seconds. You can modify this value if required by editing the [pelsetup.def](#) file.

Specifying destination folder and file name pattern for received files

Use the following parameters in the Trading Partner to specify the directory and name format of incoming files.

- **Directory path** (or [dir_path](#) if you use online commands)
- **File name pattern** (or [dir_name](#) if you use online commands)

Axway Gateway: Configuration

Modifying conf file parameters

You can modify certain configuration parameters in the `conf` file. To make the modification permanent, you must modify the `pelsetup.def` file and then rebuild the configuration file.

[Modifying pelsetup.def](#)

[SWIFTNet parameters in the pelsetup.def file on page 2032](#)

[Configure where log files are stored in the pelsetup.def file on page 2033](#)

Modifying pelsetup.def

1. Go to the directory: `<Gateway installation directory>/run_time/etc`.
2. Edit the `pelsetup.def` file as required. (See the examples for SWIFTNet below.)
3. Rebuild the configuration file using the following command:
`peluconf standalone rebuild -f`
4. **Restart** Gateway so that the new values are taken into account.

Gateway's environment variables may contain either absolute or relative paths. You must run the profile before any operation related to your Gateway installation.

If you have issues after an update or upgrade (e.g: upgrade pack), you can use peluconf standalone rebuild -f to regenerate the configuration.

SWIFTNet parameters in the pelsetup.def file

The table below provides details of SWIFTNet parameters in the `pelsetup.def` file. You can find these parameters (or add them) in the SUP section of the file.

Parameter	Description	Exists by default?	Default value
<code>swift_job_retry_count_max</code>	This parameter controls the number of retries in the case of a connection error with SAG. The parameter value is taken into account by client and server processes. Example: If the connection to SAG is broken, the client/server process tries to reestablish the connections as many times as specified by this parameter.	no	10
<code>swift_job_restart_delay</code>	This parameter stores the timeout (in seconds) between two successive connections attempts to SAG. The parameter value is taken into account by client and server processes. Example: If the connection to SAG is broken, the client/server process tries to reestablish the connections after the period of time given by this parameter.	no	30

Example of part of pelsetup.def file contents

```
[SUP]
trace=0
.....
swift_job_retry_count_max = 7
swift_job_restart_delay = 600
.....
```

Configure where log files are stored in the pelsetup.def file

You can determine where log files are stored.

To do that, set the `archived_logdir` parameter in the `[LOG]` section of `pelsetup.def` file. For example:

```
[LOG]
archived_logdir="%p_database%"
```

If the `archived_logdir` parameter is not set, the archived logs will be saved in the same folder where the `log.dat` file is located.

Axway Gateway: Connectors

Working with Axway Gateway/SWIFTNet

[Working with SWIFTNet on Gateway](#)

[Submitting a SWIFTNet Transfer Request](#)

[After the transfer](#)

Working with SWIFTNet on Gateway

Before transferring files or messages using SWIFTNet you need to [configure Gateway and create various objects](#).

After completing these tasks, you can:

- Transfer a file – submit a Transfer Request to send a file to a business partner via SWIFTNet, optionally requesting a receipt
- Transfer a message – exchange SWIFTNet messages with remote business partners
- Transfer an Acknowledgment Message – generate and return receipts to the sending partner

Submitting a SWIFTNet Transfer Request

Transferring a file (FileAct)

To execute a FileAct file transfer using Gateway, [create and submit a Transfer Request](#).

Transferring a message (InterAct)

You can send limited-length text messages to a remote partner. To transfer a SWIFTNet InterAct message using Gateway, create and submit a Transfer Request. Perform the following operations in the *New Transfer Request* window:

1. In the **General** tab, go to the **Type** field and select **INTERACT**.
Note: This is equivalent to type MSG using command lines.
2. Define the message in one of two ways:
 - In the **Attributes** tab, go to the **File component** field and enter the name of the file containing the message
 - In the **SWIFTNet** tab, in the **InterAct** sub-tab, select the option **Send text message** and enter the message in the frame below

Transferring an Acknowledgement Message (FileAct)

Real Time mode only

After receiving a File transfer from a Remote Site, you may need to return an acknowledgement message to the sending machine. To do this, create a message text and associate it with the local ID of the received Transfer Request. You can enter the acknowledgement text directly in the acknowledgement transfer, or alternatively, associate a file containing the message text. You can pre-program the date and time when the acknowledgement message is to be sent.

After the transfer

To check that transfers have been made successfully, or to help you analyze problems, you can view the Mailbox and the Log file.

Using the Mailbox

For each SWIFTNet transfer you make using Gateway, Gateway records one or more transfer records in the Mailbox. The number of records depends on the message type (FileAct or InterAct) and the routing mode (real time or SnF).

SnF Incoming Acknowledgment

Incoming acknowledgment (Delivery Notification) messages for Store-and-Forward transfers are stored in a separate mailbox entry with type RECEIPT. This refers both to Counter Party and Third Party (received for Y-Copy and T-Copy transfers) Store-and-Forward delivery notifications, and is now consistent with the current behavior for FileAct RealTime.

The following points should be considered in order to evaluate which areas may be affected:

- Decision decision rules can be used on Store-and-Forward delivery notifications (receipts). This is an intended feature, however it means that pre-existing decision rules could be unexpectedly triggered for incoming Store-and-Forward delivery notifications (receipts).
- Log entries for received Store-and-Forward receipts (e.g. received Store-and-Forward delivery notification from Counter Party with status Accepted) are associated with the RECEIPT mailbox entry (new behavior) instead of the original transfer mailbox entry (old behavior).
- The Acknowledgement (Delivery Notification) message is stored in a separate file associated with the RECEIPT (see [SWIFTNet Proof Keeping](#))

Using the Log file

By default, Gateway generates traces of network activities that it records to the log file.

Axway Gateway: Connectors

SWIFTNet ProductList

[Overview](#)

[Example](#)

[Specifying the ProductList file](#)

[Checking file structure for XML conformity](#)

[More information](#)

Overview

SWIFTNet 7.0 requires the use of the ProductList XML structure. The ProductList identifies the messaging interface and backend applications used to send the message or file in generating the request or response to SWIFT.

You must provide the ProductList information for your backend application(s), including Gateway if it is used as a backend. Gateway as a messaging interface always adds itself in the ProductList. You can specify up to a maximum of three ProductInfo entries in a ProductList to specify the backend applications.

The ProductList is sent to SWIFT, but is not forwarded further to the partners. It is present in the following request/response structures:

- SwInt:RequestControl
- SwInt:ResponseControl
- Sw:FileRequestControl
- Sw:FileResponseControl
- Sw:FetchFileRequest
- Sw:SnFRequestControl
- Sw:SnFResponseControl

Example

Here is an example of an XML file containing ProductList information.

```
<AddSwParams>
  <SwProductList xmlns:Sw="urn:swift:snl:ns:Sw">
```

```
<Out>
  <Sw:ProductInfo>
    <Sw:VendorName>Axway</Sw:VendorName>
    <Sw:ProductName>Gateway</Sw:ProductName>
    <Sw:ProductVersion>6.17.0</Sw:ProductVersion>
  </Sw:ProductInfo>
</Out>
</SwProductList>
</AddSwParams>
```

Specifying the ProductList file

Gateway offers two ways of specifying the ProductList file:

- Per transfer - in the Transfer Request
- For all transfers exchanged with a specific partner - in the Trading Partner

For outgoing flows, both methods for specifying the Product List work (Transfer Request information overrides the Trading Partner).

For incoming flows, only the Trading Partner method is available.

The file containing the ProductList information must respect a specific structure, as described in "Checking file structure for XML conformity" later in this topic.

Transfer Request

In the Transfer Request, specify the full path of the local physical file containing ProductList information.

In the Gateway GUI, use the **Additional params** parameter on the **SWIFTNet** tab.

Alternatively, use the **peltrans** command:

```
peltrans -sw_add_params (-swaddpar) <file reference to additional
SWIFTNet structures>
```

Trading Partner

In the Trading Partner, specify the full path of the local physical file containing ProductList information.

When Gateway accepts requests through this Trading Partner, the ProductList file is used to fill the responses sent back to SWIFT.

In the Gateway GUI, use the **Additional params** parameter.

Checking file structure for XML conformity

The XML file must be well-formed and have a specific format. Refer to the XML schemas (.xsd files) supplied with Gateway for examples of file structure. You can use the `AddSwParams.xsd` file to validate XML files for ProductList usage. The .xsd files are located in:

```
<Gateway installation directory>/run_time/swnet
```

More information

For examples of files containing SWIFTNet ProductList information, refer to the SWIFTNet official documentation.

Axway Gateway: Connectors

Backup remote sites for SWIFTNet

[About backup Remote Sites for SWIFTNet](#)

[Backup site hierarchy limitation on page 2037](#)

[Defining a backup Site](#)

[Starting and stopping SWIFTNet Remote Sites](#)

[Transfer behavior](#)

[Static and Dynamic backup](#)

About backup Remote Sites for SWIFTNet

To ensure service continuity, you can define backup Sites for use with SWIFTNet transfers. If a transfer to a SWIFTNet Remote Site (the nominal Site) fails, for example if the connection with SWIFTNet is lost, Gateway will use the backup Site as an alternative.

Gateway can only use a backup Site if it is active. Depending on your requirements, you can start and stop backup Sites. If a transfer cannot be sent to the nominal Site or to a backup Site, it remains in the TO_BEGIN state until the connection with SWIFTNet is available.

Backup site hierarchy limitation

The backup Site hierarchy is limited to one. This means Gateway does not support a chain of backup Sites (more than one element in chain when the backup also has set a backup Site). Selecting a Site as backup for more than one nominal Sites is not supported.

One SWIFTNet remote sites can have up to 4 backup Sites selected.

See [Dynamically filtering duplicates for queues and output channels on page 2103](#)

Defining a backup Site

Define the backup Site when *creating* or *modifying* the nominal SWIFTNet Remote Site. You specify the alias of the backup Site in the nominal Site.

Prerequisite: The backup Site must exist (create the backup Site first).

To modify the nominal SWIFTNet Remote Site and define the backup Site:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
Gateway displays a list of all Remote Sites in the right pane.
3. Right-click the name of the Remote Site for which you want to modify the Site attributes and select **Modify...**
Gateway displays the *Remote Site* editing window.
4. Click the **SWIFTNet** tab.
The currently selected backup Sites are displayed in the Backup Sites list.
5. Click the **Change...** button next to the **Backup Sites** list.
6. In the newly created window, you can select up to 4 Sites as backup Sites.
Click **OK** to save the current selection and return to the Remote Site editing window.
7. Click **OK**.

Alternatively, use one of the following online commands:

- `peladm create_site -a ABCD -bks "BKPSITE1 BKPSITE2"` (create nominal Site)

- `peladm update_site -a ABCD -bks "BKPSITE1 BKPSITE2"` (update nominal Site)

Up to 4 backup Sites can be specified for a remote Site. Gateway checks that the specified backup Site aliases exists.

Starting and stopping SWIFTNet Remote Sites

Note: After restarting Gateway, all SWIFTNet Remote Sites are automatically started if both Initiator and Responder modes are set.

Stopping a SWIFTNet Remote Site

To stop a SWIFTNet Remote Site:

1. Right-click the name of the Site.
2. Deselect **Initiator mode (dynamic)**.

You may need to wait for up to 30 seconds for the Site to stop.

After stopping a Site, all client and server processes are stopped.

Alternatively, use the online command:

```
pelctl stop_site -a ABCD -im
```

Starting a SWIFTNet Remote Site

To start a SWIFTNet Remote Site:

1. Right-click the name of the Site.
2. Select **Initiator mode (dynamic)**.

You may need to wait for up to 30 seconds for the Site to start.

After starting a Site, all server processes are started. If required, client processes are started, to start acquiring queues.

Alternatively, use the online command:

```
pelctl start_site -a ABCD -im
```

Auto Deactivate

You can set a SWIFTNet Remote Site to automatically deactivate after a maximum number of attempts

- by configuring [monitor]stop_site_on_max_retry_reached parameter with peluconf command,
- or from Gateway Navigator

See section Configuring Gateway parameters > [Advanced parameters and Trace levels on page 156](#)

Number of retries

The `swift_job_retry_count_max` parameter in the `conf` file controls the number of retries in the case of a connection error with SAG. The value of this parameter is taken into account by the client and server processes. You can modify the value of this parameter if required by editing the [pelsetup.def](#) file.

Example: If the connection to SAG is broken, the client/server process retries to establish the connections as many times as specified by this parameter.

Transfer behavior

Consider a configuration with two SWIFTNet Sites:

- Nominal Site
- Backup Site

In this example, the following possible cases of transfer behavior exist:

Case	Nominal Site	Backup Site	Transfer behavior
1	Inactive	None	The transfer remains in the TO_BEGIN state. If you want the transfer to change to the CANCELED state, specify the latest start date when submitting the transfer.
2	Inactive	Defined (in nominal Site) and active	The transfer is made directly to the backup Site.
3	Inactive	Defined (in nominal Site) and inactive	Same behavior as case 1.
4	Inactive	Defined (in nominal Site) but the backup Site no longer exists because it has been deleted	Same behavior as case 1.

Static and Dynamic backup

Two types of backup situation exist:

Static backup

When a transfer is launched on the nominal Site but the Site is not active, the transfer is automatically started on the backup Site.

Prerequisites: The nominal Site has a backup Site and the backup Site is active.

Dynamic backup

When a transfer is launched on the nominal Site and the maximum number of retries `retry_count_max` (`-rc_max`) for this transfer has been reached, the transfer is automatically started on the backup Site.

Prerequisites: The nominal Site has a backup Site and the backup Site is active.

Axway Gateway: Connectors

Backup remote sites for SWIFTNet

[About backup Remote Sites for SWIFTNet](#)

[Backup site hierarchy limitation on page 2041](#)

[Defining a backup Site](#)

[Starting and stopping SWIFTNet Remote Sites](#)

[Transfer behavior](#)

[Static and Dynamic backup](#)

About backup Remote Sites for SWIFTNet

To ensure service continuity, you can define backup Sites for use with SWIFTNet transfers. If a transfer to a SWIFTNet Remote Site (the nominal Site) fails, for example if the connection with SWIFTNet is lost, Gateway will use the backup Site as an alternative.

Gateway can only use a backup Site if it is active. Depending on your requirements, you can start and stop backup Sites. If a transfer cannot be sent to the nominal Site or to a backup Site, it remains in the TO_BEGIN state until the connection with SWIFTNet is available.

Backup site hierarchy limitation

The backup Site hierarchy is limited to one. This means Gateway does not support a chain of backup Sites (more than one element in chain when the backup also has set a backup Site). Selecting a Site as backup for more than one nominal Sites is not supported.

One SWIFTNet remote sites can have up to 4 backup Sites selected.

See [Dynamically filtering duplicates for queues and output channels on page 2103](#)

Defining a backup Site

Define the backup Site when *creating or modifying* the nominal SWIFTNet Remote Site. You specify the alias of the backup Site in the nominal Site.

Prerequisite: The backup Site must exist (create the backup Site first).

To modify the nominal SWIFTNet Remote Site and define the backup Site:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
Gateway displays a list of all Remote Sites in the right pane.
3. Right-click the name of the Remote Site for which you want to modify the Site attributes and select **Modify...**

Gateway displays the *Remote Site* editing window.

4. Click the **SWIFTNet** tab.
The currently selected backup Sites are displayed in the Backup Sites list.
5. Click the **Change...** button next to the **Backup Sites** list.
6. In the newly created window, you can select up to 4 Sites as backup Sites.
Click **OK** to save the current selection and return to the Remote Site editing window.
7. Click **OK**.

Alternatively, use one of the following online commands:

- `peladm create_site -a ABCD -bks "BKPSITE1 BKPSITE2"` (create nominal Site)

- `peladm update_site -a ABCD -bks "BKPSITE1 BKPSITE2"` (update nominal Site)

Up to 4 backup Sites can be specified for a remote Site. Gateway checks that the specified backup Site aliases exists.

Starting and stopping SWIFTNet Remote Sites

Note: After restarting Gateway, all SWIFTNet Remote Sites are automatically started if both Initiator and Responder modes are set.

Stopping a SWIFTNet Remote Site

To stop a SWIFTNet Remote Site:

1. Right-click the name of the Site.
2. Deselect **Initiator mode (dynamic)**.

You may need to wait for up to 30 seconds for the Site to stop.

After stopping a Site, all client and server processes are stopped.

Alternatively, use the online command:

```
pelctl stop_site -a ABCD -im
```

Starting a SWIFTNet Remote Site

To start a SWIFTNet Remote Site:

1. Right-click the name of the Site.
2. Select **Initiator mode (dynamic)**.

You may need to wait for up to 30 seconds for the Site to start.

After starting a Site, all server processes are started. If required, client processes are started, to start acquiring queues.

Alternatively, use the online command:

```
pelctl start_site -a ABCD -im
```

Auto Deactivate

You can set a SWIFTNet Remote Site to automatically deactivate after a maximum number of attempts

- by configuring [monitor]stop_site_on_max_retry_reached parameter with peluconf command,
- or from Gateway Navigator

See section Configuring Gateway parameters > [Advanced parameters and Trace levels on page 156](#)

Number of retries

The `swift_job_retry_count_max` parameter in the `conf` file controls the number of retries in the case of a connection error with SAG. The value of this parameter is taken into account by the client and server processes. You can modify the value of this parameter if required by editing the [pelsetup.def](#) file.

Example: If the connection to SAG is broken, the client/server process retries to establish the connections as many times as specified by this parameter.

Transfer behavior

Consider a configuration with two SWIFTNet Sites:

- Nominal Site
- Backup Site

In this example, the following possible cases of transfer behavior exist:

Case	Nominal Site	Backup Site	Transfer behavior
1	Inactive	None	The transfer remains in the TO_BEGIN state. If you want the transfer to change to the CANCELED state, specify the latest start date when submitting the transfer.
2	Inactive	Defined (in nominal Site) and active	The transfer is made directly to the backup Site.
3	Inactive	Defined (in nominal Site) and inactive	Same behavior as case 1.

Case	Nominal Site	Backup Site	Transfer behavior
4	Inactive	Defined (in nominal Site) but the backup Site no longer exists because it has been deleted	Same behavior as case 1.

Static and Dynamic backup

Two types of backup situation exist:

Static backup

When a transfer is launched on the nominal Site but the Site is not active, the transfer is automatically started on the backup Site.

Prerequisites: The nominal Site has a backup Site and the backup Site is active.

Dynamic backup

When a transfer is launched on the nominal Site and the maximum number of retries `retry_count_max` (`-rc_max`) for this transfer has been reached, the transfer is automatically started on the backup Site.

Prerequisites: The nominal Site has a backup Site and the backup Site is active.

Axway Gateway: Connectors

Using pre/post transfer compression for SWIFTNet

The Gateway SWIFTNet connector supports compression for outgoing transfers and decompression for incoming transfers.

You can compress files or messages before sending them for reasons of efficiency, or compatibility with your SWIFTNet partner. The files or messages then need to be decompressed using the same algorithm at the receiving end.

Compression

By default, Gateway uses ZIP compression when sending a file or message. Alternatively, you can choose to compress with GZIP or ZLIB. [Manual compression](#) (using algorithms other than ZIP, GZIP or ZLIB) is also possible.

Decompression

When receiving a file, Gateway supports ZIP, ZLIB and GZIP. If a received file is compressed using one of these compression algorithms, the received information can be decompressed. For other algorithms, [manual decompression](#) is possible.

Requesting compression

You can request compression for a transfer in two different ways:

- In the **Trading Partner** object, using the [Compress](#) parameter (or [compress](#) if you use online commands)
- In the **Transfer Request** object, when submitting the transfer, using the [Compress](#) parameter (or [trade compress](#) if you use online commands)

File naming on Gateway

After compression or decompression, Gateway creates a new file in the directory where the files are stored. The new file has the following format:

After...	New file name
Compression	transfer_id.cmp (for example, F0000006.cmp)
Decompression	transfer_id.dcm (for example, F0000006.dcm)

By default, the files are stored in the directory `<Gateway installation directory>/run_time/tmp`. You can change the default directory for incoming files in the Remote Site.

FileAct transfers

If a transfer is made *without* compression, the **Sw:FileInfo** field in the request contains **SwCompression=None**.

If a transfer is made *with* compression, the file is compressed using ZIP (by default) and the **Sw:FileInfo** field in the request contains **SwCompression=Zip**.

To compress with GZIP or ZLIB, set **SwCompression=GZIP** or **SwCompression=ZLIB** in the [File information](#) ([file info](#)) parameter of the Trading Partner or in the [rcv msg](#) parameter when submitting a transfer.

InterAct transfers

If a transfer is made *without* compression, there are no changes to the request that is sent.

If a transfer is made *with* compression, Gateway sets compression attributes in the **RequestPayload/ ResponsePayload** tag. The compression attributes have the form: dt="Zip-Base64". For example, <RequestPayload dt="Zip-Base64">.

Gateway encodes the compressed message in base 64 and carries out the following operations:

- When sending a message: compression with ZIP (by default) and conversion to base 64
- When receiving a message: conversion from base 64 and decompression (if the compression algorithm is known - ZIP, ZLIB or GZIP)

To compress using GZIP or ZLIB, you must specify it in the Transfer Request when starting a transfer. In the [HTTP/FTP](#) tab, **HTTP CGI parameters/ FTP user command** field, enter dt="\GZIP-Base64\" or dt="\ZLIB-Base64\".

Alternatively, if you are using online commands, set the [ftp command](#) parameter to dt="\GZIP-Base64\" or dt="\ZLIB-Base64\".

Note: Gateway supports compression/decompression for embedded payload.

Manual compression and decompression

Gateway supports manual compression and decompression for algorithms other than ZIP, GZIP or ZLIB. You have to compress the file/message before it is sent and decompress the file/message after it is received.

Sending a message or a file using an algorithm other than ZIP, GZIP or ZLIB

FileAct: Before sending the file, compress it using your specific compression algorithm. Specify the algorithm in the Trading Partner object or Transfer Request object as described in [FileAct transfers](#).

InterAct: Before sending the message, compress it using your specific compression algorithm. Carry out a normal message transfer as you would for a message that is not compressed.

Receiving a message or a file that it compressed using an unknown compression algorithm

For both FileAct and InterAct, after a file/message is received with an unknown compression algorithm, an information message stating that the compression type is not supported is written to the log and the transfer ends normally. You then need to decompress the file/message.

Axway Gateway: Connectors

Possible Duplicate management for SWIFTNet

[About Possible Duplicate management](#)

[Overview of parameters for SWIFTNet Possible Duplicate](#)

[Specifying Possible Duplicate parameters](#)

[Using online commands](#)

[GikAPI](#)

[Possible Duplicate management log messages](#)

About Possible Duplicate management

When transferring files (or messages) it can happen that a file is transferred more than once. Gateway enables you to manage SWIFTNet Possible Duplicate transfers in real-time mode, by setting various parameters.

For outgoing SWIFTNet transfers, Gateway enables you (the initiator) to warn the responder that a file or message is a *Possible Duplicate*. To do this, you include Possible Duplicate parameters when creating a Transfer Request. The responder must then save these parameters and decide whether to keep the original file or use the new file.

For incoming transfers, a configuration parameter enables you to control the acceptance or rejection of Possible Duplicates.

This function is also known as PDE (Possible Duplicate Emission).

Overview of parameters for SWIFTNet Possible Duplicate

To manage situations in which Possible Duplicate transfers may occur, you need to:

- Specify certain transfer parameters
- Set a configuration parameter

Transfer parameters

For outgoing transfers, the following optional parameters in the Transfer Request are used to indicate a Possible Duplicate:

- SWIFTNet **Message id** parameter
If provided explicitly by the user, the value of this parameter overwrites the message id created by Gateway in the Transfer Request. The UUID should be used.
- **Possible duplicate** parameter
The initiator of a transfer can set this parameter to make the responder aware that the incoming transfer may be a duplicate of another transfer

These parameters are used for both FileAct and InterAct transfers. They can be exploited by Decision Rules, Models and end-of-transfer procedures.

An additional parameter is available when creating Transfer Requests using the peltrans command:

- Empty message id parameter
Sometimes the sender might want to mark a transfer as a Possible Duplicate but *not* generate a message id for it. In such cases the empty message id parameter should be used to prevent Gateway from automatically generating a message id. For details of how to use this parameter, refer to [Using online commands](#) below.

Configuration parameter

For incoming FileAct transfers, the configuration parameter **Accept possible duplicate transfers** controls the acceptance/rejection of Possible Duplicates. Depending on the value of this parameter, Gateway behaves in two different ways:

1. **Accept possible duplicate transfers** *not* selected: Gateway rejects Possible Duplicate transfers

Gateway checks if the incoming request corresponds to a request in its Mailbox.

The fields used to identify a request are: message id, requester DN, responder DN, service, request type and protocol file name, SWIFT LogicalName. If this original request exists, Gateway rejects the duplicate request (returning "duplicated" to the request initiator). Otherwise it accepts the transfer.

2. **Accept possible duplicate transfers** *selected*: Gateway accepts Possible Duplicate transfers

Gateway checks its Mailbox to see if a transfer exists with the same parameters as mentioned above. If one is found, Gateway accepts it and warns the user about the possible duplicate transfer via a log message and the **Possible duplicate** parameter

This parameter affects FileAct transfers only. In InterAct, no validation is made (all messages are accepted).

The incoming SWIFTNet Possible Duplicate parameters (possible duplicate indication and SWIFTNet message id) are used to set the corresponding Gateway parameters (at responder), in order to be exploited by Decision Rules, Models and end-of-transfer procedures.

SWIFTNet message id

In this version of Gateway, the SWIFTNet message id and the transfer identifier (`xfer_ident`) are set to the same value.

For *outgoing* SWIFTNet requests, both parameters are set to a Universal Unique Identifier (UUID) calculated by Gateway. The message id is no longer set to the local transfer identifier. This greatly reduces the risk of having two outgoing requests with the same message id in the receiver Mailbox.

For *incoming* SWIFTNet requests, the SWIFTNet message id and the transfer identifier are set to the value of the SWIFTNet <Sw:MsgId> tag.

Specifying Possible Duplicate parameters

Transfer Request parameters

You can set the SWIFTNet message id of the new transfer and the Possible Duplicate transfer indicator in the Transfer Request.

In the **SWIFTNet** tab of the *New Transfer Request* window, on the **E2E** (End To End) tab, complete the following parameters:

- **Message id** – Set the SWIFTNet message id of the new transfer to the message id of the initial transfer

- **Possible duplicate** – Select this option

These parameters are received by the responder. They can be exploited by Decision Rules, exits, etc.

You can view these Transfer Request parameters in the **SWIFTNet** tab of the *Transfer Request (Read only)* window.

Setting the configuration parameter for accepting/rejecting Possible Duplicates

You can enable or disable the option to accept SWIFTNet Possible Duplicate transfers (for FileAct) in the Gateway configuration menu.

1. In the [configuration menu](#), select:
Connectivity > Axway Connectivity > SWIFTNet
2. Check or uncheck the option **Accept possible duplicate transfers**.
3. Click OK.

The modification is taken into account only after restarting Gateway.

New attributes for Models and Decision Rules

New attributes for use with Models and Decision Rules:

- **swift message id** Type: string.
This corresponds to the SWIFTNet transfer identifier.
Related symbolic variable: **&(sw_message_id)**
- **possible duplicate** Type: enum (Y, y, N, n).
This corresponds to the Transfer Request **Possible duplicate** parameter.
Related symbolic variable: **&(sw_pd_indication)**

Model attributes for Transfer Requests

Examples of arguments (and values) for the General Model:

- **possible duplicate (Y)**
- **swift message id (<message id>)**

Decision Rules attributes for incoming/outgoing transfers

Examples of arguments for Decision Rules:

- **possible duplicate = N**
- **protocol = SWIFTNET**
- **swift message id = <message id>**

Using online commands

Setting Transfer Request parameters using the peltrans command

Use the `peltrans` command to create a Transfer Request and indicate a possible duplicate transfer. Refer to the following table for parameter details.

Transfer Request parameters specific to Possible Duplicate

In this table, the corresponding field name (if available) in the Gateway GUI is shown in *italics*.

Parameter	Meaning	Value
<code>-sw_empty_message_id (-swemptymsgid)</code> (None)	Empty message id parameter. Use this parameter to prevent Gateway from automatically generating a message id. This is useful if you want to mark a transfer as a Possible Duplicate but <i>not</i> generate a message id for it, because it is unknown. Note: Do not use this parameter in combination with the <code>sw_message_id</code> parameter! If you specify both parameters, then the <code>sw_empty_message_id</code> parameter is not taken into account.	• <code>y</code> – Prevent Gateway from automatically generating a message id • <code>N</code> – Gateway automatically generates a message id (default)

Parameter	Meaning	Value
-sw_message_id (-swmsgid) Message id	SWIFTNet message id. The value of this parameter overwrites the message id created by Gateway in the Transfer Request. Enter the SWIFTNet message id of the new transfer. This should be a Universal Unique Identifier (UUID). If the transfer is a Possible Duplicate, enter the message id of the initial transfer.	Maximum: 40 alphanumeric characters
-sw_pd_indication (-swpdind) Possible duplicate	Possible Duplicate indicator	• Y – Select this value to indicate to the responder that the transfer is a Possible Duplicate. • N (default)

For details of all other Transfer Request parameters, refer to [Transfers: Parameters List](#).

Examples

```

peltrans -da REMOTE_SITE -tradeoa LOCAL_TP -tradedata REMOTE_TP
-ap FTP_A
    -fc c:\info.txt -swmsgid 467246F2-5C7E-1D36-302A-6B4100000017
peltrans -da REMOTE_SITE -tradeoa LOCAL_TP -tradedata REMOTE_TP
-ap FTP_A
    -fc c:\info.txt -swpdind Y
peltrans -da REMOTE_SITE -tradeoa LOCAL_TP -tradedata REMOTE_TP -
ap FTP_A
    -fc c:\info.txt -swmsgid 467246F2-5C7E-1D36-302A-6B4100000017
-swpdind N

```

Setting Transfer Request parameters using peladm create_model or peladm update_model

You can add the two attributes **swift message id** and **possible duplicate** to general Models and/or SWIFTNet Models using the `swmsgid` and `swpdind` parameters.

Examples

General Model:

```
peladm create_model -ml PD_MODEL -pr GENERAL -swpdind Y  
-swmsgid "&(sw_message_id)"
```

SWIFTNet Model:

```
peladm create_model -ml PD_MODEL -pr SWIFTNET -swpdind Y  
-swmsgid "&(sw_message_id)"
```

Setting Transfer Request parameters in Decision Rules using peladm create_decisionrule or peladm update_decisionrule

The attribute names for use in Decision Rules are: `-sw_pd_indication` (possible duplicate) and `-sw_message_id` (SWIFTNet message id).

Example

```
peladm create_decisionrule -n PD_DECRULE -type XFER_CHANGE_STATE  
-et MODEL -ml PD_MODEL -ca "sw_pd_indication=Y;  
-sw_message_id=467246F2-5C7E-1D36-302A-6B4100000019;  
protocol=SWIFTNET"
```

GikAPI

Two new parameters have been added to the `gikapic.h` header to allow your applications to set the possible duplicate flag and overwrite the message id with a Universal Unique Identifier (UUID):

- GIK_T_SW_PD_INDICATION:
 - int: use GikSet/GetInt;
 - values: ITP_NO / ITP_YES;
- GIK_T_SW_MESSAGE_ID:
 - string: use GikSet/GetString;
 - maximum size: 40 characters

Possible Duplicate management log messages

Log messages (SNET102E and SNET103W) inform you about possible duplicate transfers in the following situations:

- The initiator sends a Transfer Request and gets an answer from the responder that the transfer was rejected because it is a duplicate. For example:
RSWIFT(DCmp) [29] Server rejected the transfer: duplicated request.
- The responder receives a possible duplicate transfer and the configuration option **Accept possible duplicate transfers** is selected: a warning message is displayed and the transfer is accepted. For example:
RSWIFT(DCmp) Duplicated incoming request, with message id "467246F2-5C7E-1D36-302A-6B4100000017", accepted. The configuration option "accept_possible_duplicate" is set.
Otherwise (if **Accept possible duplicate transfers** is not selected), an error message is displayed and the transfer is rejected. For example:
RSWIFT(DCmp) Duplicated incoming request, with message id "467246F2-5C7E-1D36-302A-6B4100000017", rejected.

Axway Gateway: Connectors

SWIFTNet state transitions

[About SWIFTNet state transitions in Gateway](#)[State transitions in current version of Gateway](#)[State transitions in Gateway 6.11](#)[Backwards compatibility](#)[State transitions for duplicate transfers](#)

About SWIFTNet state transitions in Gateway

In Gateway, final state transitions for SWIFTNet transfers are different from those in other protocols. This is because SWIFTNet is not really a protocol implemented in Gateway but a connector to the SWIFT network. The internal states of SWIFTNet transfers are fundamentally different and more complex than for the other protocols supported by Gateway. This is why an internal state has been chosen for each relevant step of the internal SWIFTNet transitions.

The SWIFTNet way of handling transfers is quite different from that used by the protocols originally implemented in Gateway. Experience has shown that the SWIFTNet state transition behavior in Gateway 6.11 was not ideal. Therefore from Gateway 6.12 onwards some of the SWIFTNet state transitions have been changed to be more coherent. Note however that the behavior is not backward compatible with 6.11. This topic explains the main differences.

This topic also explains the special case of state transitions for duplicate transfers.

State transitions in current version of Gateway

In theory, all transfers that do not require an acknowledgement should end in the ENDED state. However, due to the particularities of SWIFTNet this is not always the case.

Transfers that do require acknowledgement only use the ENDED_TO_ACK and ACKNOWLEDGED states.

FileAct / Real Time

All transfers (incoming and outgoing) that do not require an acknowledgement remain in the ENDED state.

Transfers (incoming and outgoing) that require acknowledgement do not use the ENDED state. They remain in the ENDED_TO_ACK state until the acknowledgement is received or sent and then they switch to ACKNOWLEDGED or NACK BY XFB/ NACK BY USER.

FileAct / Store-and-Forward

For Store-and-Forward messages, the acknowledgement is mandatory and does not have the same meaning as for Real Time. In this case the acknowledgement should not be used for business logic but only to inform the SWIFT network that the file has been safely stored. This is why all incoming transfers skip the ENDED state and go directly to ENDED_TO_ACK and then to ACKNOWLEDGED.

For outgoing transfers the situation is slightly different. If the sender does not require acknowledgement it does not wait for it and SWIFT will not send a delivery notification after the receiver has successfully retrieved the file. This means that Store-and-Forward transfers that do not require acknowledgement simply remain in the ENDED state. Store-and-Forward transfers that do require acknowledgement go directly to ENDED_TO_ACK and then to ACKNOWLEDGED.

State transitions in Gateway 6.11

Outgoing

For outgoing transfers there is no difference in behavior between Gateway 6.11 and this version of Gateway.

Incoming FileAct/ Real Time

There are differences for incoming FileAct/ Real Time transfers that require acknowledgement and use **Receipt option** = *Ack by monitor* in the Trading Partner. In Gateway 6.11 these transfers used all the final states: ENDED, then ENDED_TO_ACK, and then ACKNOWLEDGED.

Incoming FileAct/ Store-and-Forward

There are also differences for all incoming FileAct/ Store-and-Forward transfers. In Gateway 6.11 these transfers went directly to ENDED state.

Backwards compatibility

Previous and current SWIFTNet transfer state transitions for incoming transfers

This table summarizes the different SWIFTNet transfer state transition behavior for incoming transfers between Gateway versions.

SWIFTNet message type and mode	Acknowledgement?	Transfer state	6.11	6.12 onwards
FileAct / Real Time	No Ack	E	YES	YES
		W then A	-	-
	Ack By Monitor	E	YES	-
		W then A	YES	YES
	Ack By User	E	-	-
		W then A	YES	YES
	No Ack	E	YES	-
		W then A	-	YES
FileAct / Store-and-Forward	Ack By Monitor	E	YES	-
		W then A	-	YES
	Ack By User	E	-	-
		W then A	-	YES
	No Ack	E	YES	-
		W then A	-	YES

Legend

YES = corresponding state is routed

Transfer states:

- E = Ended
- W = Ended_to_ack
- A = Acked

State transitions for duplicate transfers

State transitions for duplicate transfers depend on the global parameter `accept_possible_duplicate`. If this parameter is set to YES, then the Possible Duplicate indication is ignored and the transfer behaves as any normal transfer. If however this parameter is set to NO, then the state transitions behave as explained below.

FileAct / Real Time when `accept_possible_duplicate=NO`

Outgoing

Outgoing duplicate transfers that do *not* require acknowledgement remain in the CANCELLED state.

Outgoing duplicate transfers that do require acknowledgement stay in the CANCELLED state until the acknowledgement has been received. They then go to ACKNOWLEDGED.

Incoming

Incoming duplicate transfers are saved in the Mailbox. They behave as any normal FileAct / Real Time incoming transfer.

FileAct / Store-and-Forward when `accept_possible_duplicate=NO`

Outgoing

Outgoing duplicate transfers that do *not* require delivery notification remain in the ENDED state.

Outgoing duplicate transfers that do require delivery notification stay in the ENDED-TO-ACK state until the delivery notification arrives. They then go to NACK.

Incoming

Incoming duplicate transfers are simply rejected by Gateway if the original file exists. They do not appear as Mailbox entries.

Axway Gateway: Connectors

SWIFTNet Proof keeping

[About SWIFTNet Proof keeping](#)

[Enabling SWIFTNet Proof keeping](#)

[Proof files](#)

About SWIFTNet Proof keeping

The purpose of proof keeping is to permanently store the XML requests and responses that were exchanged for a transfer in order to have an additional proof in case a misunderstanding appears between the partners.

You can activate or deactivate the proof keeping feature and, if the feature is activated, configure the directory where the proof will be kept (a default directory is proposed). By default, this option is deactivated.

When the proof keeping feature is activated, for every InterAct or FileAct transfer, Gateway stores the request and the response as long as there is no error at the SNL layer. This means that for a certain request (client side) you receive a response from your partner and from a received request (server side) you provide a response to the sender.

Enabling SWIFTNet Proof keeping

Enable SWIFTNet proof keeping in the configuration menu.

1. In the left pane of the GUI main window, right-click the Gateway server for which you want to enable proof logging and select **Configure** from the context menu.
The *Configuration* window is displayed.
2. In the left pane of the *Configuration* window, select **Connectivity > Axway Connectivity > SWIFTNet**.
For full details of the following parameters, refer to [Configuration: Connectors](#).
3. Activate (or deactivate) **Proof keeping**.
4. Specify the destination folder in **Default proof directory**. Gateway environment variables can be used here:
 - %VAR% and "\" separator for Windows
 - \$VAR and "/" separator for UNIX

If it is not possible to write to the specified directory, the <Gateway installation directory>/run_time/tmp directory is used. A corresponding warning message is output to client and server trace files. If other writing or file permission problems occur, an error trace message is generated.

5. Click **Activate** to enable the connector.
6. Restart Gateway for the change in configuration to be taken into account.

Proof files

Each complete transfer (FileAct or InterAct) will have its own proof file (InterAct has two entries in the Mailbox for one transfer but it is stored in one proof file).

The proof file name depends on whether the transfer is FileAct or InterAct.

Proof files for FileAct

GatewayXferLocalIdent_Sw:TransferRef

Example: FileAct RealTime

Send a file (Sw:PutFileRequest)

Gateway transfer id: 101

Sw:TransferRef: SNL01568XX1192006337083316C

In this case, the proof file name will be: 101_SNL01568XX1192006337083316

Proof files for InterAct

GatewayXferLocalIdent_SwInt:SwiftRef

Example: InterAct RealTime

(SwInt:ExchangeRequest)

Gateway transfer id: 221

SwInt:SwiftRef: SNL01568-2007-10-10T08:25:16.4912.000012Z

In this case, the proof file name will be: 221_SNL01568-2007-10-10T0825164912000012Z

Acknowledgments

For a FileAct transfer, in order to be sure that the transfer was completed correctly, the Sw:HandleFileEventRequest with the Completed status will also be stored.

The acknowledgment for a specific FileAct RealTime transfer is added in the file that already contains the proof for that transfer, if the file exists. For example, for a file transfer where an ACK was requested, the proof file contains information regarding the transfer. If an ACK arrives for this transfer, the ACK information will be added at the end of the same file.

Changed in 6.15.0. For InterAct and FileAct Store-and-Forward transfers, the acknowledgment (delivery notification) is added in the proof file associated with the RECEIPT mailbox entry.

Axway Gateway: Connectors

SWIFTNet dump to XML

Gateway dumps certain key elements from the communication with SWIFT to the `sw_add_param` XML file. This topic contains examples of files for the following elements:

- [HeaderInfo](#)
- [SignatureList](#)
- [Recipient List](#)
- [File or Message Copy](#)
- [Delivery notification \(SnF\)](#)
- [Delivery notification \(RealTime\)](#)
- [Authorization System Message \(xsys.002.001.01\)](#)
- [Refusal System Message \(xsys.003.001.01\)](#)
- [Delivery Notification System Message \(xsys.011.001.01\)](#)
- [Failed Delivery Notification System Message \(xsys.012.001.01\)](#)

Example of XML file containing HeaderInfo

Gateway supports the service profiles supported by SWIFT.

```
<AddSwParams>
  <SwHeaderInfo>
    <Out>
      <ApplSpfcfc xmlns="urn:swift:xsd:ApplSpfcfc.TxsCntr.01">
        <TxsCntr>
          <TtlNbOfTxs>650</TtlNbOfTxs>
        </TxsCntr>
      </ApplSpfcfc>
    </Out>
  </SwHeaderInfo>
</AddSwParams>

<AddSwParams>
  <SwHeaderInfo>
    <Out>
      <ApplSpfcfc xmlns="urn:swift:xsd:ApplSpfcfc.PmtSummry.01">
        <PmtSummry>
          <TtlIntrBkSttlmAmt Ccy="RON">768900</TtlIntrBkSttlmAmt>
          <TtlNbOfTxs>1467</TtlNbOfTxs>
          <CdtDbtInd>CRDT</CdtDbtInd>
          <IntrBkSttlmDt>2008-05-09</IntrBkSttlmDt>
          <FileId>OurFile</FileId>
        </PmtSummry>
      </ApplSpfcfc>
    </Out>
  </SwHeaderInfo>
</AddSwParams>

<AddSwParams>
  <SwHeaderInfo>
```

```
<Out>
<ApplSpcfc xmlns = "urn:swift:xsd:ApplSpcfc.BAM.01">
<FileTrfHdrBAM>
<File>
<Tp>terms</Tp>
<Nm> BankTermsConditions.pdf </Nm>
</File>
<File>
<Tp>passport</Tp>
<Nm> Passport.jpg </Nm>
</File>
</FileTrfHdrBAM>
</ApplSpcfc>
</Out>
</SwHeaderInfo>
</AddSwParams>

<AddSwParams>
<SwHeaderInfo>
<Out>
<ApplSpcfc xmlns = "urn:swift:xsd:ApplSpcfc.Trade.01">
<FileHdrTrfTrad>
<Doc>
<Nm>SiemensScannedDocs.zip</Nm>
<Ref>WQ667589A-20100519</Ref>
</Doc>
</FileHdrTrfTrad>
</ApplSpcfc>
</Out>
</SwHeaderInfo>
</AddSwParams>
```

Examples of XML files containing SignatureList

SignatureList – FileAct PUT Request – Initiator

```
<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
  <Out>
    <SwSec:SignatureList>
      <SwSec:Signature>
        <SwSec:KeyInfo>
          <SwSec:SignDN>AnyDN</SwSec:SignDN>
        </SwSec:KeyInfo>
        <SwSec:Manifest>
          <Sw:Reference>
```

```
<Sw:DigestRef>Sw.File.Put.Request and RND</Sw:DigestRef>
</Sw:Reference>
<Sw:Reference>
<Sw:DigestRef>Sw.FARequestHeader</Sw:DigestRef>
</Sw:Reference>
</SwSec:Manifest>
</SwSec:Signature>
</SwSec:SignatureList>
</Out>
</SwSignatureList>
</AddSwParams>
```

SignatureList – FileAct PUT Request – Responder

```
<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
  <Out>
    <SwSec:SignatureList>
      <SwSec:Signature>
        <SwSec:KeyInfo>
          <SwSec:SignDN>AnyDN</SwSec:SignDN>
        </SwSec:KeyInfo>
        <SwSec:Manifest>
          <Sw:Reference>
            <Sw:DigestRef>Sw.File.Put.Response</Sw:DigestRef>
          </Sw:Reference>
          <Sw:Reference>
            <Sw:DigestRef>Sw.FAResponseHeader</Sw:DigestRef>
          </Sw:Reference>
        </SwSec:Manifest>
      </SwSec:Signature>
    </SwSec:SignatureList>
  </Out>
</SwSignatureList>
</AddSwParams>
```

SignatureList – FileAct GET Request – Initiator

```
<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
  <Out>
    <SwSec:SignatureList>
      <SwSec:Signature>
        <SwSec:KeyInfo>
          <SwSec:SignDN>AnyDN</SwSec:SignDN>
        </SwSec:KeyInfo>
      </SwSec:Signature>
    </SwSec:SignatureList>
  </Out>
</SwSignatureList>
</AddSwParams>
```

```

</SwSec:KeyInfo>
<SwSec:Manifest>
<Sw:Reference>
<Sw:DigestRef>Sw.File.Get.Request</Sw:DigestRef>
</Sw:Reference>
<Sw:Reference>
<Sw:DigestRef>Sw.FARequestHeader</Sw:DigestRef>
</Sw:Reference>
</SwSec:Manifest>
</SwSec:Signature>
</SwSec:SignatureList>
</Out>
</SwSignatureList>
</AddSwParams>

```

SignatureList – FileAct GET Request – Responder

```

<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
    <Out>
      <SwSec:SignatureList>
        <SwSec:Signature>
          <SwSec:KeyInfo>
            <SwSec:SignDN>AnyDN</SwSec:SignDN>
          </SwSec:KeyInfo>
          <SwSec:Manifest>
            <Sw:Reference>
              <Sw:DigestRef>Sw.File.Get.Response</Sw:DigestRef>
            </Sw:Reference>
            <Sw:Reference>
              <Sw:DigestRef>Sw.FAResponseHeader</Sw:DigestRef>
            </Sw:Reference>
          </SwSec:Manifest>
        </SwSec:Signature>
      </SwSec:SignatureList>
    </Out>
  </SwSignatureList>
</AddSwParams>

```

SignatureList – FileAct SnF – Sender

```

<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
    <Out>
      <SwSec:SignatureList>

```



```

<SwSec:Signature>
<SwSec:KeyInfo>
<SwSec:SignDN>AnyDN</SwSec:SignDN>
</SwSec:KeyInfo>
<SwSec:Manifest>
<Sw:Reference>
<Sw:DigestRef>Sw.File.Put.Request</Sw:DigestRef>
</Sw:Reference>
<Sw:Reference>
<Sw:DigestRef>Sw.FARequestHeader</Sw:DigestRef>
</Sw:Reference>
</SwSec:Manifest>
</SwSec:Signature>
</SwSec:SignatureList>
</Out>
</SwSignatureList>
</AddSwParams>

```

SignatureList – FileAct SnF – Receiver

```

<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
  <Out>
  <SwSec:SignatureList>
  <SwSec:Signature>
  <SwSec:KeyInfo>
  <SwSec:SignDN>AnyDN</SwSec:SignDN>
  </SwSec:KeyInfo>
  <SwSec:Manifest>
  <Sw:Reference>
  <Sw:DigestRef>Sw.File.Put.Response</Sw:DigestRef>
  </Sw:Reference>
  <Sw:Reference>
  <Sw:DigestRef>Sw.FAResponseHeader</Sw:DigestRef>
  </Sw:Reference>
  </SwSec:Manifest>
  </SwSec:Signature>
  </SwSec:SignatureList>
  </Out>
  </SwSignatureList>
</AddSwParams>

```

SignatureList – FileAct ACK Request

```

<AddSwParams>

```

```
<SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
  <Out>
    <SwSec:SignatureList>
      <SwSec:Signature>
        <SwSec:KeyInfo>
          <SwSec:SignDN>AnyDN</SwSec:SignDN>
        </SwSec:KeyInfo>
        <SwSec:Manifest>
          <Sw:Reference>
            <Sw:DigestRef>Sw.File.Ack.Request</Sw:DigestRef>
          </Sw:Reference>
        </SwSec:Manifest>
      </SwSec:Signature>
    </SwSec:SignatureList>
  </Out>
</SwSignatureList>
</AddSwParams>
```

SignatureList – FileAct ACK Response

```
<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
    <Out>
      <SwSec:SignatureList>
        <SwSec:Signature>
          <SwSec:KeyInfo>
            <SwSec:SignDN>AnyDN</SwSec:SignDN>
          </SwSec:KeyInfo>
          <SwSec:Manifest>
            <Sw:Reference>
              <Sw:DigestRef>Sw.File.Ack.Response</Sw:DigestRef>
            </Sw:Reference>
          </SwSec:Manifest>
        </SwSec:Signature>
      </SwSec:SignatureList>
    </Out>
  </SwSignatureList>
</AddSwParams>
```

SignatureList – InterAct Request

```
<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
    <Out>
```

```

    <SwSec:SignatureList>
    <SwSec:Signature>
    <SwSec:KeyInfo>
    <SwSec:SignDN>AnyDN</SwSec:SignDN>
    </SwSec:KeyInfo>
    <SwSec:Manifest>
    <Sw:Reference>
    <Sw:DigestRef>Sw.RequestHeader and
RequestPayload</Sw:DigestRef>
    </Sw:Reference>
    </SwSec:Manifest>
    </SwSec:Signature>
    </SwSec:SignatureList>
  </Out>
</SwSignatureList>
</AddSwParams>

```

SignatureList – InterAct Response

```

<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwSec="urn:swift:snl:ns.SwSec">
    <Out>
    <SwSec:SignatureList>
    <SwSec:Signature>
    <SwSec:KeyInfo>
    <SwSec:SignDN>AnyDN</SwSec:SignDN>
    </SwSec:KeyInfo>
    <SwSec:Manifest>
    <Sw:Reference>
    <Sw:DigestRef>Sw.ResponseHeader and
ResponsePayload</Sw:DigestRef>
    </Sw:Reference>
    </SwSec:Manifest>
    </SwSec:Signature>
    </SwSec:SignatureList>
    </Out>
  </SwSignatureList>
</AddSwParams>

```

Examples of XML files containing Recipient List

RecipientList – FileAct SnF PUT Request – Initiator

This file also contains two Failed Delivery Notification Messages and one Delivery Notification Message.

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwInt="urn:swift:snl:ns.SwInt"
xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
<SwRecipientList>
<Out>
<Sw:RecipientList>
<Sw:RecipientDN>
cn=recipient1,ou=fr,o=bankfrpp,o=swift
</Sw:RecipientDN>
<Sw:RecipientDN>
cn=recipient2,ou=fr,o=bankfrpp,o=swift
</Sw:RecipientDN>
<Sw:RecipientDN>
cn=recipient3,ou=fr,o=bankfrpp,o=swift
</Sw:RecipientDN>
</Sw:RecipientList>
</Out>
</SwRecipientList>

<FailedDeliveryNotificationMessage>
<In>
<Doc:Document xmlns:Doc="urn:swift:xsd:xsys.012.001.01"
xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwInt="urn:swift:snl:ns.SwInt"
xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
<Doc:xsys.012.001.01>
<Doc:DlvryNtfctn>
<Sw:SnFRef>snp20005D21313673884370601</Sw:SnFRef>
<Sw:OrigSnFRef>SNL02026D11313674007922954</Sw:OrigSnFRef>
<Sw:RecipientDN>cn=recipient1,ou=fr,o=bankfrpp,o=swift</Sw:RecipientDN>
<Sw:SnFRefType>FileAct</Sw:SnFRefType>
<Sw:AcceptStatus>Rejected</Sw:AcceptStatus>
<Sw:DeliveryTime>2011-08-18T13:24:44Z</Sw:DeliveryTime>
<Sw:AckSwiftTime>2011-08-18T13:24:48Z</Sw:AckSwiftTime>
<Sw:AckDescription>Destination trading partner does not exist</Sw:AckDescription>
<Sw:FileRequestHeader>
<SwInt:Requestor>cn=recipient3,ou=fr,o=bankfrpp,o=swift</SwInt:Requestor>
<SwInt:Responder>cn=distributor,o=swift,o=swift</SwInt:Responder>
<SwInt:Service>swift.test.sf.iafa!x</SwInt:Service>
<SwInt:RequestType>pain.xxx.cfonbl70.dct</SwInt:RequestType>
<SwInt:Priority>Normal</SwInt:Priority>
</Sw:FileRequestHeader>
</Doc:DlvryNtfctn>
</Doc:xsys.012.001.01>

```

```

</Doc:Document>

<Doc:Document xmlns:Doc="urn:swift:xsd:xsys.012.001.01"
xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwInt="urn:swift:snl:ns.SwInt"
xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
<Doc:xsys.012.001.01>
<Doc:DlvryNtfctn>
<Sw:SnFRef>snp20005D21313673884371601</Sw:SnFRef>
<Sw:OrigSnFRef>SNL02026D11313674007922954</Sw:OrigSnFRef>
<Sw:RecipientDN>cn=recipient2,ou=fr,o=bankfrpp,o=swift</Sw:RecipientDN>
<Sw:SnFRefType>FileAct</Sw:SnFRefType>
<Sw:AcceptStatus>Rejected</Sw:AcceptStatus>
<Sw:DeliveryTime>2011-08-18T13:24:44Z</Sw:DeliveryTime>
<Sw:AckSwiftTime>2011-08-18T13:24:55Z</Sw:AckSwiftTime>
<Sw:AckDescription>Destination trading partner does not exist</Sw:AckDescription>
<Sw:FileRequestHeader>
<SwInt:Requestor>cn=recipient3,ou=fr,o=bankfrpp,o=swift</SwInt:Requestor>
<SwInt:Responder>cn=distributor,o=swift,o=swift</SwInt:Responder>
<SwInt:Service>swift.test.sf.iafa!x</SwInt:Service>
<SwInt:RequestType>pain.xxx.cfonb170.dct</SwInt:RequestType>
<SwInt:Priority>Normal</SwInt:Priority>
</Sw:FileRequestHeader>
</Doc:DlvryNtfctn>
</Doc:xsys.012.001.01>
</Doc:Document>
</In>
</FailedDeliveryNotificationMessage>

<DeliveryNotificationMessage>
<In>
<Doc:Document xmlns:Doc="urn:swift:xsd:xsys.011.001.01"
xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwInt="urn:swift:snl:ns.SwInt"
xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
<Doc:xsys.011.001.01>
<Doc:DlvryNtfctn>
<Sw:SnFRef>snp20005D21313673884372601</Sw:SnFRef>
<Sw:OrigSnFRef>SNL02026D11313674007922954</Sw:OrigSnFRef>
<Sw:RecipientDN>cn=recipient3,ou=fr,o=bankfrpp,o=swift</Sw:RecipientDN>
<Sw:SnFRefType>FileAct</Sw:SnFRefType>
<Sw:AcceptStatus>Accepted</Sw:AcceptStatus>
<Sw:DeliveryTime>2011-08-18T13:24:44Z</Sw:DeliveryTime>
<Sw:AckSwiftTime>2011-08-18T13:24:56Z</Sw:AckSwiftTime>
<Sw:FileRequestHeader>

```

```

<SwInt:Requestor>cn=recipient3,ou=fr,o=bankfrpp,o=swift</SwInt:Requestor>
<SwInt:Responder>cn=distributor,o=swift,o=swift</SwInt:Responder>
<SwInt:Service>swift.test.sf.iafa!x</SwInt:Service>
<SwInt:RequestType>pain.xxx.cfonbl70.dct</SwInt:RequestType>
<SwInt:Priority>Normal</SwInt:Priority>
</Sw:FileRequestHeader>
</Doc:DlvryNtfctn>
</Doc:xsys.011.001.01>
</Doc:Document>
</In>
</DeliveryNotificationMessage>
</AddSwParams>

```

RecipientList – FileAct SnF PUT Request – Responder

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
<SwRecipientList>
<In>
<Sw:RecipientList xmlns:Sw="urn:swift:snl:ns.Sw">
<Sw:RecipientDN>
cn=recipient1,ou=fr,o=bankfrpp,o=swift
</Sw:RecipientDN>
<Sw:RecipientDN>
cn=recipient2,ou=fr,o=bankfrpp,o=swift
</Sw:RecipientDN>
<Sw:RecipientDN>
cn=recipient3,ou=fr,o=bankfrpp,o=swift
</Sw:RecipientDN>
</Sw:RecipientList>
</In>
</SwRecipientList>
</AddSwParams>

```

Example of XML file containing File or Message Copy information

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
  <Sw:ThirdPartyToReceiverInformation
xmlns:Sw="urn:swift:snl:ns.Sw">
    <Sw:TPInfo>generic third party to receiver
information</Sw:TPInfo>
  </Sw:ThirdPartyToReceiverInformation>
</AddSwParams>

```

Example of XML file containing delivery notification (SnF) information

The Store-and-Forward delivery notification is deposited in the SWIFT parameter file. Both the incoming request and the outgoing reply are saved.

```
<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
  <SwSnFDeliveryNotification>
    <In>
      <Sw:HandleSnFRequest xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
        <Sw:SnFRequestHandle>
          <Sw:SnFRequestDescriptor>
            <SwInt:MRRResult>
              <SwInt:SNLId>snl01568</SwInt:SNLId>
              <SwInt:SNLEP>MC_EP</SwInt:SNLEP>
            </SwInt:MRRResult>
            <Sw:SnFOutputInfo>
              <Sw:SnFSessionId>bic1_
dev02fadn!x:p:000868</Sw:SnFSessionId>
              <Sw:SnFOutputSeq>2374</Sw:SnFOutputSeq>
              <Sw:DeliveryTime>2010-01-
12T15:47:32Z</Sw:DeliveryTime>
              <Sw:SnFInputTime>0101:2010-01-
12T15:47:31</Sw:SnFInputTime>
            </Sw:SnFOutputInfo>
          </Sw:SnFRequestDescriptor>
          <Sw:SnFOpRequestHandle>
            <Sw:NotifySnFRequestHandle>

<Sw:SnFRef>SNL01568D11263311013740630C</Sw:SnFRef>
              <Sw:SnFRefType>FileAct</Sw:SnFRefType>
              <Sw:AcceptStatus>Accepted</Sw:AcceptStatus>
              <Sw:AckSwiftTime>2010-01-
12T15:47:31Z</Sw:AckSwiftTime>
            </Sw:NotifySnFRequestHandle>
          </Sw:SnFOpRequestHandle>
        </Sw:SnFRequestHandle>
      </Sw:HandleSnFRequest>
    </In>
    <Out>
      <Sw:HandleSnFResponse xmlns:Sw="urn:swift:snl:ns.Sw"
xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec"
xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
        <SwSec:AuthorisationContext>
          <SwSec:UserDN>cn=xfer,o=bic2,o=swift</SwSec:UserDN>
        </SwSec:AuthorisationContext>
        <Sw:SnFResponse>
          <Sw:SnFResponseControl>
```

```

        <SwInt:NRIndicator>FALSE</SwInt:NRIndicator>
        <Sw:ProductList>
            <Sw:ProductInfo>
                <Sw:VendorName>Axway</Sw:VendorName>
                <Sw:ProductName>Gateway</Sw:ProductName>
                <Sw:ProductVersion>6130</Sw:ProductVersion>
            </Sw:ProductInfo>
        </Sw:ProductList>
    </Sw:SnFResponseControl>
    <Sw:SnFOpResponse>
        <Sw:NotifySnFResponse>
            <Sw:AckMessage>

<Sw:SnFRef>SNL01568D11263311013740630C</Sw:SnFRef>
                <Sw:SnFRefType>FileAct</Sw:SnFRefType>
                <Sw:AcceptStatus>Accepted</Sw:AcceptStatus>
            </Sw:AckMessage>
        </Sw:NotifySnFResponse>
    </Sw:SnFOpResponse>
</Sw:SnFResponse>
</Sw:HandleSnFResponse>
</Out>
</SwSnFDeliveryNotification>
</AddSwParams>

```

Example of XML file containing Real Time Delivery Notification

The Real Time delivery notification is deposited in the SWIFT parameter file. Both the incoming request and the outgoing reply are saved.

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
    <SwRTDeliveryNotification>
        <In>
            <Sw:HandleFileRequest xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec" xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
                <Sw:FileRequestHandle>
                    <Sw:FileRequestDescriptor>
                        <SwInt:SwiftRequestRef>SNL02026-2014-03-04T15:56:13.8716.005035Z</SwInt:SwiftRequestRef>
                        <Sw:SwiftTime>2014-03-04T15:40:29Z</Sw:SwiftTime>
                        <SwInt:MRRResult>
                            <SwInt:SNLId>snl02026</SwInt:SNLId>
                            <SwInt:SNLEP>swci_rtl</SwInt:SNLEP>
                        </SwInt:MRRResult>
                    </Sw:FileRequestDescriptor>
                <Sw:FileRequestE2EControl>
                    <Sw:MsgId>5315F3EC-3773-5D55-01F2-6F00000003DC</Sw:MsgId>
                </Sw:FileRequestE2EControl>
            </Sw:HandleFileRequest>
        </In>
    </SwRTDeliveryNotification>
</AddSwParams>

```



```

        <Sw:CreationTime>2014-03-04T15:40:28Z</Sw:CreationTime>
    </Sw:FileRequestE2EControl>
    <Sw:FileRequestHeader>
        <SwInt:Requestor>cn=rnda,ou=ro,o=ptsafrrk,o=swift</SwInt:Requestor>
        <SwInt:Responder>cn=rnd9,ou=ro,o=ptsafrrk,o=swift</SwInt:Responder>
        <SwInt:Service>swift.test.rt.iafa!x</SwInt:Service>
        <SwInt:RequestType>xsys.xxx.delnotif</SwInt:RequestType>
        <SwInt:Priority>Normal</SwInt:Priority>
    </Sw:FileRequestHeader>
    <Sw:FileOpRequestHandle>
        <Sw:AckFileRequestHandle>
            <Sw:TransferRef>SNL02026D11393948546273789S</Sw:TransferRef>
            <Sw:Digest>
                <Sw:DigestAlgorithm>SHA-256</Sw:DigestAlgorithm>
                <Sw:DigestValue>ypeBESobvcr6wjGzmiPcTaeG7/gUfE5yuYB3ha/uSLs=</Sw:Di
gestValue>
            </Sw:Digest>
            <Sw:Accepted>TRUE</Sw:Accepted>
        </Sw:AckFileRequestHandle>
    </Sw:FileOpRequestHandle>
</Sw:FileRequestHandle>
</Sw:HandleFileRequest>
</In>
<Out>
    <Sw:HandleFileResponse xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwSec="urn:swift:snl:ns.SwSec" xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
        <SwSec:AuthorisationContext>
            <SwSec:UserDN>cn=xfer2,o=ptsafrrk,o=swift</SwSec:UserDN>
        </SwSec:AuthorisationContext>
        <Sw:FileResponse>
            <Sw:FileResponseControl>
                <SwInt:ResponseCrypto>FALSE</SwInt:ResponseCrypto>
                <SwInt:NRIndicator>FALSE</SwInt:NRIndicator>
                <Sw:ProductList>
                    <Sw:ProductInfo>
                        <Sw:VendorName>Axway</Sw:VendorName>
                        <Sw:ProductName>Gateway</Sw:ProductName>
                        <Sw:ProductVersion>6150IR14</Sw:ProductVersion>
                    </Sw:ProductInfo>
                </Sw:ProductList>
            </Sw:FileResponseControl>
            <Sw:FileResponseE2EControl>
                <Sw:MsgId>5315F3EC-3773-5D55-01F2-6F00000003DC</Sw:MsgId>
                <Sw:CreationTime>2014-03-04T15:40:30Z</Sw:CreationTime>
            </Sw:FileResponseE2EControl>
            <Sw:FileResponseHeader>
                <SwInt:Responder>cn=rnd9,ou=ro,o=ptsafrrk,o=swift</SwInt:Responder>
            </Sw:FileResponseHeader>

```

```

        <Sw:FileOpResponse>
            <Sw:AckFileResponse/>
        </Sw:FileOpResponse>
    </Sw:FileResponse>
</Sw:HandleFileResponse>
</Out>
</SwRTDeliveryNotification>
</AddSwParams>

```

Example of XML file containing an Authorization Notification system message

The Authorization Notification system message is deposited in the SWIFT parameter file.

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
    <YCopyAuthorisationNotification>
        <In>
            <Doc:Document xmlns:Doc="urn:swift:xsd:xsys.002.001.01" xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwGbl="urn:swift:snl:ns.SwGbl" xmlns:SwSec="urn:swift:snl:ns.SwSec">
                <Doc:xsys.002.001.01>
                    <Doc:AuthstnNtfctn>
                        <Sw:SnFRef>SNL02026D11392743452505115</Sw:SnFRef>
                        <Sw:FileRequestHeader>
                            <SwInt:Requestor>cn=rnd9,ou=ro,o=ptsafrrk,o=swift</SwInt:Requestor>
                            <SwInt:Responder>cn=rnda,ou=ro,o=ptsafrrk,o=swift</SwInt:Responder>
                            <SwInt:Service>swift.qualif.ycopf!x</SwInt:Service>
                            <SwInt:RequestType>admi.xxx.fa.nro.hdo</SwInt:RequestType>
                            <SwInt:Priority>Normal</SwInt:Priority>
                        </Sw:FileRequestHeader>
                        <Sw:ThirdPartyToSenderInformation>
                            <Sw:TPInfo>generic 3rd party to sender information</Sw:TPInfo>
                        </Sw:ThirdPartyToSenderInformation>
                    </Doc:AuthstnNtfctn>
                </Doc:xsys.002.001.01>
            </Doc:Document>
        </In>
    </YCopyAuthorisationNotification>
</AddSwParams>

```

Example of XML file containing a Refusal Notification system message

The Refusal Notification system message is deposited in the SWIFT parameter file.

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<AddSwParams>
  <YCopyRefusalNotification>
    <In>
      <Doc:Document xmlns:Doc="urn:swift:xsd:xsys.003.001.01" xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwGbl="urn:swift:snl:ns.SwGbl" xmlns:SwSec="urn:swift:snl:ns.SwSec">
        <Doc:xsys.003.001.01>
          <Doc:AuthstnRfslNtfctn>
            <Sw:SnFRef>SNL00112D10045533050123736</Sw:SnFRef>
            <Sw:FileRequestHeader>
              <SwInt:Requestor>cn=rnd9,ou=ro,o=ptsafrrk,o=swift</SwInt:Requestor>
              <SwInt:Responder>cn=rnda,ou=ro,o=ptsafrrk,o=swift</SwInt:Responder>
              <SwInt:Service>swift.qualif.ycopf!x</SwInt:Service>
              <SwInt:RequestType>admi.xxx.ia.nro.hdo</SwInt:RequestType>
            </Sw:FileRequestHeader>
          </Doc:AuthstnRfslNtfctn>
        </Doc:xsys.003.001.01>
      </Doc:Document>
    </In>
  </YCopyRefusalNotification>
</AddSwParams>

```

Example of XML file containing a Delivery Notification system message

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
  <DeliveryNotificationMessage>
    <In>
      <Doc:Document xmlns:Doc="urn:swift:xsd:xsys.011.001.01" xmlns:Sw="urn:swift:snl:ns.Sw" xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
        <Doc:xsys.011.001.01>
          <Doc:DlvryNtfctn>
            <Sw:SnFRef>snpl0005D11392632259369601</Sw:SnFRef>
            <Sw:OrigSnFRef>SNL02026D11392633171168895</Sw:OrigSnFRef>
            <Sw:RecipientDN>cn=rnda,ou=ro,o=ptsafrrk,o=swift</Sw:RecipientDN>
            <Sw:SnFRefType>FileAct</Sw:SnFRefType>
            <Sw:AcceptStatus>Accepted</Sw:AcceptStatus>
            <Sw:DeliveryTime>2014-02-17T10:17:39Z</Sw:DeliveryTime>
            <Sw:AckSwiftTime>2014-02-17T10:17:48Z</Sw:AckSwiftTime>
            <Sw:FileRequestHeader>
              <SwInt:Requestor>cn=rnd9,ou=ro,o=ptsafrrk,o=swift</SwInt:Requestor>
              <SwInt:Responder>cn=distributor,o=swift,o=swift</SwInt:Responder>
              <SwInt:Service>swift.test.sf.iafa!x</SwInt:Service>
              <SwInt:RequestType>admi.xxx.fa.nro.hdo</SwInt:RequestType>
              <SwInt:Priority>Normal</SwInt:Priority>
            </Sw:FileRequestHeader>
          </Doc:DlvryNtfctn>
        </Doc:xsys.011.001.01>
      </Doc:Document>
    </In>
  </DeliveryNotificationMessage>
</AddSwParams>

```

```

    </In>
  </DeliveryNotificationMessage>
</AddSwParams>

```

Example of XML file containing a Failed Delivery Notification

```

<?xml version="1.0" encoding="UTF-8"?>
<AddSwParams>
  <FailedDeliveryNotificationMessage>
    <In>
      <Doc:Document xmlns:Doc="urn:swift:xsd:xsys.012.001.01" xmlns:Sw="urn:swift:snl:ns.S
w" xmlns:SwInt="urn:swift:snl:ns.SwInt" xmlns:SwGbl="urn:swift:snl:ns.SwGbl">
        <Doc:xsys.012.001.01>
          <Doc:DlvryNtfctn>
            <Sw:SnFRef>snpl0005D11391767405984600</Sw:SnFRef>
            <Sw:OrigSnFRef>SNL02026D11391768304001932</Sw:OrigSnFRef>
            <Sw:RecipientDN>cn=rndbad,ou=ro,o=ptsafrrk,o=swift</Sw:RecipientDN>
            <Sw:SnFRefType>FileAct</Sw:SnFRefType>
            <Sw:AcceptStatus>Rejected</Sw:AcceptStatus>
            <Sw:DeliveryTime>2014-02-07T10:03:26Z</Sw:DeliveryTime>
            <Sw:AckSwiftTime>2014-02-07T10:03:28Z</Sw:AckSwiftTime>
            <Sw:AckDescription>Originator trading partner does not exist</Sw:AckDescriptio
n>
            <Sw:FileRequestHeader>
              <SwInt:Requestor>cn=rnd9,ou=ro,o=ptsafrrk,o=swift</SwInt:Requestor>
              <SwInt:Responder>cn=distributor,o=swift,o=swift</SwInt:Responder>
              <SwInt:Service>swift.test.sf.iafa!x</SwInt:Service>
              <SwInt:RequestType>admi.xxx.fa.nro.hdo</SwInt:RequestType>
              <SwInt:Priority>Normal</SwInt:Priority>
            </Sw:FileRequestHeader>
          </Doc:DlvryNtfctn>
        </Doc:xsys.012.001.01>
      </Doc:Document>
    </In>
  </FailedDeliveryNotificationMessage>
</AddSwParams>

```

Restart a SAG

This procedure explains the steps to restart a SAG (SWIFT Alliance Gateway):

1. Before shutting down the SAG/SNL, stop the active SWIFTNet Remote Sites linked to this SAG/SNL.
2. After the SAG/SNL is restarted and their status is up and running, restart the previously stopped SWIFTNet remote sites.

3. Once the SWIFTNet remote sites are restarted, check the status of the SnF queue.

Use the command `pelcsp status_swoutputsession`

See [Axway Gateway: Connectors](#) on page 2089

4. If a queue status is not acquired, then either:

- restart the related SWIFTNet remote site, or
- use the Gateway command to do the reacquisition, such as:

```
pelctl acquire_swqueue -qn queue_name -si remote_site -f
```

Axway Gateway: Connectors

SWIFTNet input channels

About SWIFTNet input channels

Input channels are designed to guarantee that the store-and-forward traffic sent to the SWIFT network respects a particular sequence.

SWIFT 7.0 only supports the sending of *Interact* messages through input channels.

It is not necessary to use input channels if the business applications exchanging messages already support the exchange of non-sequenced traffic.

Creating input channels

Gateway automatically creates a default input channel for the BIC. If required, create additional input channels using the `pelctl create_input_channel` command. Make sure that you respect the SWIFT naming conventions for input channels. For example, input channel names must be in lower case. Gateway does not check or correct the name or case of the channel. If the name is not SWIFT-compliant it will be rejected by SWIFT.

It is possible to restrict the creation of input channels. If user security is activated, only a user who has administrator control of the Remote Site used for the creation will be able to create the input channel.

Security considerations relating to input channels

See *Security Guide > Security best practices > [Identified threats and possible mitigations](#)* for further information (requires login).

Creating an input channel using command line

```
pelctl create_input_channel
-input_channel (-ic) CHANNELNAME
-site (-si) SITENAME
-wait
```

pelctl create_input_channel parameters

Parameter	Description
<code>-input_channel (-ic)</code>	The name of the input channel
<code>-site (-si)</code>	Remote Site used to create the input channel
<code>-wait</code>	<i>Optional</i> Wait for SWIFT to process the channel creation request. If the channel creation fails, an error is displayed.

Example 1

Create an input channel for BIC bankfrpp:

```
pelctl create_input_channel -input_channel bankfrpp_sample -site SWNET_EP1
```

This request should be successful.

Example 2

Attempt to create the following input channel and wait for the reply:

```
pelctl create_input_channel -input_channel BankfrpP_sAmple -site SWNET_EP1 -wait
```

The creation of this channel is refused by SWIFT (input channel name must be lower case).

Error message:

```
pelctl create_input_channel: bad request
```

Opening an input channel using command line

You should open an input channel from the command line if you added it in the Site before creating it and the Site is not able to open it.

You can force the input channel to open if there were some errors at normal opening. This mode allows the channel to be opened even if it has not been properly closed or it is locked.

```
pelctl open_input_channel
  -site (-si) SITENAME
  -input_channel_window (-icw) 1...12
  -input_channel_force (-icf) { (Y) | N }
```

pelctl open_input_channel parameters

Parameter	Description
<code>-site (-si)</code>	The Site where the input channel is defined.
<code>-input_channel window(-icw)</code>	The size of the input channel window. See SWIFT documentation for more information.
<code>-input channel force (-icf)</code>	<i>Optional</i> Force the opening of the input channel. Possible values: • Y (yes) • N (no) If set to Y, the open is successful even if the input channel is already open.

Example

```
pelctl open_input_channel -site SWIFT7 -input_channel_window 20
```

Closing an input channel using command line

The command line enables you to close the input channel without stopping the Site.

```
pelctl close_input_channel
-site (-si) SITENAME
```

pelctl close_input_channel parameters

Parameter	Description
<code>-site (-si)</code>	The Site where the input channel is defined.

Example

```
pelctl close_input_channel -site SWIFT7
```

Deleting an input channel using command line

```
pelctl delete_input_channel
```

```
-input_channel (-ic) CHANNELNAME
-site (-si) SITENAME
-wait
```

pelctl delete_input_channel parameters

Parameter	Description
<code>-input_channel (-ic)</code>	The name of the input channel
<code>-site (-si)</code>	The site where the input channel is defined.
<code>-wait</code>	<i>Optional</i> Wait for SWIFT to process the request.

Specifying the input channel in the Remote Site

You specify the input channel to be used in the Remote Site. A Remote Site can use only one input channel at a time. If you want to use more than one input channel, use a separate Remote Site for each one. To guarantee the correct ordering of messages, an input channel cannot be used by several Remote Sites at the same time.

By default, the input channel is opened in force mode. This mode allows the channel to be opened even if it has not been properly closed or it is locked.

A window size is provided as a parameter of the open input channel request. However, only the parameter provided in the reply will be the one used by SWIFT.

Specifying the input channel using the Gateway GUI

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node, then select **New** to display the *New Remote Site* window.
3. Complete the fields on the [General](#) tab. In the **Protocol** field, make sure you select **SWIFTNet**.
4. Complete the fields on the [Options](#) tab.
5. On the [SWIFTNet](#) tab, select the **SnF** sub-tab and complete the **Input Channel** field.
6. Click **OK**.

Gateway creates a new Remote Site with the specified characteristics.

Specifying the input channel using online commands

Create a Remote Site using the [peladm create site](#) online command.


```

peladm create_site
  -input_channel (-ic)
  -input_channel_window (-icw) 1...12
  -input_channel_force (-icf) { (Y) | N }

```

For details of parameters, refer to the table below.

Remote Site parameters specific to SWIFTNet input channels

In this table, the corresponding field name (if available) in the Gateway GUI is shown in ***italics***.

Parameter	Description
-input_channel (-ic) <i>Input Channel</i>	The name of the input channel. Note: The input channel must already exist! First create the channel using the pelctl create_input_channel command.
-input_channel_force (-icf)	<i>Optional</i> Force the opening of the input channel. Possible values: • Y (yes) • N (no) Default = Y. You should normally need to change the default value.
-input_channel_window (-icw)	<i>Optional</i> The proposed input channel window size. You can specify a value when opening an input channel. However, only the value provided in the reply will be the one used by SWIFT. Normally you do not need to change the default value. Only specify the window size if you want to restrict the value. Possible values: 1 - 12. Default = 12

For details of all other Remote Site parameters, refer to [Site objects: Parameters List](#).

Example

```

peladm create_site -input_channel CHANNELNAME -input_channel_window 8

```

Updating the input channel in the Remote Site

This section explains how to update an input channel in the Remote Site.

Updating the input channel using the Gateway GUI

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
Gateway displays a list of all Remote Sites in the right pane.
3. In the right pane, right-click the name of the Remote Site for which you want to modify the input channel. Select **Modify...** from the context menu.
Gateway displays the *Remote Site* editing window.
4. On the **SWIFTNet** tab, select the **SnF** sub-tab and modify the **Input Channel** field.
5. Click **OK**.

Updating the input channel using online commands

Use the [peladm update_site](#) command to update the input channel specified in the Remote Site.

```
peladm update_site
-input_channel (-ic)
-input_channel_window (-icw) 1...12
-input_channel_force (-icf) { (Y) | N }
```

For details of parameters, refer to the table [above](#).

Sending messages using input channels

Gateway manages the order in which traffic is sent. Messages are automatically assigned a sequence number based on the order of the creation of the transfers.

This ordering is guaranteed for all transfers using the same priority.

An InterAct Store and Forward message will automatically use the input channel from its Remote Site if it has been specified.

A Remote Site using an input channel can still handle real time and FileAct transfers. It is however advisable to use a separate Remote Site since the ordering of the InterAct Store and Forward messages can slow down the processing of the rest of the traffic.

It is still possible to send InterAct Store and Forward traffic without using input channels through a Remote Site where such a channel has not been specified.

Input channel ordering will work with PassPort or a user exit as long as the Transfer Request contains enough information to identify an InterAct Store and Forward transfer.

Example

Enter:

```
peltrans -type MSG -snd_msg XXX -ap FTP_B -tradeoa L_TRADE -tradedea TRADE_IA_ST -da SWIFT_EP1
```

Gateway returns:

```
18
```

Enter:

```
peldsp display_trans -i 18
```

Gateway returns:

```
[...]
```

```
x_full_state='ENDED'
```

```
[...]
```

```
x_sequence_number='2'
```

```
x_ic_name='bankfrpp_sample'
```

```
x_snf_input_time='0124:2011-06-21T15:49:29'
```

Resolving gaps

A *gap* exists when the sequencing of the accepted transfers is not contiguous. A gap is created when a transfer using an input channel is rejected by SWIFT. This case is also likely to occur when Gateway is stopped while exchanges are being made using the input channel.

Transfers that have sequence numbers higher than the gap are accepted as long as the window of the input channel allows this. However, until the gap is resolved, such outgoing transfers remain in a non-final state and have an empty SnFInputTime.

It is therefore very important to avoid creating gaps by checking the validity of the transfers or to resolve the gaps as quickly as possible.

Example

Send a transfer using input channels, check SnFInputTime, sequence number and input channel for ended transfers.

When a transfer using an input channel is canceled, a gap resolution process is launched based on the sequence number attached to the transfer.

Gateway will also try to resolve gaps if they are detected as a result of opening an input channel.

Example

Send 3 transfers configured so that the 1st is accepted by SWIFT, the 2nd is rejected and 3rd is accepted. Then cancel transfer 2. A gap resolution process is launched. Check the message log. Transfer 3 should be ended with an SnF Input Time at the end of the gap resolution.

Closing input channels

An input channel is automatically closed when you stop a Site (using the [pelctl stop site](#) command) or when you stop Gateway. However, when you stop Gateway, there is no guarantee that it will have the time to close the input channel.

The closure of an input channel is only accepted if there is no gap. It is however possible to re-open an existing opened input channel which has gaps.

The command line enables you to close the input channel without stopping the Site.

More information

For more information about SWIFTNet input channels, refer to the SWIFT official documentation.

Axway Gateway: Connectors

SWIFTNet input channels

About SWIFTNet input channels

Input channels are designed to guarantee that the store-and-forward traffic sent to the SWIFT network respects a particular sequence.

SWIFT 7.0 only supports the sending of *Interact* messages through input channels.

It is not necessary to use input channels if the business applications exchanging messages already support the exchange of non-sequenced traffic.

Creating input channels

Gateway automatically creates a default input channel for the BIC. If required, create additional input channels using the `pelctl create_input_channel` command. Make sure that you respect the SWIFT naming conventions for input channels. For example, input channel names must be in lower case. Gateway does not check or correct the name or case of the channel. If the name is not SWIFT-compliant it will be rejected by SWIFT.

It is possible to restrict the creation of input channels. If user security is activated, only a user who has administrator control of the Remote Site used for the creation will be able to create the input channel.

Security considerations relating to input channels

See *Security Guide > Security best practices > [Identified threats and possible mitigations](#)* for further information (requires login).

Creating an input channel using command line

```
pelctl create_input_channel
-input_channel (-ic) CHANNELNAME
-site (-si) SITENAME
-wait
```

pelctl create_input_channel parameters

Parameter	Description
<code>-input_channel (-ic)</code>	The name of the input channel
<code>-site (-si)</code>	Remote Site used to create the input channel
<code>-wait</code>	<i>Optional</i> Wait for SWIFT to process the channel creation request. If the channel creation fails, an error is displayed.

Example 1

Create an input channel for BIC bankfrpp:

```
pelctl create_input_channel -input_channel bankfrpp_sample -site SWNET_EP1
```

This request should be successful.

Example 2

Attempt to create the following input channel and wait for the reply:

```
pelctl create_input_channel -input_channel BankfrpP_sAmple -site SWNET_EP1 -wait
```

The creation of this channel is refused by SWIFT (input channel name must be lower case).

Error message:

```
pelctl create_input_channel: bad request
```

Opening an input channel using command line

You should open an input channel from the command line if you added it in the Site before creating it and the Site is not able to open it.

You can force the input channel to open if there were some errors at normal opening. This mode allows the channel to be opened even if it has not been properly closed or it is locked.

```
pelctl open_input_channel
  -site (-si) SITENAME
  -input_channel_window (-icw) 1...12
  -input_channel_force (-icf) { (Y) | N }
```

pelctl open_input_channel parameters

Parameter	Description
-site (-si)	The Site where the input channel is defined.
-input_channel_window (-icw)	The size of the input channel window. See SWIFT documentation for more information.
-input_channel_force (-icf)	<i>Optional</i> Force the opening of the input channel. Possible values: • Y (yes) • N (no) If set to Y, the open is successful even if the input channel is already open.

Example

```
pelctl open_input_channel -site SWIFT7 -input_channel_window 20
```

Closing an input channel using command line

The command line enables you to close the input channel without stopping the Site.

```
pelctl close_input_channel
  -site (-si) SITENAME
```

pelctl close_input_channel parameters

Parameter	Description
-site (-si)	The Site where the input channel is defined.

Example

```
pelctl close_input_channel -site SWIFT7
```

Deleting an input channel using command line

```
pelctl delete_input_channel
-input_channel (-ic) CHANNELNAME
-site (-si) SITENAME
-wait
```

pelctl delete_input_channel parameters

Parameter	Description
-input_channel (-ic)	The name of the input channel
-site (-si)	The site where the input channel is defined.
-wait	<i>Optional</i> Wait for SWIFT to process the request.

Specifying the input channel in the Remote Site

You specify the input channel to be used in the Remote Site. A Remote Site can use only one input channel at a time. If you want to use more than one input channel, use a separate Remote Site for each one. To guarantee the correct ordering of messages, an input channel cannot be used by several Remote Sites at the same time.

By default, the input channel is opened in force mode. This mode allows the channel to be opened even if it has not been properly closed or it is locked.

A window size is provided as a parameter of the open input channel request. However, only the parameter provided in the reply will be the one used by SWIFT.

Specifying the input channel using the Gateway GUI

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Right-click the **Remote Site** sub-node, then select **New** to display the *New Remote Site* window.
3. Complete the fields on the [General](#) tab. In the **Protocol** field, make sure you select **SWIFTNet**.
4. Complete the fields on the [Options](#) tab.
5. On the [SWIFTNet](#) tab, select the **SnF** sub-tab and complete the **Input Channel** field.
6. Click **OK**.

Gateway creates a new Remote Site with the specified characteristics.

Specifying the input channel using online commands

Create a Remote Site using the [peladm create site](#) online command.

```
peladm create_site
-input_channel (-ic)
-input_channel_window (-icw) 1...12
-input_channel_force (-icf) { (Y) | N }
```

For details of parameters, refer to the table below.

Remote Site parameters specific to SWIFTNet input channels

In this table, the corresponding field name (if available) in the Gateway GUI is shown in *italics*.

Parameter	Description
-input_channel (-ic) <i>Input Channel</i>	The name of the input channel. Note: The input channel must already exist! First create the channel using the pelctl create_input_channel command.
-input_channel_force (-icf)	<i>Optional</i> Force the opening of the input channel. Possible values: • Y (yes) • N (no) Default = Y. You should normally need to change the default value.

Parameter	Description
<code>-input_channel_window (-icw)</code>	<i>Optional</i> The proposed input channel window size. You can specify a value when opening an input channel. However, only the value provided in the reply will be the one used by SWIFT. Normally you do not need to change the default value. Only specify the window size if you want to restrict the value. Possible values: 1 - 12. Default = 12

For details of all other Remote Site parameters, refer to [Site objects: Parameters List](#).

Example

```
peladm create_site -input_channel CHANNELNAME -input_channel_window 8
```

Updating the input channel in the Remote Site

This section explains how to update an input channel in the Remote Site.

Updating the input channel using the Gateway GUI

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Click the **Remote Site** sub-node.
Gateway displays a list of all Remote Sites in the right pane.
3. In the right pane, right-click the name of the Remote Site for which you want to modify the input channel. Select **Modify...** from the context menu.
Gateway displays the *Remote Site* editing window.
4. On the [SWIFTNet](#) tab, select the **SnF** sub-tab and modify the **Input Channel** field.
5. Click **OK**.

Updating the input channel using online commands

Use the [peladm update site](#) command to update the input channel specified in the Remote Site.

```
peladm update_site
-input_channel (-ic)
-input_channel_window (-icw) 1...12
-input_channel_force (-icf) { (Y) | N }
```

For details of parameters, refer to the table [above](#).

Sending messages using input channels

Gateway manages the order in which traffic is sent. Messages are automatically assigned a sequence number based on the order of the creation of the transfers.

This ordering is guaranteed for all transfers using the same priority.

An InterAct Store and Forward message will automatically use the input channel from its Remote Site if it has been specified.

A Remote Site using an input channel can still handle real time and FileAct transfers. It is however advisable to use a separate Remote Site since the ordering of the InterAct Store and Forward messages can slow down the processing of the rest of the traffic.

It is still possible to send InterAct Store and Forward traffic without using input channels through a Remote Site where such a channel has not been specified.

Input channel ordering will work with PassPort or a user exit as long as the Transfer Request contains enough information to identify an InterAct Store and Forward transfer.

Example

Enter:

```
peltrans -type MSG -snd_msg XXX -ap FTP_B -tradeoa L_TRADE -tradedea TRADE_IA_ST -da SWIFT_EP1
```

Gateway returns:

```
18
```

Enter:

```
peldsp display_trans -i 18
```

Gateway returns:

```
[...]  
x_full_state='ENDED'  
[...]  
x_sequence_number='2'  
x_ic_name='bankfrpp_sample'  
x_snf_input_time='0124:2011-06-21T15:49:29'
```

Resolving gaps

A *gap* exists when the sequencing of the accepted transfers is not contiguous. A gap is created when a transfer using an input channel is rejected by SWIFT. This case is also likely to occur when Gateway is stopped while exchanges are being made using the input channel.

Transfers that have sequence numbers higher than the gap are accepted as long as the window of the input channel allows this. However, until the gap is resolved, such outgoing transfers remain in a non-final state and have an empty SnFInputTime.

It is therefore very important to avoid creating gaps by checking the validity of the transfers or to resolve the gaps as quickly as possible.

Example

Send a transfer using input channels, check SnFInputTime, sequence number and input channel for ended transfers.

When a transfer using an input channel is canceled, a gap resolution process is launched based on the sequence number attached to the transfer.

Gateway will also try to resolve gaps if they are detected as a result of opening an input channel.

Example

Send 3 transfers configured so that the 1st is accepted by SWIFT, the 2nd is rejected and 3rd is accepted. Then cancel transfer 2. A gap resolution process is launched. Check the message log. Transfer 3 should be ended with an SnF Input Time at the end of the gap resolution.

Closing input channels

An input channel is automatically closed when you stop a Site (using the [pelctl stop site](#) command) or when you stop Gateway. However, when you stop Gateway, there is no guarantee that it will have the time to close the input channel.

The closure of an input channel is only accepted if there is no gap. It is however possible to re-open an existing opened input channel which has gaps.

The command line enables you to close the input channel without stopping the Site.

More information

For more information about SWIFTNet input channels, refer to the SWIFT official documentation.

Axway Gateway: Connectors

SWIFTNet queues and output channels

About SWIFTNet output channels

SWIFT release 7.0 introduces the concept of *output channels*.

Output channels are an extension of the previously-implemented Store-and-Forward (SnF) queue mechanism which allows the opening of multiple sessions for the same queue (queue sharing). The output channel is used when the SWIFT network pushes transfers to the messaging interface (Gateway). Output channels apply to both InterAct & FileAct. For more details about output channels, output sessions, queues and queue sharing, refer to the official SWIFT documentation.

Output channels, queues and output sessions

Output sessions existed already before the concept of output channels was introduced. Output sessions were in fact *queue sessions*. At a given time only one session existed for a given queue.

Starting an output session was equivalent to acquiring the queue. Stopping an output session was equivalent to releasing the queue.

The concept of *output channel* allows multiple output sessions on a queue in such a way that these output sessions are easily identified and managed. Indeed, without output channels, there is only one output session possible at a given time for a queue.

The need for output channels

The reason for using output channels is to be able to have multiple sessions on a queue.

Multiple sessions are required to be able to deliver messages from the same queue in parallel to several endpoints to achieve:

- High availability of output from a queue, where messages can still be delivered even when one of the sessions fails for whatever reason (typically connectivity problems or a problem at the receiving system)
- Higher performance, where in order to spread the traffic load messages from a queue can be handled by several messaging interfaces or by several SWIFTNet Link

When SWIFT provisions a queue, SWIFT automatically creates a corresponding output channel with the same name. Therefore, the creation of additional output channels is only required when queue sharing is used.

Naming output channels

For each queue, a default output channel with the same name is created by SWIFT.

The default output channel can be used in the case of an exclusive queue for example. When using a shared queue, several output channels can be used to start a session on the same queue. The output channels used for queue sharing should be named according to SWIFT rules (for example, when using alphabetic characters, only use lower case).

Here are some examples of how you could name output channels:

- bic8_oc1, bic8_oc2, ... for live output channels
- bic8_oc1!p, bic8_oc2!p, ... for pilot output channels

Configuring SWIFTNet output channels

In Gateway, you configure SWIFTNet output channels in the Remote Site.

Creating a Remote Site using the Gateway GUI

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node, then select **New** to display the *New Remote Site* window.
3. Complete the fields on the [General](#) tab. In the **Protocol** field, make sure you select **SWIFTNet**.
4. Complete the fields on the [Options](#) tab.
5. Complete the fields on the [SWIFTNet](#) tab, including any required fields on the **General** sub-tab.
6. Select the **SnF** sub-tab.
7. Click **New**.
8. Complete the *Output Channel Parameters* window.
9. *Optional* Click **Select subset** and select and re-order the information in the *Output Channel Subset* window.
10. Click **OK** to close each window.

Gateway creates a new Remote Site with the specified characteristics.

Creating a Remote Site using command line

Create a Remote Site using the [peladm create site](#) online command.

```
peladm create_site
  -add_channel
  -queue_name QUEUENAME
  -channel_name CHANNELNAME
  -window_size WINDOWSIZE
  -channel_subset SUBSETS
```

For details of parameters, refer to the table at the end of this section.

When creating a Site, only one output channel can be specified. Afterwards you can add additional output channels using the `peladm update_site -add_channel` command.

Note: Even though `delete_queue` and `delete_channel` are shown as subcommands of `peladm create_site` (when you enter `peladm create_site`), they are available only for `peladm update_site`.

Example

```
peladm create_site
  -alias SWIFT_REMOTE_SITE -protocol SWIFTNET
  -login_ident swift_user_name -login_password swift_
user_password
  -add_channel -queue_name bic8_q1 -channel_name bic8_
oc1
  -window_size 10 -channel_subset "InterAct_Urgent
FileAct_Urgent"
```

Remote Site parameters specific to SWIFTNet output channels

In this table, the corresponding field name in the Gateway UI is shown in *italics*.

Parameter	Description
-add_channel <i>New button on SnF sub-tab</i>	Indicates that you are adding a channel.
-channel_name (-chnname) <i>Channel name</i>	The name of the output channel.
-new_channel_name (-newcn)	The new name for the output channel (when renaming the output channel).
-new_queue_name (-newqn)	The new output channel's queue name (when renaming the queue).
-queue_name (-qname) <i>Queue name</i>	The output channel's queue name.
-window_size (-wndsz) <i>Window size</i>	<i>Optional</i> The output channel window size. In SWIFT, the window size is an optional element that is used to limit the number of concurrent messages or files sent by SWIFT on a push session. If the element is not present, then a default value is used for the window size (specified in SWIFT documentation). SWIFT adjusts the window size to the maximum value in case the application requests a higher value than the maximum allowed value. Normally you do not need to change the default value. Only specify the window size if you want to restrict the value.

Parameter	Description
-channel_subset (-chnsbs) Select subset...	<i>Optional</i> A string describing the output channel subset elements. This string can contain <i>up to six</i> subset elements. Enclose the string between quotes and use a space to separate elements. Example: <pre>-channel_subset "InterAct_Urgent FileAct_Urgent"</pre> Possible values: • InterAct • FileAct • System • Urgent • Normal • InterAct_Urgent • InterAct_Normal • FileAct_Urgent • FileAct_Normal • System_Urgent • System_Normal Notes: These values are case sensitive. The order in which the subsets appear is important. If no subset is shown, the output channel will receive all types of message of all priorities, in a first in, first out order.
-receive_retrieved_messages (-rcvretmsg) Receive retrieved messages	<i>Optional</i> Enable the attribute in order to receive retrieved messages. By setting this parameter, the output channel will be opened having the element "protocolLevel" set to value "2", as indicated in the SWIFT documentation. Example: <code>-receive_retrieved_messages "Y"</code> Note: To receive retrieved messages, the queue for which the session is started: • Must have been defined for service "swift.snf.system (!p, !x)". • Must have been used on the retrieval request payload.

Example display (SWIFTNet-related data only)

Modifying SWIFTNet output channels configured in Gateway

Note: The following operations only affect the configuration of the output channels in Gateway. This does not automatically modify the output channels or the queues defined on the SWIFT network side!

This section explains how to:

- Add a channel on a queue
- Delete a channel
- Update a channel
- Rename a queue
- Delete a queue

Modifying a Remote Site using the Gateway GUI

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Click the **Remote Site** sub-node.

Gateway displays a list of all Remote Sites in the right pane.

3. In the right pane, right-click the name of the Remote Site for which you want to modify the output channel. Select **Modify...** from the context menu.

Gateway displays the *Remote Site* editing window.

4. On the [SWIFTNet](#) tab, select the **SnF** sub-tab.
5. Select the required output channel and click **Modify**.
6. Complete the *Output Channel Parameters* window and (if required) the *Output Channel Subset* window.
7. Click **OK** to close each window.

Modifying a Remote Site using command line

Modify a Remote Site using one of the following [peladm update site](#) online commands.

For details of parameters, refer to the table [above](#).

Adding a channel

```
peladm update_site -add_channel
                  -alias SITE_NAME
                  -queue_name QUEUENAME
```



```
-channel_name CHANNELNAME  
-window_size VALUE  
-channel_subset SUBSETS
```

Deleting a channel

```
peladm update_site -delete_channel  
-alias SITE_NAME  
-queue_name QUEUENAME  
-channel_name CHANNELNAME
```

Updating a channel

```
peladm update_site -update_channel  
-alias SITE_NAME  
-channel_name CHANNELNAME  
-new_channel_name NEWCHANNELNAME  
-window_size VALUE  
-channel_subset SUBSETS
```

Current limitations:

- You cannot move a channel from one queue to another queue.
- You cannot have the same queue name in two different channels.

Renaming a queue

```
peladm update_site -rename_queue  
-alias SITE_NAME  
-queue_name QUEUENAME  
-new_queue_name NEWQUEUENAME
```

Note: This affects the configuration of Gateway only (not the configuration on the SWIFT network side).

Deleting a queue

```
peladm update_site -delete_queue  
-alias SITE_NAME  
-queue_name QUEUENAME
```

Notes:

When you delete a queue, all the channels associated with that queue will be deleted. This affects the configuration of Gateway only (not the configuration on the SWIFTNet side).

Exporting and importing Site data

You can use the `pelbase export_site` and `pelbase import` commands to export or import SWIFTNet output channel data, in addition to all other Site data.

pelbase export_site

Use [pelbase export_site](#) to export Site data, including SWIFTNet output channel information (if defined for the specified Site).

For each output channel, a line is displayed in the following format:

```
s_output_channel_info = channel_name queue_name [window_size] [subset1...
[subset6]] [receive_retrieved_messages]
```

For details of parameters, refer to the table [above](#).

Example display (SWIFTNet-related data only)

```
[...]

s_ra_instance='Ra3'

s_message_partner='rndl_mp'

s_event_end_point='rndl_eep'

s_snf_queues='bic1_rndlroqasfiafatf!x bic2_rndlroqasfiafatfdn!x'

s_snl_end_point='rndl_sep'

s_ra_pathname=''

s_ra_config_file=''

s_sign_ident_list=''

s_decrypt_ident_list=''

s_backup_site=''

s_output_channel_info='bic3_oc1!x bic4_rndlroqasfiafatf!x 10 InterAct
FileAct_Urgent N'
```

pelbase import

Use [pelbase import](#) to import Site data, including SWIFTNet output channel information, from an importable object file

The format in which the output channel information must appear in the import file is:

channel_name queue_name [window_size] [subset1...[subset6]] [receive_retrieved_messages]

Notes:

- channel_name and queue_name are mandatory
- window_size and subset are optional. If specified, they must appear in this order.

Controlling output channels

You can use the following commands to control output channels:

- pelctl open_output_channel
- pelctl close_output_channel
- pelctl create_output_channel
- pelctl delete_output_channel

pelctl open_output_channel

```
pelctl open_output_channel
  -outputchannel_name (-ocn) CHANNELNAME
  -queue_name (-qn) QUEUENAME
  [-site (-si) SITENAME]
```

This command opens the output channel with the name CHANNELNAME, for the queue QUEUENAME. You can use the optional `site` parameter to specify the name of a Remote Site associated with the output channel.

pelctl close_output_channel

```
pelctl close_output_channel
  -outputchannel_name (-ocn) CHANNELNAME
  [-site (-si) SITENAME]
```

This command closes the output channel with the name CHANNELNAME. You can use the optional `site` parameter to specify the name of a Remote Site associated with the output channel.

pelctl create_output_channel

```
pelctl create_output_channel
  [-site (-si) SITENAME]
  -outputchannel_name (-ocn) CHANNELNAME
```

This command creates the output channel on the specified site with the specified name.

Example

```
pelctl create_output_channel -site SWIFT7 -outputchannel_name ptsafrrk_outchannel1!x
```

pelctl delete_output_channel

```
pelctl delete_output_channel
    [-site (-si) SITENAME]
    -outputchannel_name (-ocn) CHANNELNAME
```

This command deletes the output channel on the specified site with the specified name.

Example

```
pelctl delete_output_channel -site SWIFT7 -outputchannel_name ptsafrrk_outchannel1!x
```

Displaying information about output channels

peldsp display_site

Use [peldsp display_site](#) to display Site data, including SWIFTNet output channel information (if defined for the specified Site).

For each output channel, a line is displayed in the following format:

```
s_output_channel_info = channel_name queue_name [window_size] [subset1...
[subset6]] [receive_retrieved_messages]
```

For details of parameters, refer to the table [above](#).

Example display (SWIFTNet-related data only)

```
[...]

s_ra_instance='Ra3'

s_message_partner='rndl_mp'

s_event_end_point='rndl_eep'

s_snf_queues='bic1_rndlroqasfiafatf!x bic2_rndlroqasfiafatfdn!x'

s_snl_end_point='rndl_sep'

s_ra_pathname=''

s_ra_config_file=''

s_sign_ident_list=''
```

```
s_decrypt_ident_list=''

s_backup_site=''

s_output_channel_info='bic3_oc1!x bic4_rnd1roqasfiafatf!x 10 InterAct
FileAct_Urgent N'
```

peldsp status_swoutputsession

Use `peldsp status_swoutputsession` to display the status of an output session.

```
peldsp status_swoutputsession
  [-site (-si) SITENAME]
  [-queue_name (-qn) QUEUENAME]
  [-outputchannel_name (-ocn) CHANNELNAME]
```

peldsp status_swoutputsession parameters

The following parameters (all optional) enable you to filter the response.

Parameter	Description
<code>-site (-si)</code>	Specify a Site name to get the status for this Site only.
<code>-queue_name (-qn)</code>	Specify a queue name to get the status for this queue only.
<code>-outputchannel_name (-ocn)</code>	Specify an output channel name to get the status for this output channel only.

Command output

For each output session, the `peldsp status_swoutputsession` command displays several parameters:

- Queue name
- Output channel defined for that queue (if any)
- The Site on which the queue is defined
- `sw_sn1_status` – the actual status of the output session returned by the SAG (response to status request from SWIFT). Possible values:
 - Active
 - Acquiring
 - TimedOut
 - Closed

- `sw_gtw_status` – the internal status of the output session. Possible values:
 - Shut down
 - Starting
 - Running
 - Asked to be closed
 - Broken
 - Stopped
 - Status checking

Example output

```
sw_queue_name='bic1_rndnrottsfiafa!x'
sw_outputchannel_name='bic2_rndnrottsfiafa!x'
sw_session_id='bic3_rndnrottsfiafa!x:p:000142'
sw_site='SAGLIVE_RNDN'
sw_snl_status='Active'
sw_gtw_status='Running'

sw_queue_name='bic4_rndnrottsfiafadn!x'
sw_outputchannel_name='bic5_rndnrottsfiafadn!x'
sw_session_id='bic6_rndnrottsfiafadn!x:p:000098'
sw_site='SAGLIVE_RNDN'
sw_snl_status='Active'
sw_gtw_status='Running'
```

Monitoring output sessions

Currently there are two ways of monitoring output sessions:

- **Subscribing to receive unsolicited events**

When Gateway starts up, during initialization, it subscribes to an EventEP (defined in the Remote Site **Event end point**/ `event_end_point` parameter) which enables output session events to be received. Currently there is only one event, generated each time SNL detects that an output session (queue session or output channel session) was closed by SWIFT. Every time this event is received, it is logged by Gateway.

- **Requesting status for each output session**

Gateway requests the status for each output session automatically. You can specify the time interval at which Gateway initiates the requests in the [SnF check timer](#) parameter in the configuration menu.

Controlling SnF session behavior

You can modify the following parameters in the configuration menu according to your SnF session requirements:

- [SnF check timer](#): Specify the time interval (in seconds) during which Gateway checks the status of the acquired SnF session or opened SnF sessions in case no activity has been noticed on that queue or output channel in that time interval.
- [Reacquire SnF session automatically](#): Select whether or not an SnF session should be reacquired/reopened automatically if it is lost.

Controlling queues

For queues, you can use the following commands to control the queue operations manually:

Command	Function
pelctl acquire swqueue	Acquire a queue
pelctl release swqueue	Release a queue

pelctl acquire_swqueue

Syntax	<code>pelctl acquire_swqueue {-queue_name (-n)} [-site (-si)] [-force (-f)]</code>
Description	Use this command to trigger the acquisition of a registered queue.
Mandatory parameters	<code>-queue_name (-n)</code> : Enter the name of the queue.

Optional parameters

`-site (-si)` : Enter the name of the Site through which the queue is to be acquired (the same queue can be defined on more than one Site, in a nominal-backup relationship).

`-force (-f)` : Use this parameter to force the acquisition of the queue.

You can issue an acquire request but the queue may already be acquired. In that case Gateway checks the status of the current session. If the session is reported as Active then nothing further is done. If not, the queue is reacquired. When using the `-force` parameter, the SnF session is renewed.

Example

```
pelctl acquire_swqueue -n ptsafrkk_crfa!x -f
```

OK. Queue scheduled for operation.

```
SNET031I 19.09.2008 14:32:46 SWIFTLIVE_SNF_
NT(CRNTSNFMP) acquired queue ptsafrkk_
crfa!x session ptsafrkk_crfa!x:p:631215.
```

```
SNET039I 19.09.2008 14:31:44 SWIFTLIVE_SNF_
NT(CRNTSNFMP) SnF session ptsafrkk_
crfa!x:p:630940 returned status Active.
```

```
SNET103W 19.09.2008 14:31:43 Initiating
queue status request for ptsafrkk_
crfa!x:p:630940.
```

*pelctl release_swqueue***Syntax**

```
pelctl release_swqueue {-queue_name (-n)}
[-site (-si)]
```

Description

Use this command to release the current SnF session of a specified queue.

Mandatory parameters

`-queue_name (-n)` : Enter the name of the queue.

Optional parameters

`-site (-si)` : Enter the name of the Site. This is useful if a queue is registered on multiple Sites (in a nominal-backup relationship).

Usage	Once an SnF session has been released manually, the queue is no longer monitored or reacquired, unless: • The Site on which the queue is registered is restarted • Gateway is restarted • The queue is manually reacquired using the <code>pelctl acquire_swqueue</code> command
Example	<pre>pelctl release_swqueue -n ptsafrkk_crfa!x OK. Queue scheduled for operation SNET032I 19.09.2008 14:38:03 SWIFTLIVE_SNF_ NT(CRNTSNFMP) released session ptsafrkk_ crfa!x:p:631215.</pre>

Displaying information about queues

Use `peldsp display_site` to display Site data, including SWIFTNet queue information (if defined for the specified Site).

For the queue list, a line is displayed in the following format:

```
s_snf_queues='bic1_rndlroqasfiafatf!x bic2_rndlroqasfiafatfdn!x'
```

Use [peldsp status_swoutputsession](#) to display the status of a queue session.

More information

For more information about SWIFTNet output channels, refer to the SWIFT official documentation.

Axway Gateway: Connectors

SWIFTNet filtering duplicates for queues and output channels

The following sections describe SWIFTNet filtering duplicates for queues and output channels.

Dynamically filtering duplicates for queues and output channels

Pay attention when creating, updating or importing Remote Sites with queues or output channels. Doing so avoids possible inconsistencies when there are duplicate queues and output channel definitions in several Remote Sites.

Gateway verifies all sites at startup and takes action when it encounters duplicate definitions of queues or output channels.

Note: Output sessions are the dynamic representation of queues and output channels from acquiring queues or opening output channels.

Filtering is dynamic. A global list of output sessions is created at Gateway startup and updated as sites are started or stopped. Gateway checks each new potential session for conflicts with existing sessions:

- At startup Gateway checks all active SWIFTNet Remote Sites in the order from the database. If a site has a queue or output channel that conflicts with the global list, the site is disabled.
- At manual start of a Remote Site, if a conflict with the global list is detected for a Remote Site queue or output channel, the site is disabled.

Because the disabled state is an internal state, a user can know a site became unusable only from logs.

Every 30 seconds a check is made of disabled sites to determine whether conflicts persist with the global list. If no conflicts are identified, the site is restarted.

A user cannot use a disabled site until the conflict is resolved.

Static configuration behaviors

The following are details of the conflicting situations described in the preceding section.

About the notations in the following examples:

- For example, Q1, Q2 for legacy queues (SWIFT 6 version) definitions
- For example, Q1:C1, Q2:C2 for output channels definitions (channel-queue pairs)

1. Same queue present in different or same sites

Example 1

Site1: Q1, Q1, Q2

Action: (Site1 stopped) Gateway start or manual start of Site1.

Result: Site1 is not allowed to start and is disabled.

Example 2

Site1 (first in database): Q1, Q2

Site2: Q2, Q3

Action: (both sites stopped) Gateway start or manual start of Site1 followed immediately by a manual start of Site2.

Result: Site2 is not allowed to start and is disabled.

Example 3

Site1 (first in database): Q1, Q2

Site2: Q2, Q3

Action: (both sites stopped) Manual start of Site2 followed by manual start of Site1 after the log message:

```
Site2 site has been started
```

Result: Site1 is not allowed to start and is disabled.

2. Same output channel present in different or same sites

Example 1

Site1: Q1:C1, Q1:C1

Action: (Site1 stopped) Gateway start or manual start of Site1.

Result: Site1 is not allowed to start and is disabled.

Example 2

Site1 (first in database): C1:Q1, C2:Q2

Site2: C2:Q2, C3: Q3

Action: (both sites stopped) Gateway start or manual start of Site1 followed immediately by a manual start of Site2.

Result: Site2 is not allowed to start and is disabled.

Example 3

Site1 (first in database): C1:Q1, C2:Q2

Site2: C2:Q2, C3: Q3

Action: (both sites stopped) Manual start of Site2 followed by manual start of Site1 after the log message:

```
Site2 site has been started
```

Result: Site1 is not allowed to start and becomes disabled.

3. Mixed queue or output channel present in different or same sites

Example 1

Site1: Q1, Q2, C3:Q1

Action: (Site1 stopped) Gateway start or manual start of Site1.

Result: Site1 is not allowed to start and is disabled.

Example 2

Site1: C1:Q1, C1:Q2

Action: (Site1 stopped) Gateway start or manual start of Site1.

Result: Site1 is not allowed to start and is disabled.

Example 3

Site1: C1:Q1 (first in database)

Site2: C1:Q2

Action: (both sites stopped) Gateway start or manual start of Site1 followed immediately by a manual start of Site2.

Result: Site1 is not allowed to start and is disabled.

Example 4

Site1 (first in database): C1:Q1, C2:Q2

Site2: C2:Q2, C3: Q3

Action: (both sites stopped) Manual start of Site2 followed by manual start of Site1 after the log message:

```
Site2 site has been started
```

Result: Site1 is not allowed to start and is disabled.

Example 5

Site1 (first in database): Q1, Q2

Site2: C2:Q2, C3: Q3

Action: (both sites stopped) Gateway start or manual start of Site1 followed immediately by a manual start of Site2.

Result: Site1 is not allowed to start and is disabled.

Example 6

Site1 (first in database): Q1, Q2

Site2: C2:Q2, C3: Q3

Action: (both sites stopped) Manually start Site2 and manually start Site1 after the log message:

```
Site2 site has been started
```

Result: Site1 is not allowed to start and is disabled.

Observation

Next site configuration (Site1: C1:Q1, C2:Q1) is permitted, but the success of opening the output channels depends on the queue-sharing mode. A shared queue can only be opened with one or more distinct output channels, but cannot be acquired in legacy mode. An exclusive queue can be acquired in legacy mode or opened with one output channel.

Using duplicate queues and output channels with backup sites

If a nominal site contains definitions of queues or output channels, define the same queues or channels in the backup site. Normally, duplicate definitions of queues or output channels between standard sites should not exist. Gateway treats such definitions specially (see [Dynamically filtering duplicates for queues and output channels on page 2103](#)). Backup sites, however, should define the same queues and output channels as their nominal site.

Queues

Queue definitions are shared between the nominal and backup site. This means all queues are acquired through the nominal site and, only when the nominal site is stopped, the same queues are acquired through the backup site.

Output channels

Output channels (channel-queue pairs) are shared between nominal and backup sites. This means all output channels are opened through the nominal site and, only when the nominal site is stopped, the same output channels are opened through the backup site.

Only common declarations between nominal and backup sites are allowed. Mixed queues with output channel definitions are forbidden.

Queues tables

About the notations in the tables:

- For example, Q1, Q2 for legacy queues (SWIFT 6 version) definitions
- For example, Q1:C1, Q2:C2 for output channels definitions (channel-queue pairs)
- For example, S(Q1), S(Q1:C1) for output sessions (dynamic representation of queues and output channels resulting from acquiring queues or opening output channels)

Table 1

In the following table are examples of common definitions of queues and output channels for a nominal and backup site. The Gateway actions and results are included.

No.	Static configuration example	Site status	Action and result
1	Nominal1 – Q1, Q2, C3:Q3, C4:Q4 Backup1 – Q1, Q5, C4:Q4, C6:Q6	Both sites stopped	Action: Gateway start or manual start of Nominal1, followed immediately by a manual start of Backup1. Result: Nominal site has all its sessions active. Nominal1 – S(Q1), S(Q2), S(C3:Q3), S(C4:Q4) Backup1 – S(Q5), S(C6:Q6)
2	Nominal1 – Q1, Q2, C3:Q3, C4:Q4 Backup1 – Q1, Q5, C4:Q4, C6:Q6	Both sites stopped	Action: Manual start of Backup1 followed by a manual start of Nominal1 after the log message: Backup1 site has been started Result: Initially, Backup1 opens all its sessions, but Nominal1 takes back and owns the duplicates. Nominal1 – S(Q1), S(Q2), S(C3:Q3), S(C4:Q4) Backup1 – S(Q5), S(C6:Q6)
3	Nominal1 – Q1, Q2, C3:Q3, C4:Q4 Backup1 – Q1, Q5, C4:Q4, C6:Q6	Both sites started Nominal1 – S(Q1), S(Q2), S(C3:Q3), S(C4:Q4) Backup1 – S(Q5), S(C6:Q6)	Action: Stop Nominal1. Result: Common active sessions pass to Backup1. Nominal1 -- Backup1 – S(Q1), S(C4:Q4) S(Q5), S(C6:Q6)

No.	Static configuration example	Site status	Action and result
4	Nominal1 – Q1, Q2, C3:Q3, C4:Q4 Backup1 – Q1, Q5, C4:Q4, C6:Q3 Q3 should be shared to have two channels opened with success for the same queue.	Both sites stopped	Action: Gateway start or manual start of Nominal1 followed immediately by a manual start of Backup1. Result: Nominal site has all its active sessions. Nominal1 – S(Q1), S(Q2), S(C3:Q3), S(C4:Q4) Backup1 – S(Q5), S(C6:Q3)

Table 2

In the following table are examples of duplicated definitions of queues and output channels, generating conflicts between nominal and backup sites. The Gateway actions and results are included.

Resolve the conflicts in the logs to enable a site to start and become operational.

No.	Static configuration example	Site status	Action and result
1	Nominal1 – Q1, Q2 , C3:Q3, C4:Q4 (first in database) Backup1 – Q1, Q5, C4:Q4, C2:Q2	Both sites stopped	Action: Gateway start or manual start of Nominal1 followed immediately by a manual start of Backup1. Result: Backup1 is not allowed to start. It becomes disabled because a conflict was identified (bold items).
2	Nominal1 – Q1, Q2 , C3:Q3, C4:Q4 Backup1 – Q1, Q5, C4:Q4, C2:Q2	Both sites stopped	Action: Manual start of Backup1 followed by a manual start of Nominal1 after the log message: <code>Backup1 site has been started</code> Result: Nominal1 is not allowed to start. It becomes disabled because a conflict was identified (bold items).

No.	Static configuration example	Site status	Action and result
3	Site1 – Q1, Q3 , C5:Q4 (first in database) Nominal1 – Q1, Q2, C3:Q3 , C4:Q4 (second in database) Backup1 – Q1 , Q5, C4:Q4, C6:Q6 Q3 should be shared to have two channels opened with success for the same queue	All sites stopped	Action: Gateway start or manual start of Site1, then Nominal1, then Backup1 (starting order is database order) . Result: Nominal1 and Backup1 are not allowed to start. They are disabled because a conflict was identified (bold items).
4	Site1 – Q1 , Q3, C5:Q4 (first in database) Nominal1 – Q1, Q2, C3:Q3, C4:Q4 Backup1 – Q1 , Q5, C4:Q4, C6:Q6 Q3 should be shared to have two channels opened with success for the same queue	All sites stopped	Action: Manual start of Site1, followed by a manual start of Backup1 after the log message: <code>Site1 site has been started</code> Result: Backup1 is not allowed to start and is disabled because a conflict was identified (bold items).

Do not allow nominal and backup sites to have the same message partner, SNL and event endpoint on same SAG. When receiving multiple SnF transfers on a common queue or output channel, Swift balances the transfers among all server processes (nominal and backup). This creates inconsistencies for duplicates and routing.

Axway Gateway: Connectors

SWIFTNet queue sharing

[Overview](#)

[Changing the queue sharing mode](#)

Overview

Output channels allow the opening of multiple sessions for the same queue (queue sharing).

This topic explains how to change the queue sharing mode from *exclusive* to *shared* or vice-versa.

Changing the queue sharing mode

You can change the queue sharing mode by sending a system message to SWIFT. To do this, create a SWIFTNet InterAct Store-and-Forward Transfer Request with an embedded message requesting the queue to change sharing mode. In Gateway you send the system message as a transfer requesting the queue to change the mode.

Example message

The information provided here is an example of the message to send to SWIFT. Adapt the contents according to your own requirements.

```
<SwInt:RequestPayload>
  <Ah:AppHdr xmlns:Ah="urn:swift:xsd:$ahV10">
    <Ah:MsgRef>REFERENCE</Ah:MsgRef>
    <Ah:CrDate>DATE</Ah:CrDate>
  </Ah:AppHdr>
  <Doc:Document xmlns:Doc="urn:swift:xsd:xsys.028.001.01">
    <Doc:xsys.028.001.01>
      <SwInt:Queue>QUEUE_NAME_TO_CHANGE_SHARING_
MODE</SwInt:Queue>
      <Doc:ShrMd>SHARING_MODE</Doc:ShrMd>
    </Doc:xsys.028.001.01>
  </Doc:Document>
</SwInt:RequestPayload>
```

In this message, you need to specify the following parameters:

- **<Ah:MsgRef>** – the message reference
- **<Ah:CrDate>** – the date, in the format yy-mm-ddThh:mm:ss (example: 11-06-14T11:40:30)
- **<SwInt:Queue>** – the name of the queue that you want to change
- **<Doc:ShrMd>** – the sharing mode:
 - **Exclusive** - not using output channels
 - **Shared** - using output channels

Create a Transfer Request, making sure that you use the *embedded* mode for the InterAct message.

Example peltrans command

```
peltrans
  -type MSG
  -direction O
  -mode I
  -file_component CHANGE_QUEUE_SHARING_MODE_SAMPLE_
MODIFIED_FILE
  -org_alias LOCAL
  -dest_alias SWIFT
```

```
-trade_org_alias MY_SWIFT_LOCAL
-trade_dest_alias SWIFT_SYSTEM_SHAREQUEUE
-appli FTP_B
-sw_payload_type embedded
```

Trading Partner

The destination Trading Partner SWIFT_SYSTEM_SHAREQUEUE to which the transfer is sent must contain:

- DN = cn=system,o=swift,o=swift
- Service name = swift.snf.system!x (for pilot network) or swift.snf.system (for live network)

Error message on an acquired queue

You cannot change the queue sharing mode on an acquired queue. If the queue has already been acquired, you will get a message similar to:

```
-- 2011/06/14 11:30:06 - ERROR from SwApiSwCallEx
2011/06/14 11:30:06 Reponse:
<?xml version="1.0" encoding="UTF-
8"?><SwInt:ExchangeResponse><SwInt:SwiftRequestRef>SNL0202
6-2011-06-
14T09:30:18.5748.000135Z</SwInt:SwiftRequestRef><SwGbl:Sta
tus>
<SwGbl:StatusAttributes>
<SwGbl:Severity>Transient</SwGbl:Severity>
<SwGbl:Code>Sw.SnF.OutputSessionActive</SwGbl:Code>
<SwGbl:Parameter>Level=1</SwGbl:Parameter>
<SwGbl:Text>A session is active on the queue. The share
mode cannot be changed while a session is open on the
queue</SwGbl:Text>
</SwGbl:StatusAttributes>
</SwGbl:Status></SwInt:ExchangeResponse>
```

In this case, stop the Remote Site that makes use of the queue before modifying it.

Axway Gateway: Connectors

SWIFTNet OverdueTime

[Overview](#)

[Sending OverdueTime](#)

[Receiving an Overdue file or message](#)

[Verifying the OverdueTime in the transfer properties](#)

Overview

SWIFTNet OverdueTime allows the sender of a SWIFT transfer to specify the latest date/time when the receiver should receive the file or message. After this time, a message or file is considered as being overdue.

The receiver will receive the file or message even after the overdue time. The receiver will be warned about this, but no further action is performed. The sender will also receive a message saying that the transfer is overdue.

Time zone

The overdue time must be entered as local time (the seconds are truncated to '00') and will appear as such in Gateway's interface. However, it is converted to UTC when sending to SWIFT.

Sending OverdueTime

When you send a file or message you can specify the OverdueTime in the Transfer Request or in the Transfer Model.

Specifying OverdueTime in the Transfer Request

Using the peltrans command

peltrans [-sw_overdue_time (-swot)]

Specify the overdue time using the `-sw_overdue_time (-swot)` parameter.

Enter the date and time in the format `YYYYMMDD HHMMSS`. Note however that the seconds are truncated to '00' when sending.

Using the Gateway GUI

On the Transfer Request **SWIFTNet** tab, **SnF** sub-tab, check and complete the **Date** and **Time** fields as follows:

Field	Description
SnF sub-tab [for Store and forward Delivery mode]	
Overdue time	
Date	To set a date value for the OverdueTime, check this option and select the required date using the drop-down calendar. When the option is unchecked, no OverdueTime will be sent.

Field	Description
Time	To set a time value for the OverdueTime, check this option and set the required hours and minutes (the seconds are truncated to '00' when sending). When the option is unchecked, but Date is checked, the OverdueTime sent is the date with time '00:00:00'. When both Time and Date are unchecked, no OverdueTime will be sent. Note: If you use the Time option, you must also set the Date option.

Specifying OverdueTime in the Transfer Model

You can also specify the SWIFTNet overdue time in the Transfer Model.

Receiving an Overdue file or message

Even if the overdue date has passed, the transfer is accepted. A log entry will appear, indicating the transfer was overdue.

Example:

```
SNET080W 10.05.2011 15:43:31 SAGLIVEGTW(rnd9_mp) [93] Transfer is
overdue; due date: 2011-05-10 15:40:00.
```

Verifying the OverdueTime in the transfer properties

Sender

For the sender, the presence of an OverdueTime means the transfer should be received before the indicated date. The overdue time is set by the sender himself.

When the due time arrives, SWIFT automatically creates a system message for the sender and stores it in the notification queue to inform him about the partner not receiving the file before the specified time. The system message is present in the Mailbox and linked to the original transfer by the `receipt_reply_to` field (in the system message).

To receive overdue warning system messages, you need a Trading Partner that allows the reception of these system message types. For information about the default Trading Partner used to receive system messages, refer to [SYSTEM_MESSAGES remote Trading Partner](#). For information about request type values associated with different system message types, refer to the SWIFTNet documentation.

Receiver

For the receiver, the presence of an OverdueTime means the transfer was overdue when receiving.

Displaying transfer details

Use the `peldsp display_trans` command to display the details of a transfer.

In the details, Gateway displays the `OverdueTime` (if present).

Alternatively you can view the transfer details, including the `OverdueTime` (if present), in the **SWIFTNet** / **SnF** tab of the View transfer dialog.

SWIFTNet Store-and-Forward Cold Start

Overview

SWIFTNet Store-and-Forward Cold Start is an exceptional situation in which SWIFT has to restore the Store-and-Forward service from an empty state. When a SWIFTNet Cold Start occurs, all traffic and data in the Store-and-Forward system is lost, queues are reset to Exclusive mode and non-generic input channels and non-default output channels are deleted.

Gateway offers mechanisms allowing you to detect the occurrence of a Cold Start, and offers guidelines on how to resume normal operation.

Note Currently, SWIFTNet Store-and-Forward messaging is exclusively housed in the European SWIFTNet zone. If this zone is subjected to a Cold Start, Store-and-Forward traffic will be lost for all SWIFTNet zones.

After a Cold Start

After the SWIFTNet Store-and-Forward system resumes its operation following a Cold Start, its state is as follows:

- The transfers awaiting delivery in the intended recipients queues in the Store-and-Forward system are lost (including positively acknowledged messages and files). It is the responsibility of each sending partner to take action in re-sending the transfers with an uncertain delivery status, marked as Possible Duplicates. SWIFT makes no further delivery attempts for these messages and files.
- Historical data related to message receipts or message delivery before the Cold Start is lost.
- The SWIFTNet queues are reset to an Exclusive share mode. All non-default output channels and all non-generic input channels are removed.
- All Input Sequence Numbers on the generic input channels, and all Output Sequence Numbers on the default output channels are reset to 0.
- The database identifier specified in the `Sw:SnFInputTime` changes, as if the BIC was reassigned to a new database.
- The SWIFTNet infrastructure resumes its operation in the default post-Cold Start configuration, i.e. the users can connect, send and receive files and messages using the generic output channels and default input channels.

In effect, you must perform the following actions to resume operation:

- Update queue sharing modes to reflect the pre-Cold Start state. For more information, see [SWIFTNet queue sharing](#).
- Recreate the non-default output channels and the non-generic input channels, according to the pre-Cold Start configuration. The [SWIFTNet queues and output channels](#) and [Controlling output channels](#) sections describe how to create output channels using Gateway. For creating input channels, see [SWIFTNet input channels](#).
- Resend transfers with an uncertain delivery status, marked as Possible Duplicates.

For more information, see [Resuming Normal Operations](#).

Detecting a SWIFTNet Store-and-Forward Cold Start

A SWIFTNet Store-and-Forward Cold Start follows a period in which Gateway was unable to communicate with the SWIFT system. However, it is important to differentiate between a Cold Start and normal connectivity issues (including those generating a system recovery).

In the case of prolonged SWIFTNet unavailability, you can go to the SWIFT page (www.swift.com) and check for information, or contact the SWIFT support. It is thus possible to obtain the information that a Cold Start just occurred over the Store-and-Forward infrastructure, in this way.

Another method of detecting a Store-and-Forward Cold Start is the receipt of an Unsolicited Undelivered Traffic Report system message (xsys.005.002.01). SWIFT will queue an Unsolicited Undelivered Traffic Report system message after a Cold Start for the generic queue of each BIC8. For more information, see [Unsolicited Undelivered Traffic Reports](#).

Gateway receives and logs these system messages, as long as the generic queue is successfully acquired or its associated output channel is successfully opened. For this, a SWIFT Remote Site must be configured with the generic queue, and must be started after the SWIFTNet system's operation recovers after encountering connectivity issues. Failure to do this could result in the inability to detect a Cold Start situation, and perform the operations required to correctly resume normal operation.

Example of Gateway logging an incoming Unsolicited Undelivered Traffic Report

```
SNET044I 15.10.2013 19:37:01 SAGLIVEGTW(rnd9_mp) Unsolicited Undelivered Traffic Report system message identified at check-in (looking for Trading Partner).
```

```
SNET143W 15.10.2013 19:37:01 SAGLIVEGTW(rnd9_mp) [117] received Unsolicited Undelivered Traffic Report (xsys.005.002.01) generated after Cold Start. ReportOption: CS; Page number: 1; is last page: Yes; execution status: Success.
```

In cases where a Store-and-Forward Cold Start was detected, appropriate actions should be taken before resuming normal operations, as described in the following sections.

If the SWIFTNet unavailability was not related to a Cold Start, none of the described actions should be taken.

If the SWIFTNet connectivity is lost for extended periods, it is recommended to stop the SWIFTNet Remote Sites until SWIFT is functional again. This is to avoid repeated attempts to reestablish the connection. The SWIFTNet system availability should be determined either by periodically attempting to restart a SWIFT Remote Site, by using the `sag_test_connect` tool provided by RAHA installation, or by checking with SWIFT.

This would be done regardless of what caused the connectivity loss, as before reestablishing connectivity it is not possible to determine if a Cold Start occurred (except if communicated by SWIFT).

Unsolicited Undelivered Traffic Reports

Immediately after a Store-and-Forward Cold Start, SWIFTNet queues the Unsolicited Undelivered Traffic Reports system message (xsys.005.002.01) to the generic queues of each BIC8. In order to receive this system message, after regaining SWIFTNet connectivity the generic queue corresponding to your BIC must be acquired, or its associated output channel must be opened.

To receive an Unsolicited Undelivered Traffic Report system message after a connectivity loss:

1. Ensure the SWIFTNet Store-and-Forward system is available. This can be done by starting a SWIFTNet Remote Site which has a queue or a default open channel configured. If the Remote Site starts successfully and it acquires the queue or opens the default output channel, SWIFTNet Store-and-Forward system can be considered functional.
2. Acquire the BIC8's generic queue, or open its default output channel. Note that after a Cold Start, the queue sharing mode is reset to Exclusive. If it remains in the Shared mode (if previously set), this would be an indication that a Cold Start has not occurred.

Opening the default output channel of the generic queue with the System subset would ensure that only system messages are delivered. This way Unsolicited Undelivered Traffic Reports and other Cold Started-related system messages can be received and fully processed before receiving business-related messages.

It is possible to receive the Unsolicited Undelivered System Messages on a non-default output channel. For this, you would have to change the generic queue sharing mode to Shared, and (re)create non-default output channels for the generic queue before attempting to open the output channel.

After a successful acquire of the BIC8's generic queue or open of its associated output channel, in the event of a Cold Start, an Unsolicited Undelivered Traffic Report (xsys.005.002.01), system message should be received.

Recommendations regarding the shared generic queue configuration

There are situations in which more than one Messaging Interface (Gateway) would use the same BIC8, and might share the associated generic queue. However, after a Cold Start the generic queue is in Exclusive mode, and only its default output channel exists.

It is recommended to designate a single Messaging Interface for receiving Unsolicited Undelivered Traffic Reports. The designated Messaging Interface could be configured to open the default output channel of the generic queue with the System subset, which would work in either Shared or Exclusive modes.

Configuring more than one Messaging Interface (Gateways) to receive Unsolicited Undelivered Traffic Reports is possible, but not recommended. In any case, you must know precisely which Messaging Interfaces can receive Cold Start-related system messages and monitor them accordingly.

Processing Unsolicited Undelivered Traffic Report (xsys.005.002.01) system messages.

Gateway does not automatically process Undelivered Traffic Report system messages. However, it detects and logs receipt allowing manual actions to be taken to ensure the correct resumption of the normal operations following a Cold Start.

The Undelivered Traffic Report system messages contain an Undelivered Message List, with a sequence of Undelivered Message information. For a complete description of the structure of an Undelivered Traffic Report and an Undelivered Message List, please consult the SWIFTNet documentation.

The information logged by Gateway includes:

- The transfer identifier associated with the Unsolicited Undelivered Traffic Report system message
- The type of system message
- The page number
- Whether it's the last page
- The report option

The transfer identifier associated with the Unsolicited Undelivered Traffic Report system message can be used to retrieve the payload which contains the system message itself (and the Undelivered Message List)

If there are too many Undelivered Messages, the Unsolicited Undelivered Traffic Report will be split into multiple pages, and a system message will be received for each page. The page number and the last page indicator can be used to recompose the complete Undelivered Message List.

For an Unsolicited Undelivered Traffic Report generated by a Store-and-Forward Cold Start operation, the report option field will be 'CS'.

Note The payload file(s) must to be shared with all the message interfaces belonging to the company (share the same BIC8), because the restore actions involve all of them, not only the message interface acquiring the generic queue or opening his (default) output channel. For more information, see [Undelivered Traffic Report payload](#) and [Processing the Undelivered Traffic Report payload](#).

Solicited and Unsolicited Undelivered Traffic Report system messages.

SWIFTNet offers the possibility of interrogating the Store-and-Forward system about the Undelivered Traffic - files and messages sent but not yet delivered. This is done by sending an Undelivered Traffic Report Request (xsys.004.001.01) system message to the SWIFTNet Store-and-Forward system.

There are however differences between the Solicited (xsys.005.001.01) and the Unsolicited (xsys.005.002.01) Undelivered Traffic Reports regarding which information they retrieve. Additionally, there is no system message to request an Unsolicited Undelivered Traffic Report.

The Unsolicited Undelivered Traffic Report contains information about undelivered traffic (associated with a BIC8) which was in the Store-and-Forward system before a Cold Start event occurred, ignoring the pending traffic initiated after the Cold Start. This system message is automatically queued to each BIC8 generic queue after the Store-and-Forward system resumes operation.

In contrast with the Unsolicited Undelivered Traffic Report, the *Solicited* Undelivered Traffic Report can be requested by sending a request (xsys.004.001.01) to SWIFTNet. It also contains information about the current pending traffic. If requested after a Cold Start, this report contains information about the pending Store-and-Forward traffic initiated after the Cold Start. No information about the traffic lost due to the Cold Start is provided.

Undelivered Traffic Report payload

An Undelivered Traffic Report contains a sequence of Undelivered Message information (this can be missing if there is no undelivered traffic to report).

You should monitor Gateway's log, detect the receipt of the Unsolicited Undelivered Traffic Report system message(s), reformat the full Undelivered Message list (if the report is sent in more than one page) and forward the information to an appropriate authority, for processing.

Note that the situation may arise where pieces of reports are received by different message interfaces (if the generic queue has assigned output channels opened by multiple message interfaces). In this case the reports should be centralized for processing.

Processing the Undelivered Traffic Report payload

Processing the Undelivered Traffic Report payload involves determining which traffic should be resent.

If re-queuing a message or a file to be sent is decided, the corresponding transfer will have to be marked as a Possible Duplicate. For more information, see [Possible Duplicate management for SWIFTNet](#).

Resuming Normal Operations

In order to completely resume normal operations following a SWIFTNet Store-and-Forward Cold Start, the following procedure must be followed by all message interfaces:

- Detect the availability of Store-and-Forward: confirm that the Store-and-Forward system is operational, by:
 - a. verifying that the Gateway SWIFTNet processes are running and connected to SWIFT
 - b. checking that at least one queue was acquired, or at least one output channel was successfully opened
- Re-create the non-generic input channels and the non-default output channels (output channels with a name other than that of a corresponding queue) according to the pre-Cold Start configuration.
- Set the Shared mode for queues which were shared before the Cold Start. Changing a queue sharing mode is described in [SWIFTNet queue sharing](#).
- Resume normal operations.

SWIFTNet Header Info

[Overview](#)

[Specifying parameters](#)

[End of transfer parameters](#)

[GikAPI parameters](#)

[XML file structure](#)

[Examples](#)

Overview

You can use the enhanced FileAct header structure (HeaderInfo field) to send additional business-related header information such as payment instructions through FileAct. This is useful if you use service profiles like Transaction Count and Payment File Summary. Gateway makes a validation to ensure the content is consistent with the data and meta-data being transported.

When acting as a Responder, Gateway saves the HeaderInfo structure, if present, in a file and signals its presence so that upper applications can carry out additional processing. For FileAct RT GET transfers, Gateway also offers the possibility to put such information in the E2EControl.

Specifying parameters

Transfer Request parameters

The following parameters are references to files with the same internal structure. The file must respect a specific structure, as described in "XML File structure" later in this topic.

If you have both HeaderInfo and SignatureList in the same file, and you want to use them both, use the **Additional params** parameter to specify the file. If you only want to use the HeaderInfo information from the file, use the **Header info** parameter.

Header info

Reference to a file on the disk containing HeaderInfo.

```
peltrans -header_info (-hinf) <file reference to header info structure>
```

Additional params

Reference to a file on the disk containing HeaderInfo, SignatureList or Recipient List.

```
peltrans -sw_add_params (-swaddpar) <file reference to additional
SWIFTNet structures>
```

End of transfer parameters

The following end of transfer parameters are available:

`x_sw_add_params` = <Reference to a file on disk where Header Info and/or Signature List and/or Additional parameters are stored>

`x_header_info` = Y / N

Examples

```
x_sw_add_params='d:\builds\gtw6120dev\srvininstall\run_
time\tmp\SW000001.xml'
```

```
x_header_info='N'
```

GikAPI parameters

The following parameters can be found in the `gikapic.h` header:

- `GIK_T_SW_HEADER_INFO` 383 /* String */
- `GIK_T_SW_ADD_PARAMS` 385 /* String */

XML file structure

Refer to the XML schemas (`.xsd` files) supplied with Gateway for examples of file structure. You can use the `AddSwParams.xsd` file to create or validate XML files for HeaderInfo, SignatureList and Recipient List usage. The `.xsd` files are located in:

```
<Gateway installation directory>/run_time/swnet
```

Typical XML file with HeaderInfo

```
<AddSwParams>
  <SwHeaderInfo>
    <Out> ...
  </Out>
  <In> ...
  </In>
  </SwHeaderInfo>
</AddSwParams>
```

Adding more HeaderInfo fields

It is possible to extend HeaderInfo support with additional HeaderInfo fields.

Gateway, by default, supports the following HeaderInfo fields:

- Payment File Summary (`ApplSpcfc.PmtSummry.01`)
- Transaction Count (`ApplSpcfc.TxsCntr.01`)
- Trade Files (`ApplSpcfc.Trade.01`)
- Bank Account Management Files (`ApplSpcfc.BAM.01`)

The HeaderInfo data (specified when creating a transfer) is validated against an XML schema consisting on a master schema (`AddSwParams.xsd`) and several element schemas.

Adding a new HeaderInfo element schema requires the following operations:

- add the corresponding `.xsd` file (e.g. `ApplSpcfc.NewSchema.01.xsd`) in the `run_time/swnet` directory
- alter the `AddSwParams.xsd` file to include the new schema:
 - add a `namespace` entry, for example:

```
<xs:import namespace="urn:swift:xsd:ApplSpcfc.NewSchema.01"
schemaLocation="ApplSpcfc.NewSchema.01.xsd"/>
```
 - add the corresponding `xs:element` entry into the HeaderInfo `xs:complexType`, for example:

```
<xs:element xmlns="urn:swift:xsd:ApplSpcfc.NewSchema.01" ref="ApplSpcfc"
minOccurs="0" maxOccurs="1"/>
```

Note Make sure the added element schemas are well formed, and that the changes in `AddSwParams` are correct - an incorrect schema will impact the product functionality.
Prior validation with an external XML tool is recommended.

Note The same mechanism is used to validate other `AddSwParams` elements: Signature Lists, System Messages, ProductInfo Lists, ThirdParty and Recipient DN Lists and so on. Please **do not remove any element of the XML schemas**.

Note Consult the *SWIFTNet Standards* before extending HeaderInfo support to new fields. This should be done only for SWIFTNet Standards-compliant fields not included by a default Gateway installation.

Examples

For examples of files containing SWIFTNet Header Info, refer to [SWIFTNet dump to XML](#).

SWIFTNet Header Info

[Overview](#)

[Specifying parameters](#)

[End of transfer parameters](#)

[GikAPI parameters](#)

[XML file structure](#)

[Examples](#)

Overview

You can use the enhanced FileAct header structure (HeaderInfo field) to send additional business-related header information such as payment instructions through FileAct. This is useful if you use service profiles like Transaction Count and Payment File Summary. Gateway makes a validation to ensure the content is consistent with the data and meta-data being transported.

When acting as a Responder, Gateway saves the HeaderInfo structure, if present, in a file and signals its presence so that upper applications can carry out additional processing. For FileAct RT GET transfers, Gateway also offers the possibility to put such information in the E2EControl.

Specifying parameters

Transfer Request parameters

The following parameters are references to files with the same internal structure. The file must respect a specific structure, as described in "XML File structure" later in this topic.

If you have both HeaderInfo and SignatureList in the same file, and you want to use them both, use the **Additional params** parameter to specify the file. If you only want to use the HeaderInfo information from the file, use the **Header info** parameter.

Header info

Reference to a file on the disk containing HeaderInfo.

```
peltrans -header_info (-hinf) <file reference to header info structure>
```

Additional params

Reference to a file on the disk containing HeaderInfo, SignatureList or Recipient List.

```
peltrans -sw_add_params (-swaddpar) <file reference to additional  
SWIFTNet structures>
```

End of transfer parameters

The following end of transfer parameters are available:

`x_sw_add_params` = <Reference to a file on disk where Header Info and/or Signature List and/or Additional parameters are stored>

`x_header_info` = Y / N

Examples

```
x_sw_add_params='d:\builds\gtw6120dev\srvininstall\run_time\tmp\SW000001.xml'
```

```
x_header_info='N'
```

GikAPI parameters

The following parameters can be found in the `gikapic.h` header:

- `GIK_T_SW_HEADER_INFO` 383 `/* String */`
- `GIK_T_SW_ADD_PARAMS` 385 `/* String */`

XML file structure

Refer to the XML schemas (`.xsd` files) supplied with Gateway for examples of file structure. You can use the `AddSwParams.xsd` file to create or validate XML files for HeaderInfo, SignatureList and Recipient List usage. The `.xsd` files are located in:

```
<Gateway installation directory>/run_time/swnet
```

Typical XML file with HeaderInfo

```
<AddSwParams>
  <SwHeaderInfo>
    <Out> ...
  </Out>
  <In> ...
  </In>
  </SwHeaderInfo>
</AddSwParams>
```

Adding more HeaderInfo fields

It is possible to extend HeaderInfo support with additional HeaderInfo fields.

Gateway, by default, supports the following HeaderInfo fields:

- Payment File Summary (`ApplSpcfc.PmtSummry.01`)
- Transaction Count (`ApplSpcfc.TxsCntr.01`)
- Trade Files (`ApplSpcfc.Trade.01`)
- Bank Account Management Files (`ApplSpcfc.BAM.01`)

The HeaderInfo data (specified when creating a transfer) is validated against an XML schema consisting on a master schema (`AddSwParams.xsd`) and several element schemas.

Adding a new HeaderInfo element schema requires the following operations:

- add the corresponding .xsd file (e.g. `ApplSpcfc.NewSchema.01.xsd`) in the `run_time/swnet` directory
- alter the `AddSwParams.xsd` file to include the new schema:
 - add a `namespace` entry, for example:

```
<xs:import namespace="urn:swift:xsd:ApplSpcfc.NewSchema.01"
schemaLocation="ApplSpcfc.NewSchema.01.xsd"/>
```
 - add the corresponding `xs:element` entry into the HeaderInfo `xs:complexType`, for example:

```
<xs:element xmlns="urn:swift:xsd:ApplSpcfc.NewSchema.01" ref="ApplSpcfc"
minOccurs="0" maxOccurs="1"/>
```

Note Make sure the added element schemas are well formed, and that the changes in `AddSwParams` are correct - an incorrect schema will impact the product functionality.
Prior validation with an external XML tool is recommended.

Note The same mechanism is used to validate other `AddSwParams` elements: Signature Lists, System Messages, ProductInfo Lists, ThirdParty and Recipient DN Lists and so on. Please **do not remove any element of the XML schemas**.

Note Consult the *SWIFTNet Standards* before extending HeaderInfo support to new fields. This should be done only for SWIFTNet Standards-compliant fields not included by a default Gateway installation.

Examples

For examples of files containing SWIFTNet Header Info, refer to [SWIFTNet dump to XML](#).

Axway Gateway: Connectors

SWIFTNet Distribution List

[Overview](#)

[Specifying parameters](#)

[End of transfer parameters](#)

[GikAPI parameters](#)

[XML file structure](#)

[Examples](#)

Overview

This topic describes SWIFTNet Distribution List (distribution to multiple recipients).

SWIFT supports the possibility to send the same message or file to multiple recipients in one single transaction.

When Gateway acts as Initiator

Gateway enables you to send InterAct SnF messages, or FileAct SnF transfers, to several recipients in a single transaction.

Gateway allows other applications to specify the list of recipients with the help of an input XML file. The XML document must be well-formed and have a specific format. Gateway makes sure that what has been requested is consistent with the content of the exchange request. If it is not, it adds the necessary information or rejects the exchange request and generates an error message.

The same file or message cannot be sent several times to the same recipient. The recipient list should only contain unique entries.

Additionally, you can specify whether or not the responder can receive the list of recipients by using the `sw_public_recipient_list_ind` parameter in either the Transfer Request or the Trading Partner.

Like non-distributed SnF traffic, Gateway receives positive or failed delivery notifications. These delivery notifications are however always received as system messages. According to the receipt option, Gateway can receive as many system message delivery notifications or failed delivery notifications as the number of specified recipients.

To be able to send a distributed file or message, you need to create a remote Trading Partner with the trade name (**Name**) "cn=distributor,o=swift,o=swift". The service (**Service name**) and request type (**Request type**) have the same significance as for normal remote Trading Partners.

To receive non-delivery warning, delivery notification or failed delivery notification system messages, you need a Trading Partner that allows the reception of these system message types. For information about the default Trading Partner used to receive system messages, refer to [SYSTEM_MESSAGES remote Trading Partner](#). For information about request type values associated with different system message types, refer to the SWIFTNet documentation.

When Gateway acts as Responder

There is no significant difference between the reception of distributed and non-distributed files or messages.

If the sender specifies that the recipient list is public, Gateway saves this list in a file and signals its presence so that upper applications can then carry out additional processing.

Specifying parameters

Transfer Request parameters

Additional params

Reference to a file on the disk containing Recipient List, HeaderInfo or SignatureList information.

Use the **Additional params** parameter to specify the file. The file must respect a specific structure, as described in "XML File structure" later in this topic.

```
peltrans -sw_add_params (-swaddpar) <file reference to additional  
SWIFTNet structures>
```

End of transfer parameters

The following end of transfer parameters are available:

`x_sw_add_params` = <Reference to a file on disk where Signature List and/or Header Info and/or Additional parameters are stored>

Example

```
x_sw_add_params='d:\builds\gtw6120dev\srvininstall\run_  
time\tmp\SW000001.xml'
```

GikAPI parameters

The following parameter can be found in the `gikapic.h` header:

- `GIK_T_SW_ADD_PARAMS` 385 /* String */

XML file structure

Refer to the XML schemas (`.xsd` files) supplied with Gateway for examples of file structure. You can use the `AddSwParams.xsd` file to create or validate XML files for Recipient List, HeaderInfo and SignatureList information usage. The `.xsd` files are located in:

<Gateway installation directory>/run_time/swnet

Typical XML file with Recipient List

```
<AddSwParams>  
  <SwRecipientList xmlns:Sw="urn:swift:snl:ns.Sw">  
    <Out>
```

```
<Sw:RecipientList>
  <Sw:RecipientDN>
    cn=recipient1,ou=fr,o=bankfrpp,o=swift
  </Sw:RecipientDN>
  <Sw:RecipientDN>
    cn=recipient2,ou=fr,o=bankfrpp,o=swift
  </Sw:RecipientDN>
  <Sw:RecipientDN>
    cn=recipient3,ou=fr,o=bankfrpp,o=swift
  </Sw:RecipientDN>
</Sw:RecipientList>
</Out>
</SwRecipientList>
</AddSwParams>
```

Examples

For examples of files containing SWIFTNet Recipient List information, refer to [SWIFTNet dump to XML](#).

Axway Gateway: Connectors

SWIFTNet File and Message Copy

[About SWIFTNet File and Message Copy](#)

[File and Message Copy support in Gateway](#)

[Using Copy services when sending files or messages](#)

[Selecting copy destinations for T-Copy and Y-Copy](#)

[Receiving files and messages containing Copy service information](#)

[GikAPI support for File and Message Copy](#)

[Considerations](#)

About SWIFTNet File and Message Copy

SWIFTNet File and Message Copy enables the sending of a file or a message to one or more third parties in addition to sending the full file or message to the intended recipient.

- For InterAct: the entire copied message is sent to the specified third parties.
- For FileAct: depending on the service configuration, either the full file or just the request header is sent to the specified third parties.

File and Message Copy is only used for services operating in Store-and-Forward mode.

Copy modes

There are two File and Message Copy modes:

- Y-Copy: The third party is the first recipient of the sent/copied file or message. The third party authorizes (or not) SWIFT to send the file or message to the final recipient. If the third party refuses to send the file or message to the recipient, a system message refusal notification is sent by SWIFT SnF to the sender party. If the third party accepts to send the file or message, the sender only receives a system message if the authorization notification option has been set, either in the request or at the service level.
- T-Copy: The request is sent to the destination and to one or more third parties. The actual destination receives the full information and the third parties receive the entire message (InterAct) or, depending on the service configuration, the full file or just the header (FileAct).

In both modes, if the delivery of the copy to the third party fails, a non-delivery notification is sent to the sender ("Failed" or "Rejected" status).

The File and Message Copy services rely on the FileAct [Header Info](#) structure and on the use of the [Signature List](#) structure to transport the signature information.

File and Message Copy support in Gateway

The Gateway SWIFTNet connector can act as Sender or Receiver in a File or Message Copy exchange:

- Send Transfer Requests marked with CopyIndicator, AuthNotifIndication (Sender)
- Receive and acknowledge a non-delivery notification from the Third Party (Sender)
- Receive and acknowledge an Authorization notification (Sender)
- Receive and acknowledge a Refusal notification (Sender)
- Receive a file or message and retrieve the Copy information (Receiver)

Using Copy services when sending files or messages

The parameters described in this section enable you to specify how to work with Copy services.

Transfer Request parameters

When creating a Transfer Request, use the following `peltrans` parameters:

Parameter	Meaning	Value
<code>-sw_copy_indicator (-ci)</code> Request copy	SWIFTNet File or Message Copy service indicator. Select this option to request a copy of the file or message to be sent to a third party defined at the service level.	One of: • Y = yes • N = no (default)
<code>-sw_auth_indicator (-ai)</code> Request authorisation notification	SWIFTNet Authorization notification indicator. Select this option to receive Authorization system messages for files or messages sent over Y-copy services.	One of: • Y = yes • N = no (default)

Trading Partner parameters

When creating a Trading Partner, use the following `peladm create_tradeprt` parameters:

Parameter	Meaning	Value
<code>-sw_copy_indicator (-ci)</code>	SWIFTNet File or Message Copy service indicator. Select this option to request a copy of the file or message to be sent to a third party defined at the service level.	One of: • Y = yes • N = no
<code>-sw_auth_indicator (-ai)</code>	SWIFTNet Authorization notification indicator. Select this option to receive Authorization system messages for files or messages sent over Y-copy services.	One of: • Y = yes • N = no

Model parameters

When creating a Model, use the following `peladm create_model` parameters:

Parameter	Meaning	Value
<code>-sw_copy_indicator (-ci)</code>	SWIFTNet File or Message Copy service indicator. Select this option to request a copy of the file or message to be sent to a third party defined at the service level.	One of: • Y = yes • N = no (default)
<code>-sw_auth_indicator (-ai)</code>	SWIFTNet Authorization notification indicator. Select this option to receive Authorization system messages for files or messages sent over Y-copy services.	One of: • Y = yes • N = no (default)

Decision Rule parameters

The following Decision Rule condition parameters are available:

`sw_copy_indicator`

`sw_auth_indicator`

End-of-transfer parameters

The following end-of-transfer parameters are available:

`x_sw_copy_indicator`

`x_sw_auth_indicator`

`x_sw_auth_by_xfer` (link the original sender transfer with the authorization transfer)

`x_sw_auth_to_xfer` (link SnF InterAct authorization transfer with the original message)

Receiving Y-Copy Authorization/Refusal notifications

Authorization notifications are SWIFTNet InterAct SnF system messages. You need a Trading Partner that allows the reception of these system message types. For information about the default Trading Partner used to receive system messages, refer to [SYSTEM_MESSAGES remote Trading Partner](#). For information about request type values associated with different system message types, refer to the SWIFTNet documentation.

Example

```
peladm create_tradepart -f SWIFTNET -a SWIFT_SYSTEM_
AUTH_OK
    -name cn=system,o=swift,o=swift -loc N
    -sc swift.snf.system!x -rqt xsys.002.001.01 -ro ACK_
BY_MONITOR
    -signl /home/users/user/SignListAckResponse.txt

peladm create_tradepart -f SWIFTNET -a SWIFT_SYSTEM_
AUTH_K0
    -name cn=system,o=swift,o=swift -loc N
    -sc swift.snf.system!x -rqt xsys.003.001.01 -ro ACK_
BY_MONITOR
    -signl /home/users/user/SignListAckResponse.txt
```

or:

```
peladm create_tradepart -f SWIFTNET -a SWIFT_SYSTEM_
AUTH_OK
    -name cn=system,o=swift,o=swift -loc N
    -sc swift.snf.system!x -rqt AnyRequestType -ro ACK_BY_
MONITOR
    -signl /home/users/user/SignListAckResponse.txt
```

Authorization system messages have the `sw_auth_indicator` parameter set.

The Authorization/Refusal notification business payload is available as the file component of the SnF InterAct system message.

Selecting copy destinations for T-Copy and Y-Copy

T-Copy

For T-Copy, a transfer can be sent to one or several copy destinations. If the service definition (ASP) allows the sender to specify the copy destinations then the message or file is sent to those partners. This can be achieved by making a normal SWIFT SnF Copy transfer but specifying the third parties in an XML file on the sender side. The minimum number of third parties defined in the XML file is one. The maximum number is defined in the SWIFT documentation.

Y-Copy

For Y-Copy, a transfer can be sent to only one copy destination (authorization / central bank). Not all Y-Copy services allow the sender to specify the copy destination (central bank), they are simply hard-coded in the service definition at SWIFT. Y-Copy services that allow the user to select the copy destination from a list of available copy destinations actually require the sender to specify it. Failing to

do so would result in the Transfer Request being rejected by SWIFT. The Y-Copy third party destination is specified in an XML file in the same way as for T-Copy, however there can be only one copy destination.

Completing the Transfer Request

In the Transfer Request, specify the full path of the local physical file containing Third Party List information. In the Gateway GUI, use the **Additional params** parameter on the **SWIFTNet** tab.

Alternatively, use the peltrans command:

```
peltrans -sw_add_params (-swaddpar) <file reference to additional SWIFTNet structures>
```

Typical XML file containing Third Party List: sender side

The XML tag: <Sw:ThirdPartyDN> contains the DN to send to. You can specify up to 10 distinguished DNs.

```
<SwThirdPartyDNList>
  <Out>
    <Sw:CopyThirdPartyList>
      <Sw:ThirdPartyDN>cn=tcop-
cid1,o=swcibeap,o=swift</Sw:ThirdPartyDN>
      <Sw:ThirdPartyDN>cn=tcop-
cid2,o=swcibeap,o=swift</Sw:ThirdPartyDN>
      <Sw:ThirdPartyDN>cn=test1,o=swhqnlml,o=swift</Sw:Thi
rdPartyDN>
    </Sw:CopyThirdPartyList>
  </Out>
</SwThirdPartyDNList>
```

Typical XML file containing Third Party List: receiver side

The receiver of a copied transfer should expect a Copied Third Party List in the addSwParams. Note the use of <In> tags instead of <Out> tags in the XML file. For example:

```
<SwThirdPartyDNList>
  <In>
    <Sw:CopiedThirdPartyList>
      <Sw:ThirdPartyDN>cn=tcop-
cid1,o=swcibeap,o=swift</Sw:ThirdPartyDN>
      <Sw:ThirdPartyDN>cn=tcop-
cid2,o=swcibeap,o=swift</Sw:ThirdPartyDN>
      <Sw:ThirdPartyDN>cn=test1,o=swhqnlml,o=swift</Sw:Thi
rdPartyDN>
    </Sw:CopiedThirdPartyList>
  </In>
</SwThirdPartyDNList>
```

Receiving files and messages containing Copy service information

If the file from SWIFT SnF, received in push-mode, contains a Sw:Copy element in the FileDescriptor, the transfer is marked with the `copy_indicator`. The CopyType and CopyState elements are recovered and saved on the transfer, available as end-of-transfer parameters and Decision Rule condition parameters.

Decision Rule parameters

The following Decision Rule condition parameters are available:

```
sw_copy_indicator
```

```
sw_copy_type
```

```
sw_copy_state
```

End-of-transfer parameters

The following end-of-transfer parameters are available:

```
x_sw_copy_indicator
```

```
x_sw_copy_type
```

```
x_sw_copy_state
```

```
x_sw_third_party_dn
```

ThirdPartyToReceiver information

The ThirdPartyToReceiver information, if present, is deposited in the SWIFT parameter file. The parameter `sw_add_parameters` contains the filename and directory path. For an example of a file containing ThirdPartyToReceiver information, refer to [SWIFTNet dump to XML](#).

GikAPI support for File and Message Copy

The following parameters can be found in the `gikapic.h` header:

- `GikSetUChar(xferParams, GIK_T_SW_COPY_IND, ITP_SWIFTNET_YES)`
- `GikSetUChar(xferParams, GIK_T_SW_AUTH_IND, ITP_SWIFTNET_NO)`

When creating a transfer, use the values `ITP_SWIFTNET_YES` and `ITP_SWIFTNET_NO` (uses `XferRequest_t`).

For `XferUpdate` and `XferGet`, use boolean values {0, 1} for the parameters (uses `XferRecord_t`).

Example in C

```
#define GIK_T_SW_COPY_IND          386    /* Uchar */
```



```

#define GIK_T_SW_AUTH_IND          387    /* Uchar */

#define GIK_T_SW_THIRD_DN          390    /* String */
#define GIK_T_SW_COPY_STATE        391    /* Uchar -
SwCopyTypeEnm */
#define GIK_T_SW_COPY_TYPE        392    /* Uchar -
SwCopyStateEnm */

/* GikXferPut and GikXferUpdate */

GikSetUChar(xferParams, GIK_T_SW_COPY_IND, ITP_SWIFTNET_
YES)
GikSetUChar(xferParams, GIK_T_SW_AUTH_IND, ITP_SWIFTNET_
NO)

/* GikXferGet */

/* Get values: 1 = TRUE; 0 = FALSE */

GikGetUChar(xferParams, GIK_T_SW_COPY_IND, &copy_
indicator);
GikGetUChar(xferParams, GIK_T_SW_COPY_IND, &copy_
indicator);

if (GikGetUChar(xferParams, GIK_T_SW_COPY_STATE, &copy_
state))
    printf("    Error GikGetUChar :
%s\n\n",GikErrMsg());
    else
    {
        char s_copy_state[30] = {0};

        switch (copy_state)
        {
            case ITP_SW_COPY_STATE_ACTIVE:
                strcpy(s_copy_state, "Active");
                break;
            case ITP_SW_COPY_STATE_BYPASS:
                strcpy(s_copy_state, "Bypass");
                break;
            case ITP_SW_COPY_STATE_FALLBACK:
                strcpy(s_copy_state,
"TCopyFallback");
                break;
            case ITP_SW_COPY_NONE:
                strcpy(s_copy_state, "none");
                break;
            default:
                strcpy(s_copy_state, "Unknown
Value!");

```

```
        }

        printf("    copy_state = %s\n", s_
copy_state);
    }

    /* Copy Type - checked */
    if (GikGetUChar(xferParams, GIK_T_SW_COPY_TYPE, &copy_
type))
        printf("    Error GikGetUChar :
%s\n\n",GikErrMsg());
    else
    {
        char s_copy_type[30] = {0};

        switch (copy_type)
        {
            case ITP_SW_COPY_TYPE_FULL:
                strcpy(s_copy_type, "Full");
                break;
            case ITP_SW_COPY_TYPE_HEADER:
                strcpy(s_copy_type, "Header");
                break;
            case ITP_SW_COPY_NONE:
                strcpy(s_copy_type, "none");
                break;
            default:
                strcpy(s_copy_type, "Unknown
Value!");
        }

        printf("    copy_type = %s\n", s_copy_
type);
    }

    if ( GikGetString(xferParams, GIK_T_SW_THIRD_DN, third_
party_dn, sizeof(third_party_dn)) )
        printf("    Error GikGetString :
%s\n\n",GikErrMsg());
    else printf("    third_party_dn = %s\n", third_party_
dn);
```

Considerations

When non-repudiation is requested on an SnF Copy service, the sender Transfer Request must be signed using a Signature List.

When using File or Message Copy parameters, the SWIFT service configuration must be compatible with the Copy options selected:

- File or Message Copy must only be used over SnF service
- The Copy indicator must be allowed on the service
- The Auth notif indication must be allowed on the service

If the SWIFT configuration is not correct, the request will generate an error at SNL level. This error will appear in the Gateway log.

Axway Gateway: Connectors

SWIFTNet Signature List

[Overview](#)

[Specifying parameters](#)

[End of transfer parameters](#)

[GikAPI parameters](#)

[XML file structure](#)

[Examples](#)

Overview

This topic describes the SWIFTNet Signature List feature (new paradigm for signing).

Signing is possible using two paradigms: Crypto or SignatureList. If signing is requested and there is reference to a SignatureList file, Gateway uses the SignatureList paradigm. Otherwise Gateway signs the message using Crypto.

When Gateway acts as Initiator

Gateway enables you to sign InterAct RT and SnF messages, and FileAct RT and SnF transfers, exchanged over the SWIFT network using SignatureList for signing. The SignatureList paradigm secures a message digest of the information that needs to be secured instead of signing the information itself (as done when using Crypto).

Gateway allows other applications to specify what needs to be signed with the help of an input XML file. The XML document must be well-formed and have a specific format. Gateway makes sure that what has been requested is consistent with the content of the exchange request. If it is not, it adds the necessary information or rejects the exchange request and generates an error message.

When Gateway acts as Responder

As a responder, Gateway saves the secured data information received from the initiator in a file and signals its presence so that backend applications can then carry out validation checks. In the first step, SNL automatically verifies the incoming secured data. Backend applications need to complete this

process by verifying that the DigestValue of an application-specific reference matches the information that is digested.

Gateway also offers the possibility to secure the responses using the same SignatureList signing paradigm.

Specifying parameters

If you request signing and you specify a SignatureList file, Gateway uses the SignatureList paradigm. Otherwise Gateway signs the message using Crypto. You must use the same signing option (Crypto or Signature List) at the sender side *and* the receiver side.

Requesting signature

On the sender side: Request signing in the Transfer Request (using the **Sign** or `trade_sign` parameter) or the Trading Partner (using the **Sign** parameter).

On the receiver side: Request signing in the Trading Partner (using the **Sign** parameter). The signing parameters are used in generating the response for the received request.

The service profile (defined by SWIFT) determines whether signing on sender requests / receiver responses is optional or mandatory.

Specifying the Signature List file

Gateway offers two ways of specifying the Signature List file:

- Per transfer (in the Transfer Request)
- For all transfers exchanged with a specific partner (in the Trading Partner). When Gateway accepts requests through this Trading Partner then the Signature List file is used to sign the responses sent back to SWIFT.

When sending or receiving an ACK, the presence of a Signature List file at the Trading Partner level ensures that the ACK Request/Response is signed with the default SWIFT digest references.

You cannot use the same Signature List file for signing both the request and the response (as they have different default SWIFT digest references) from the Trading Partner level. If you use the same Trading Partner both for sending and receiving, you must specify a Signature List file when submitting outgoing requests. The Signature List file from the Trading Partner is used to sign the responses.

Transfer Request parameters

The following parameters are references to files with the same internal structure. The file must respect a specific structure, as described in "XML File structure" later in this topic.

If you have both HeaderInfo and SignatureList in the same file, and you want to use them both, use the **Additional params** parameter to specify the file. If you only want to use SignatureList information from the file, use the **Signature list** parameter.

Signature list

Reference to a file on the disk containing a Signature List.

```
peltrans -sign_list (-signl) <file reference to signature list>
```

Additional params

Reference to a file on the disk containing SignatureList, HeaderInfo or Recipient List.

```
peltrans -sw_add_params (-swaddpar) <file reference to additional  
SWIFTNet structures>
```

*Trading Partner parameter***Signature List**

Reference to a file on the disk containing a Signature List.

```
peladm create_tradepart ? | grep sign_list  
  
-sign_list (-signl) <file reference to signature list>
```

End of transfer parameters

The following end of transfer parameters are available:

`x_sw_add_params` = <Reference to a file on disk where Signature List and/or Header Info and/or Additional parameters are stored>

`x_sign_list` = Y / N

Examples

```
x_sw_add_params='d:\builds\gtw6120dev\srvininstall\run_  
time\tmp\SW000001.xml'
```

```
x_sign_list='Y'
```

GikAPI parameters

The following parameters can be found in the `gikapic.h` header:

- `GIK_T_SW_SIGN_LIST` 384 /* String */
- `GIK_T_SW_ADD_PARAMS` 385 /* String */

XML file structure

Refer to the XML schemas (.xsd files) supplied with Gateway for examples of file structure. You can use the `AddSwParams.xsd` file to create or validate XML files for SignatureList, HeaderInfo and Recipient List usage. The .xsd files are located in:

```
<Gateway installation directory>/run_time/swnet
```

Typical XML file with SignatureList

```
<AddSwParams>
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"
    xmlns:SwSec="urn:swift:snl:ns.SwSec">
    <Out> ...
  </Out>
  <In> ...
  </In>
  </SwSignatureList>
</AddSwParams>
```

Support for different SwSec:SignDNs in the XML file

There may be cases where you want to use a single Signature List file to sign the messages exchanged with a specific Trading Partner (defined at the Trading Partner level). However, the messages can be exchanged through two different Remote Sites with two different principals (SecurityContext UserDNs).

In this case, you can use the joker string "AnyDN" for SwSec:SignDN. The DN will then be the principal from the active (used) Remote Site.

For more information, refer to [Working with signed or encrypted messages](#) in the *Configuring the Axway Gateway/SWIFTNet environment* topic.

Examples

For examples of files containing SWIFTNet Signature List information, refer to [SWIFTNet dump to XML](#).

Axway Gateway: Connectors

SWIFTNet Signature List

[Overview](#)

[Specifying parameters](#)

[End of transfer parameters](#)

[GikAPI parameters](#)

[XML file structure](#)

[Examples](#)

Overview

This topic describes the SWIFTNet Signature List feature (new paradigm for signing).

Signing is possible using two paradigms: Crypto or SignatureList. If signing is requested and there is reference to a SignatureList file, Gateway uses the SignatureList paradigm. Otherwise Gateway signs the message using Crypto.

When Gateway acts as Initiator

Gateway enables you to sign InterAct RT and SnF messages, and FileAct RT and SnF transfers, exchanged over the SWIFT network using SignatureList for signing. The SignatureList paradigm secures a message digest of the information that needs to be secured instead of signing the information itself (as done when using Crypto).

Gateway allows other applications to specify what needs to be signed with the help of an input XML file. The XML document must be well-formed and have a specific format. Gateway makes sure that what has been requested is consistent with the content of the exchange request. If it is not, it adds the necessary information or rejects the exchange request and generates an error message.

When Gateway acts as Responder

As a responder, Gateway saves the secured data information received from the initiator in a file and signals its presence so that backend applications can then carry out validation checks. In the first step, SNL automatically verifies the incoming secured data. Backend applications need to complete this process by verifying that the DigestValue of an application-specific reference matches the information that is digested.

Gateway also offers the possibility to secure the responses using the same SignatureList signing paradigm.

Specifying parameters

If you request signing and you specify a SignatureList file, Gateway uses the SignatureList paradigm. Otherwise Gateway signs the message using Crypto. You must use the same signing option (Crypto or Signature List) at the sender side *and* the receiver side.

Requesting signature

On the sender side: Request signing in the Transfer Request (using the **Sign** or `trade_sign` parameter) or the Trading Partner (using the **Sign** parameter).

On the receiver side: Request signing in the Trading Partner (using the **Sign** parameter). The signing parameters are used in generating the response for the received request.

The service profile (defined by SWIFT) determines whether signing on sender requests / receiver responses is optional or mandatory.

Specifying the Signature List file

Gateway offers two ways of specifying the Signature List file:

- Per transfer (in the Transfer Request)
- For all transfers exchanged with a specific partner (in the Trading Partner). When Gateway accepts requests through this Trading Partner then the Signature List file is used to sign the responses sent back to SWIFT.

When sending or receiving an ACK, the presence of a Signature List file at the Trading Partner level ensures that the ACK Request/Response is signed with the default SWIFT digest references.

You cannot use the same Signature List file for signing both the request and the response (as they have different default SWIFT digest references) from the Trading Partner level. If you use the same Trading Partner both for sending and receiving, you must specify a Signature List file when submitting outgoing requests. The Signature List file from the Trading Partner is used to sign the responses.

Transfer Request parameters

The following parameters are references to files with the same internal structure. The file must respect a specific structure, as described in "XML File structure" later in this topic.

If you have both HeaderInfo and SignatureList in the same file, and you want to use them both, use the **Additional params** parameter to specify the file. If you only want to use SignatureList information from the file, use the **Signature list** parameter.

Signature list

Reference to a file on the disk containing a Signature List.

```
peltrans -sign_list (-signl) <file reference to signature list>
```

Additional params

Reference to a file on the disk containing SignatureList, HeaderInfo or Recipient List.

```
peltrans -sw_add_params (-swaddpar) <file reference to additional  
SWIFTNet structures>
```


Trading Partner parameter

Signature List

Reference to a file on the disk containing a Signature List.

```
peladm create_tradepart ? | grep sign_list  
-sign_list (-signl) <file reference to signature list>
```

End of transfer parameters

The following end of transfer parameters are available:

`x_sw_add_params` = <Reference to a file on disk where Signature List and/or Header Info and/or Additional parameters are stored>

`x_sign_list` = Y / N

Examples

```
x_sw_add_params='d:\builds\gtw6120dev\srvininstall\run_  
time\tmp\SW000001.xml'  
  
x_sign_list='Y'
```

GikAPI parameters

The following parameters can be found in the `gikapic.h` header:

- `GIK_T_SW_SIGN_LIST` 384 /* String */
- `GIK_T_SW_ADD_PARAMS` 385 /* String */

XML file structure

Refer to the XML schemas (`.xsd` files) supplied with Gateway for examples of file structure. You can use the `AddSwParams.xsd` file to create or validate XML files for SignatureList, HeaderInfo and Recipient List usage. The `.xsd` files are located in:

```
<Gateway installation directory>/run_time/swnet
```

Typical XML file with SignatureList

```
<AddSwParams>  
  <SwSignatureList xmlns:Sw="urn:swift:snl:ns.Sw"  
    xmlns:SwSec="urn:swift:snl:ns.SwSec">  
    <Out> ...
```

```
</Out>
<In> ...
</In>
</SwSignatureList>
</AddSwParams>
```

Support for different SwSec:SignDNs in the XML file

There may be cases where you want to use a single Signature List file to sign the messages exchanged with a specific Trading Partner (defined at the Trading Partner level). However, the messages can be exchanged through two different Remote Sites with two different principals (SecurityContext UserDNs).

In this case, you can use the joker string "AnyDN" for SwSec:SignDN. The DN will then be the principal from the active (used) Remote Site.

For more information, refer to [Working with signed or encrypted messages](#) in the *Configuring the Axway Gateway/SWIFTNet environment* topic.

Examples

For examples of files containing SWIFTNet Signature List information, refer to [SWIFTNet dump to XML](#).

Axway Gateway: Connectors

SWIFTNet-specific user exits

Caution Please note that care should be taken when **customizing user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

About SWIFTNet-specific user exits

Gateway can use Java user-defined functions to obtain some of the SWIFTNet outgoing and incoming Transfer Request parameters from external sources (for example PassPort PM).

Two user exits are available:

- Control of outgoing requests and provision of the parameters to add to the SWIFTNet outgoing request
- Control of incoming requests and provision of the parameters to add to the SWIFTNet outgoing response

If you want to use this feature, you must first configure the use of SWIFTNet-specific user exits as described below.

Source files and compilation

The JAVA source files are installed in:

```
<Gateway installation directory>/src/javasrc/com/axway/gateway/userexit
```

This directory contains the following files:

- GwUserExit.java – JAVA interface
- SwiftNetIncomingXferRequest.java – SWIFTNet exit to control incoming requests
- SwiftNetOutgoingXferRequest.java – SWIFTNet exit to control outgoing requests

After modifying the files as required, compile them (javac) and replace the original source file:

```
<Gateway installation directory>/run_time/exithelper/userexit.jar
```

A sample user exit is provided in:

```
<Gateway installation directory>/samples/exithelper/
```

Afterwards, **restart** Gateway.

Configuring the use of SWIFTNet-specific user exits

1. In the configuration menu, select **Connectivity > Axway Connectivity > SWIFTNet**.
For full details of the following parameters, refer to [Configuration: Connectors](#).
2. Activate (or deactivate) **Activate user exit**.
3. Check, and if necessary modify, the following parameters:
 - **Address**
 - **Port**
 - **Timeout**

Managing user exit errors

When initiating SWIFTNet transfers, and the **Activate user exit** option is enabled, a transfer:

- **Starts** if all mandatory parameters for starting a transfer are provided to the user exit, or recovered from the user exit.
- **Fails** if the mandatory parameters needed by the user exit are not all present.

Consult the log for detailed error messages.

SWIFTNet exit to control outgoing requests: *SwiftNetOutgoingXferRequest*

Purpose

If the SWIFTNet **Activate user exit** option is enabled, then the exit controls the *outgoing* requests and provides the undefined parameters to include in the SWIFTNet outgoing request.

The communication between Gateway (SWIFTNet) and the user exit is done using a JAVA helper process. The data exchanged between these two entities is in XML format.

Syntax

The input XML (received by the user exit) has the following structure:

```
<%transfer_mandatory_param%>...</%transfer_mandatory_param%>
<%transfer_optional_param%>...</%transfer_optional_param%>
<%transfer_informative_param%>...</%transfer_informative_param%>
```

The output XML (sent by the user exit) must have the following structure:

```
<acceptation_code>...</acceptation_code>
<rejection_message>...</rejection_message>
<%transfer_optional_param%>...</%transfer_optional_param%>
```

where:

<%transfer_mandatory_param%> - The parameters that must be provided to the user exit.

<%transfer_optional_param%> - The parameters that can be recovered from the user exit if they were undefined in Gateway. The user exit will know that one of the optional parameters was undefined if it is not present in the XML data (the related tag will be missing). If an optional parameter was provided in Gateway as Transfer Request parameter, it will be sent to the user exit, but its value should not be overridden with another exit value.

%transfer_informative_param% - The parameters sent to user exit only as additional information about the transfer.

The table below describes the input and output XML tags for transmitting the outgoing transfer parameters to the exit and for receiving the transfer status and some transfer parameters from the exit.

Input and output XML tags

Type	XML Tag Name	Comments	Value
Output <i>Mandatory</i>	<acceptation_ code>	User exit answer about rejection or acceptance of outgoing Transfer Request.	• TRUE (transfer accepted) • FALSE (transfer rejected)

Type	XML Tag Name	Comments	Value
Output <i>Optional</i>	<rejection_message>	If <code>acceptation_code</code> is FALSE, a short message with the rejection reason is sent by the user exit.	Alphanumeric characters
	<file_pattern>	<i>FileAct Get (initiator) only</i> Specify a file name format to enable you to identify the received file in the specified directory. The file name pattern can contain symbolic variables.	Maximum: 256 characters
	<dir_path>	<i>FileAct Get (initiator) only</i> Specify the local physical path of the directory where received files are to be stored.	Maximum: 256 characters
	<payload_attribute>	<i>InterAct only</i> Specify payload attributes in the following format: <code>attrib1=\"value1\"</code>	Maximum: 80 characters
	<payload_compress_info>	<i>InterAct only</i> Specify payload compression information.	• <code>dt=\"GZIP-Base64\"</code> • <code>dt=\"ZLIB-Base64\"</code>
	<payload_type_option>	<i>InterAct only</i> Specify the payload format.	• text • embedded

Type	XML Tag Name	Comments	Value
Input <i>Mandatory</i>	<type>	Transfer type	• TRANS • MSG
	<mode>	Connection mode for the Transfer	• I (= Initiator) • R (= Responder)
	<direction>	Transfer direction	• I (= In) • O (= Out)
	<trade_org_alias>	The originator/local organization alias, linked to the collaboration profile	Maximum: 128 characters
	<trade_dest_alias>	The destination/remote organization alias, linked to the collaboration profile	Maximum: 128 characters
	<service>	SWIFT business service name	Maximum: 30 alphanumeric characters
	<request_type>	Type of request	Maximum: 30 alphanumeric characters

Type	XML Tag Name	Comments	Value
Input/ Output <i>Optional</i>	<file_name>	<i>FileAct only</i> Logical file name.	Maximum: 76 alphanumeric characters
	<snd_msg>	<i>FileAct only</i> [0 - 255]: Transfer description, supplied by the initiator. Free-text description of the transfer. [256 - 511]: Transfer information, supplied by the initiator. Structured information concerning the transfer.	SWIFTNet FileAct: Max 256 alphanumeric characters for each part. Total: 512 chars.
	<rcv_msg>	<i>FileAct only</i> [0 - 255]: File description, supplied by the sender. Free-text description of the file. [256 - 511]: File information, supplied by the sender. Structured information concerning the transfer. For more details, refer to the SWIFT documentation.	Maximum: 256 + 256 alphanumeric characters.
	<priority>	Transfer priority	• Urgent • Normal
	<receipt_request>	A return receipt is required when sending a file to a partner	• none • signed
	<receipt_dn>	<i>Only if receipt_request is signed.</i> Receipt DN	Maximum: 100 alphanumeric characters

Type	XML Tag Name	Comments	Value
	<receipt_request_type>	<i>FileAct, RealTime only</i> <i>Only if receipt_request is signed.</i> Type of acknowledgment request	Maximum: 30 alphanumeric characters
	<trade_org>	DN of the originating partner.	Maximum: 128 characters
	<trade_dest>	DN of the destination partner.	Maximum: 128 characters
	<trade_sign>	Transfer is signed.	• Y • N
	<trade_sign_dn>	<i>InterAct only</i> <i>Sent only if trade_sign is Y or non repudiation is Y.</i> If specified, name used for signing in Initiator mode. Otherwise the securityDN (supplied by SNL) is used.	Maximum: 100 alphanumeric characters
	<trade_encrypt>	<i>InterAct only</i> Transfer is encrypted.	• Y • N
	<trade_encrypt_dn>	<i>InterAct only</i> <i>Sent only if trade_encrypt is Y.</i> <i>Mandatory if trade_encrypt is Y.</i> If specified, name used for encryption in Initiator mode.	Maximum: 100 alphanumeric characters
	<non_repudiation>	Request SWIFT to keep proof of sending.	• Y • N
	<trade_delivery_mode>	The delivery mode of the file or message.	• RT • SNF

Type	XML Tag Name	Comments	Value
Input Informative	<trade_notif_queue>	<i>Only if trade_delivery_mode is SNF.</i> Reception queue for Store-and-Forward delivery notifications. It can be empty. If no value is specified, the tag is not added to the XML.	Maximum: 30 alphanumeric characters
	<trade_compress>	Use compression when sending a file or message to a partner.	• Y = Use compression • N = No compression
	<dest_alias>	Destination Site alias.	Maximum: 31 alphanumeric characters
	<org_alias>	Originator Site alias.	Maximum: 31 alphanumeric characters
	<user_ref>	User reference of the request/ response.	Maximum: 40 alphanumeric characters
	<receipt_type>	Specify the type of receipt to return to the initiator of the received transfer.	• SUCCESS • ERROR

Note: The XML Tag names are the same as the [Transfer Request parameter names](#). The maximum length for each string parameter can be found in <Gateway installation directory>/inc/itpdefs.h.

Example: User exit for an outgoing transfer

In this example, not all parameters needed to initiate a SWIFTNet transfer are given in the command line, but they are recovered from the user exit (for example, `trade_dest` and `trade_org`).

```
peltrans -ap FTP_A -oa LOCAL_SITE -da REMOTE_SITE -tradeoa ORIG_ORGANIZ
-tradedea DEST_ORGANIZ -fc c:\test.txt -sc swift.devsup.fa!x1
-rqt dc00.0000.0000
```

Gateway sends to user exit:

```
<type>TRANS</type>
<mode>I</mode>
<direction>O</direction>
<trade_org_alias>ORIG_ORGANIZ</trade_org_alias>
<trade_dest_alias>DEST_ORGANIZ</trade_dest_alias>
<service>swift.devsup.fa!xl</service>
<request_type>dc00.0000.0000</request_type>
<file_name>FTP_A</file_name>
<org_alias>LOCAL_SITE</org_alias>
<dest_alias>REMOTE_SITE</dest_alias>
```

Gateway receives from user exit:

```
<acceptation_code>TRUE</acceptation_code>
<priority>Normal</priority>
<trade_org>ou=p1,o=test,o=swift</trade_org>
<trade_dest>ou=p2,o=test,o=swift</trade_dest>
<trade_sign>Y</trade_sign>
<non_repudiation>Y</non_repudiation>
```

The user exit accepts the outgoing transfer and all basic parameters for initiating a transfer are available.

SWIFTNet exit to control incoming requests: *SwiftNetIncomingXferRequest*

Purpose

If the SWIFTNet **Activate user exit** is enabled, then the exit controls the *incoming* requests and updates some of the transfer parameters to include in the SWIFTNet outgoing response.

The communication between Gateway (SWIFTNet) and the user exit is done using a JAVA helper process. The data exchanged between these two entities is in XML format.

Syntax

The input XML (received by the user exit) has the following structure:

```
<%input_transfer_param%>...</%input_transfer_param%>
```

The output XML (sent by the user exit) must have the following structure:

```
<acceptation_code>...</acceptation_code>
```

```
<rejection_description>...</rejection_description>
<rejection_info>...</rejection_info>
<%output_transfer_param%>...</%output_transfer_param%>
```

where:

`%input_transfer_param%` - The same tags as specified in the [Input and output XML tag table](#), except that the `<trade_org_alias>` and `<trade_dest_alias>` parameters are not present.

`%output_transfer_param%` - The output parameters depend on the transfer type. Refer to the tables below:

- [FileAct PUT](#)
- [FileAct GET](#)
- [InterAct](#)

For all incoming XML, the user exit should send an XML response that contains the following tags:

XML response tags

XML Tag Name	Comments	Value
<code><acceptation_code></code>	<i>Mandatory</i> Incoming transfer is accepted or rejected by the exit.	• Accepted • Rejected
<code><rejection_description></code>	<i>FileAct:</i> Use to set the Sw:RejectDescription parameter. <i>InterAct:</i> Short message with the rejection reason.	Alphanumeric characters
<code><rejection_info></code>	<i>FileAct only</i> Use to set the Sw:RejectInfo parameter.	Alphanumeric characters

Output transfer parameters for FileAct PUT transfers

XML Tag Name	Comments	Value
<code><trade_org_alias></code>	The originator/local organization alias, linked to the collaboration profile.	Maximum: 128 characters

XML Tag Name	Comments	Value
<trade_dest_alias>	The destination/remote organization alias, linked to the collaboration profile.	Maximum: 128 characters
<trade_sign>	Transfer is signed.	• Y • N
<non_repudiation>	Request SWIFT to keep proof of sending.	• Y • N
<file_pattern>	Specify a file name format to enable you to identify the received file in the specified directory. The file name pattern can contain symbolic variables.	Maximum: 256 characters
<dir_path>	Specify the local physical path of the directory where received files are to be stored.	Maximum: 256 characters
<receipt_option>	Determines how acknowledgments are handled.	• ACK_BY_MONITOR • ACK_BY_USER
<appli>	Application name.	Any valid application name

Output transfer parameters for FileAct GET transfers

XML Tag Name	Comments	Value
<trade_org_alias>	The originator/local organization alias, linked to the collaboration profile.	Maximum: 128 characters
<trade_dest_alias>	The destination/remote organization alias, linked to the collaboration profile.	Maximum: 128 characters
<trade_sign>	Transfer is signed.	• Y • N

XML Tag Name	Comments	Value
<non_repudiation>	Request SWIFT to keep proof of sending.	• Y • N
<appli>	Application name.	Any valid application name
<receipt_request>	A return receipt is required when sending a file to a partner.	• none • signed
<receipt_dn>	<i>Only if receipt_request is signed</i> Receipt DN.	Maximum: 100 alphanumeric characters
<receipt_request_type>	<i>Only if receipt_request is signed</i> Type of acknowledgement request.	Maximum: 30 alphanumeric characters

Output transfer parameters for InterAct transfers

XML Tag Name	Comments	Value
<trade_org_alias>	The originator/local organization alias, linked to the collaboration profile.	Maximum: 128 characters
<trade_dest_alias>	The destination/remote organization alias, linked to the collaboration profile.	Maximum: 128 characters
<trade_sign>	Transfer is signed.	• Y • N
<trade_sign_dn>	Name used for signing the response. Otherwise the securityDN (supplied by SNL) is used.	Maximum: 100 alphanumeric characters
<trade_encrypt>	Transfer is encrypted.	• Y • N

XML Tag Name	Comments	Value
<trade_encrypt_dn>	<i>Mandatory</i> if trade_encrypt is Y.	Maximum: 100 alphanumeric characters
<non_repudiation>	Request SWIFT to keep proof of sending.	• Y • N
<receipt_request>	<i>InterAct SnF only</i> A return receipt is required when sending a file to a partner.	• none • signed
<payload_attribute>	Specify payload attributes in the following format: attrib1=\"value1\"	Maximum: 80 characters
<payload_compress_info>	Specify payload compression information.	• dt=\"GZIP-Base64\" • dt=\"ZLIB-Base64\"
<file_component>	Name of the local file containing the default reply message.	Maximum: 256 alphanumeric characters
<receipt_request_sync>	<i>InterAct RT only</i> Activate the waiting period for a Transfer receipt containing the reply message.	• Y • N
<payload_type_option>	Specify the payload format.	• text • embedded
<appli>	Application name.	Any valid application name

Examples: User exits for incoming transfers

Example 1: FileAct PUT

```
peltrans -ap FTP_A -oa LOCAL_SITE -da REMOTE_SITE
        -tradeoa ORIG_ORGANIZ -tradedda DEST_ORGANIZ -fc c:\out.txt
```

```
-sc swift.devsup.fa!x1 -rqt dc00.0000.0000

-tradeo ou=p1,o=test,o=swift -traded ou=p2,o=test,o=swift
```

Gateway sends the incoming transfer parameters to the user exit:

```
<type>TRANS</type>
<mode>R</mode>
<direction>I</direction>
<service>swift.devsup.fa!x1</service>
<request_type>dc00.0000.0000</request_type>
<trade_org>ou=p1,o=test,o=swift</trade_org>
<trade_dest>ou=p2,o=test,o=swift</trade_dest>
<trade_sign>N</trade_sign>
<trade_encrypt>N</trade_encrypt>
<non_repudiation>N</non_repudiation>
```

Gateway receives from the user exit:

```
<acceptation_code>Accepted</acceptation_code>
<trade_sign>Y</trade_sign>
<non_repudiation>Y</non_repudiation>
```

Example 2: FileAct GET

This example uses two **peltrans** commands. First, the Server/Responder makes the file available, then the Client/Initiator takes the file.

Responder side:

```
peltrans -ty TRANS -dir O -mode R -ap FTP_A -oa ITEM-51537

-da RSWIFT -tradeoa ORGANIZ_ORIG -tradedda ORGANIZ_DEST

-fc c:\out.txt -sc swift.devsup.fa!x1 -rqt dc00.0000.0000
```

Initiator side:

```
peltrans -ty TRANS -dir I -mode I -ap FTP_A -oa RSWIFT

-da ITEM-51537 -tradeoa ORGANIZ_DEST -tradedda ORGANIZ_ORIG

-sc swift.devsup.fa!x1 -rqt dc00.0000.0000

-tradeo ou=p1,o=test,o=swift -traded ou=p2,o=test,o=swift
```

Gateway sends the incoming transfer parameters to the user exit:

```
<type>TRANS</type>
<mode>R</mode>
```



```

<direction>O</direction>
<service>swift.devsup.fa!x1</service>
<request_type>dc00.0000.0000</request_type>
<trade_org>ou=p1,o=test,o=swift</trade_org>
<trade_dest>ou=p2,o=test,o=swift</trade_dest>
<trade_sign>N</trade_sign>
<trade_encrypt>N</trade_encrypt>
<non_repudiation>N</non_repudiation>

```

Gateway receives from the user exit:

```

<acceptation_code>Accepted</acceptation_code>
<trade_sign>Y</trade_sign>
<receipt_request>signed</receipt_request>
<receipt_request_type>dc11.1111.1111</receipt_request_type>
<non_repudiation>Y</non_repudiation>

```

Enabling logging inside user exit code

Standard output and standard errors of the user exit are, by default, redirected to the null device (messages printed via system.out interface are discarded). Therefore, if you need to write logging information regarding the processing of user exits, it is recommended to use log4j or a similar logging facility. A log4j configuration sample is provided in the samples.

To enable logging inside user exit code:

1. Copy the `log.properties` sample in the user exit directory.
2. Append the path to the `log.properties` file in the `userexit_command` parameter (`-Dlog4j.configuration=<path to log.properties>`).
3. Use `peluconf get` to retrieve the old command from the `userexit` section: (`peluconf get -s userexit userexit_command`) and copy it in a text editor.
4. Append the path to the command and write the new string in the configuration with `peluconf`: (`peluconf set -s userexit_command "<new value>"`).

Axway Gateway: Connectors

SWIFTNet system recovery

[Overview](#)

[Automatic system recovery](#)

[Selecting marked transfers](#)

[Displaying transfer details](#)

[Manually resending lost transfers](#)

Overview

System recovery ensures that all SWIFTNet Store-and-Forward transfers are successfully sent to the partner even in case of a SWIFT network crash.

Gateway has an automatic mechanism to detect a potential SWIFT crash based on output session events and the last replication time on the SWIFT side. When Gateway detects a system recovery from SWIFT, it performs a scan of all SWIFT SnF outgoing transfers to see if there is a risk that they were made after the last SWIFT database replication and marks them accordingly.

You can then manually re-send the files that were detected by using a specific algorithm (manually setting possible duplicates, using the right creation time, and so on).

Automatic system recovery

Automatic system recovery is triggered by the following actions:

- Each time Gateway starts: Gateway automatically asks SWIFT for the replication history and triggers the system recovery procedure if a crash occurred while it was offline.
- When Gateway receives a message from SWIFT that an output session was closed, after the SnF infrastructure is back online (signaled by the ability to open a new output session) Gateway asks SWIFT for the replication history and triggers the system recovery procedure if a crash occurred when the output session was closed.

Manual system recovery

In addition to the automatic system recovery detection, you can trigger the detection procedure manually using the `pelctl detect_swift_recovery` command.

Selecting marked transfers

When selecting transfers using the `peldsp select_trans` command, you can use the selection filter `sw_marked_by_system_recovery (-swmksysrec) { U | Y | N }` where:

- U = Undefined (default)
- Y = Transfers marked by system recovery
- N = Transfers that were not marked

Displaying transfer details

Use the `peldsp display_trans` command to display the details of a transfer.

In the details, Gateway displays information about system recovery such as:

```
x_sw_marked_by_system_recovery='Y|N'
```

Manually resending lost transfers

You must manually resend each lost transfer detected during the system recovery procedure using the `peltrans` command. In addition to the original parameters of each transfer, manually set the following parameters:

- The Possible Duplicate indicator `sw_pd_indication`
- Exactly the same message id as the original transfer
- Exactly the same creation time as the original transfer with the `sw_creation_time` parameter

Axway Gateway: Connectors

SWIFTNet polling mechanism

The following sections describe how the SWIFTNet FileAct TransferStatus polling mechanism enables Gateway to detect transfers stuck in process and update their states accordingly.

Overview

With help from a file handler application that is part of Swift RA, Gateway starts a file transfer after sending a request to, or receiving a request from, Swift. File events inform Gateway about transfer processing states (Swift TransferStatus). Gateway updates the transfer states as necessary after receiving each file event. A transfer reaches a final state when Gateway receives a final file event with a TransferState of Completed, Aborted, Failed, Unknown, Rejected or Duplicated.

Rarely, Gateway cannot receive and process a final file event (for example, because of a high-load network connectivity issue). Then, the transfer state is not updated, and the transfer remains in an intermediary state.

The role of the polling mechanism is to:

- Detect transfers stuck in intermediary states by determining that Gateway has stopped receiving file events (FileEvent).
- Use a parallel Swift-provided mechanism (GetFileStatus) to interrogate Swift about the transfer status.

Detecting transfers stuck in intermediary states

A missed file event can result in one of the following states for an in-process transfer:

- TO_BEGIN
- TO_RESTART
- CREATED
- SERVICING
- PROGRESSING

A transfer can be in the TO_BEGIN or TO_RESTART state without being considered stuck.

Gateway polls, at a user configurable interval, the list of transfers in intermediary states. Gateway selects for a GetFileStatus operation only transfers that meet the following conditions:

- Initiated (file request sent to Swift) or received (file request received from Swift)
- Not in a final state (for example, ENDED, ENDED_TO_ACK, CANCELLED).
- No file events have been received recently. The time to elapse before a transfer is deemed stuck is configurable.

Transfer list polling is a fast method of scanning the transfer list directly from memory, rather than accessing a mailbox file. However, you can enable and disable polling dynamically. The default is disabled.

Even when transfer list polling is disabled, Gateway at startup scans the transfer list once and updates transfers in intermediary states. However, initiator transfers in TO_BEGIN or TO_RESTART states are scheduled for reprocessing and are not selected for GetFileStatus operations.

Performing a GetFileStatus operation

Gateway sends a GetFileStatus request to Swift for each selected transfer in a GetFileStatus operation. In return, Gateway receives the actual TransferState, as if received via a file event. A GetFileStatus request is the equivalent of receiving a file event, resulting in recovering the stuck transfer.

The following are examples of log events for a GetFileStatus operation:

```
SNET080W 10.05.2011 15:43:31 SAGLIVEGTW(rnd9_mp) [100] Checking status  
for pending FileAct transfer.
```

```
SNET080W 10.05.2011 15:45:12 SAGLIVEGTW(rnd9_mp) [100] Swift status for  
pending FileAct transfer is: Completed; updating transfer state
```

Configuration parameters

The transfer status polling interval parameter specifies the frequency Gateway scans the transfer list. The transfer status polling delay parameter specifies the time to elapse before a transfer is deemed stuck. The following provides more details:

- **Transfer status polling interval.** The interval in seconds the transfer list is scanned for GetFileStatus operation candidates. A value of **0** disables polling.
- **Transfer status polling delay.** Seconds elapsed since the last file event.

SWIFT keeps a history of completed transfer information, but there are limitations on the number and age of stored transfers. The polling mechanism might be triggered for a transfer that no longer has FileStatus information available. If so, the transfer is canceled and an error writes to the log.

SWIFTNet ASP check commands

pelctl reload_asp_profiles: Trigger reload of ASPs

Syntax	<code>pelctl reload_asp_profiles</code>
Description	This command triggers reloading of ASPs. This is useful when you have: • Updated the current ASP profiles directory. • Modified the ASP directory path and want to reload the ASPs from the new directory without restarting Gateway. If the reload operation fails, the last successful loaded ASP package files remains in use. When more ASP helpers are configured, the reload is considered successful only if all helpers loaded the ASP package.
Example	Log output for a successfully reloaded ASP package: <pre>SNET I ASPLDSTS Swift ASP package reload done. Files: 179, Date: 20120319, Start: 145726, Stop: 145726</pre> Log output for a failed ASP package reloading operation: <pre>SNET E ASPLDSTS Swift ASP package reload failed. Reason: Could not open or parse the digest file : [D:\swift7_official_asp_ package\ApplProf.dig], ErrNo: 16, Date: 20120319, Start: 150433, Stop: 150433, Previous active load: NONE!</pre>

pelctl check_asp_status: Check result of ASPs reload

Syntax	<code>pelctl check_asp_status</code>
Description	This command checks the result of an operation to reload ASPs.

Example

Log output for a successfully reloaded ASP package:

```
ASP profiles reloaded status: Ok.
Load started/finished: 03/19/2012 10:42:54 /
03/19/2012 10:42:59.
Last successful load: 03/19/2012 10:42:59, 179
ASP profiles loaded.
ASP directory for loading: D:\swift7_official_
asp_package.
```

Log output for a failed ASP package loading:

```
ASP profiles loaded status: error encountered
during load process.
Last load attempt: 01/26/2012 17:46:38
Last successful load: 01/26/2012 17:44:29, 179
ASP profiles loaded.
Error code: 16, error message: Could not open
or parse the digest file : [D:\wrong_asp_
package_dir\ApplProf.dig]
ASP directory for loading: D:\swift7_official_
asp_package.
```

Axway Gateway: Connectors

SWIFTNet ASP check commands

pelctl reload_asp_profiles: Trigger reload of ASPs

Syntax	<code>pelctl reload_asp_profiles</code>
Description	This command triggers reloading of ASPs. This is useful when you have: - Updated the current ASP profiles directory. - Modified the ASP directory path and want to reload the ASPs from the new directory without restarting Gateway. If the reload operation fails, the last successful loaded ASP package files remains in use. When more ASP helpers are configured, the reload is considered successful only if all helpers loaded the ASP package.

Example

Log output for a successfully reloaded ASP package:

```
SNET I ASPLDSTS Swift ASP package reload done.
Files: 179, Date: 20120319, Start: 145726,
Stop: 145726
```

Log output for a failed ASP package reloading operation:

```
SNET E ASPLDSTS Swift ASP package reload
failed. Reason: Could not open or parse the
digest file : [D:\swift7_official_asp_
package\ApplProf.dig], ErrNo: 16, Date:
20120319, Start: 150433, Stop: 150433,
Previous active load: NONE!
```

pelctl check_asp_status: Check result of ASPs reload

Syntax

pelctl check_asp_status

Description

This command checks the result of an operation to reload ASPs.

Example

Log output for a successfully reloaded ASP package:

```
ASP profiles reloaded status: Ok.
Load started/finished: 03/19/2012 10:42:54 /
03/19/2012 10:42:59.
Last successful load: 03/19/2012 10:42:59, 179
ASP profiles loaded.
ASP directory for loading: D:\swift7_official_
asp_package.
```

Log output for a failed ASP package loading:

```
ASP profiles loaded status: error encountered
during load process.
Last load attempt: 01/26/2012 17:46:38
Last successful load: 01/26/2012 17:44:29, 179
ASP profiles loaded.
Error code: 16, error message: Could not open
or parse the digest file : [D:\wrong_asp_
package_dir\ApplProf.dig]
ASP directory for loading: D:\swift7_official_
asp_package.
```

Axway Gateway: Connectors

SWIFTNet ASP check

The following sections provide information about Gateway support for checking sent and received transfers via SWIFTNet against Application Service Profiles.

About ASP check for SWIFTNet

Starting with SWIFTNet 7.0, information is available related to all services, including InterAct and FileAct (not only for FINCopy services). Each service has an Application Service Profile (ASP). This is an XML file containing all service parameters. Based on the ASP package for all published SWIFT services loaded into Gateway, checks can be made when sending or receiving traffic to ensure use of correct transfer parameters for a service.

By performing ASP filtering, SWIFT does not reject the initiated flows for incorrect transfer parameter configuration.

Another use for ASP checking is determining which services require Relationship Management Application (RMA) interface authorization filtering.

For more details about ASP, refer to the official SWIFT documentation.

Prerequisites

Before using Gateway to check sent and received transfers via SWIFTNet against ASPs, you must provide the ASP package content for loading. ASP packages are available at <http://www.swift.com>. Decompress downloaded package archives before loading into Gateway.

Enabling ASP checking

You must configure global parameters before using the SWIFTNet ASP check functionality:

Use the following steps to enable SWIFTNet ASP checking and set the path to ASP profiles.

1. In the left pane of the main window in the user interface, right-click the Gateway server to enable for ASP checking and select **Configure**. The Configuration window is displayed.
2. In the left pane of the Configuration window, select **Connectivity > Axway Connectivity > SWIFTNet (ASP/RMA tab)**.
3. Enable ASP checking by selecting one of the following:
 - **ASP & RMA enabled.** Both ASP and RMA checks are performed (see *Axway Gateway: Connectors on page 2169*)
 - **ASP enabled, RMA disabled.** Only ASP check is performed

For details see *Connectivity > Axway Connectivity > SWIFTNet on page 229*

4. Specify the ASP profiles directory in the ASP Profiles Directory. Gateway environment variables can be used:
 - %VAR% and \ separator for Windows
 - \$VAR and / separator for UNIX

The folder path name is a dynamic configuration parameter and can be updated at runtime. To reload and use the ASP files from the new location, see *pelctl reload_asp_profiles: Trigger reload of ASPs on page 2164*.

5. Click **Activate** to enable the connector.
6. Restart Gateway for the change in configuration to take effect.

ASP package load or reload

Upon startup, Gateway validates ASP package integrity before service profiles in the ASP directory are loaded to memory when ASP checking is enabled.

In addition to the service profile files, the package must contain:

- An XML file with the digests of all the other files from the package (service profiles, XML schema files).
- The XML schema for validating the XML digest file.
- The XML schema for validating the service profiles.

All files enumerated into the digest file must be in the ASP directory. The digest Gateway computes for each file must be identical to the digest specified in the digest file in the ASP package.

You can change the values of the ASP profile schema, profile extension, digest file and digest schema for checking the package integrity and loading the ASPs. However, the best practice is to use the default values unless change is absolutely necessary (for example, SWIFT changes file names).

If errors occur at Gateway startup while checking ASP package integrity or loading ASPs, the ASP helpers are restarted and errors logged. For example:

```
SUP I ST-HPROC Starting 'swasp' helper process number 1

SNET E ASPLDSTS Swift ASP package load failed. Reason: Could not open or
parse the digest file : [D:\swift7_official_asp_package\ApplProf.dig],
ErrNo: 16, Date: 20120319, Start: 150433, Stop: 150433, Previous active
load: NONE!

JCT I JOBTERML [H_SWASP_1] Job [h_swasp_job -v (pid=5660, ph=1532)] pid
5660 terminated
```

Unless you correct the error, Gateway continually tries to load the package, and multiple errors are logged. The following are possible causes of ASP loading failures:

- One or more of the following parameters are incorrect: ASP profiles directory, profile schema, profile extension, profiles digest file, digest schema.
- One or more of the ASPs are incorrect.

When you have corrected the ASP configuration and restart the ASP helpers, the ASPs are loaded successfully and the action logged. For example:

```
SNET072I 06.02.2012 17:43:58 Swift ASP package load done. Files: 34,
Date: 20120206, Start: 174358, Stop: 174358.
```

You can use `pelctl` commands to reload a new or updated ASP package into memory while Gateway is running. See [Axway Gateway: Connectors on page 2164](#).

Outgoing and incoming checks

The next transfer parameters are checked for outgoing and incoming transfer requests when the global configuration option is enabled for ASP checking.

Outgoing requests

After a transfer is created into mailbox, but before sending the outgoing request to SWIFT, an ASP check is performed to validate the correct usage of the following transfer parameters:

- **Request type**
- **Delivery mode**
- **Signing**
- **Signature format**
Checked only when signing is set.
- **Non-repudiation**
- **Headerinfo feature used**
- **Copy feature used**
- **3rd parties DN list**
Checked only when copy feature is used.
- **Authorization notification indication used**
For store-and-forward mode, this is checked when the copy feature is used.

Possible results for ASP check are:

- **Failed**
- **Success** followed by **RMA check required, RMA check required (in trial period)** or **RMA check not required** information message when RMA check is enabled from the configuration.
- **Success with default profile** followed by **RMA check required, RMA check required (in trial period)** or **RMA check not required** information message when RMA check is enabled from the configuration.

If the ASP check fails, the transfer cancels. This is reflected in the log.

Incoming requests

After receiving a transfer request from SWIFT, but before creating an entry in the mailbox for the incoming request, Gateway performs an ASP check only if the global configuration option enables both ASP and RMA checking. This is because the scope of the check is to determine whether RMA filtering is required for the incoming request. If an error occurs during the ASP check, the error writes to the log and Gateway rejects the transfer request.

Possible results for ASP checking are:

- **Failed**
- **Success** followed by **RMA check required, RMA check required (in trial period)** or **RMA check not required** information message.
- **Success with default profile** followed by **RMA check required, RMA check required (in trial period)** or **RMA check not required** information message.

Before sending the response for the incoming request to SWIFT, Gateway does another ASP check to validate the correct usage of the non-repudiation parameter value in the response. Gateway generates a warning if the check fails. The non repudiation parameter in the response is copied from the non repudiation in the incoming request.

If Gateway cannot find the profile for a service, it uses the corresponding default profile for the service environment depending on the service name suffix (!p, !x, !xp). This occurs rarely. The default profile contains default parameters, which might not be the correct configuration for the selected service. To avoid this, use an updated ASP package.

There is a `peldsp display_trans` parameter for reporting the result of the ASP check. The parameter is `x_asprma_check_result`. Use it for displaying the check result for ASP and RMA validation. In the context of ASP checking a value of **ASP Successful** means ASP validation against the corresponding profile succeeded. If this value is missing, the ASP checking failed.

Axway Gateway: Connectors

Gateway RMA check

The following sections provide information about Gateway support for Relationship Management Application (RMA) authorization checking.

Note The Gateway RMA component is delivered with the Gateway kit, and is installed only if the license key contains the Gateway RMA option.

About RMA traffic filtering for SWIFTNet

Relationship Management Application (RMA) is a SWIFTNet service that allows customers to control the traffic they are willing to accept from other customers.

By enabling the RMA filtering support in Gateway, RMA authorization checking is performed at the sender or receiver side using an RMA user exit library. This enables the user to stop unwanted traffic. The exit library is delivered with the Gateway RMA component upon installation, only if the license key contains the RMA option, which is a component that manages RMA authorizations.

Based on the Application Service Profile (ASP) check, Gateway determines which services require RMA interface authorization. RMA filtering user exit is called before sending a transfer request or after receiving a transfer request using a service requesting RMA filtering. Depending on the RMA authorization check result, the transfer:

- Continues if valid RMA authorization is found
- Aborts if no authorization or no valid authorization is found for outgoing requests
- Rejects if no authorization or no valid authorization is found for incoming requests

For more information, refer to the [Gateway RMA section](#) of the documentation

Prerequisites

Do the following before using Gateway to perform RMA authorization checking:

- Enable ASP and RMA checking (see [Axway Gateway: Connectors on page 2165](#)).
- Install Gateway RMA and configure it as described in the [Gateway RMA section](#) of the documentation.
- Set the locations of the RMA user exit library and the RMA user exit configuration file delivered with Gateway RMA.

Note The Gateway RMA feature needs a valid license key in order to function. Enabling Gateway RMA without a proper license key will result in Gateway SWIFTNet processes being unable to start.

Enabling RMA authorization checking

You must configure global parameters before using the Gateway RMA authorization check functionality:

Use the following steps to enable Gateway RMA authorization checking and set the paths to the RMA user exit library and RMA user exit configuration file.

1. In the left pane of the main window in the user interface, right-click the Gateway server to enable for RMA checking and select **Configure**. The Configuration window is displayed.
2. In the left pane of the Configuration window, select **Connectivity > Axway Connectivity > SWIFTNet (ASP/RMA tab)**.
3. Enable RMA checking by selecting **ASP & RMA enabled**. Both ASP and RMA checks are performed.

For details see [Connectivity > Axway Connectivity > SWIFTNet on page 229](#)

4. Specify the following:
 - Path to the user exit library in the RMA User Exit Library field
 - Path to the RMA user exit configuration file in the RMA Configuration field

Gateway RMA provides the user exit library and configuration file.

Note Default values for these paths are present in the Gateway configuration. If the Gateway RMA component was installed, they already point to the RMA library and conf file. These values do not usually require editing.

5. Restart Gateway for the change in configuration to take effect.

RMA user exit library load

When RMA checking is enabled, SWIFTNet client and server processes load the RMA user exit library at Gateway startup using the specified RMA user exit configuration file.

If errors occur during RMA user exit library loading or initialization, the SWIFTNet processes performing the operation are terminated. Correct the errors reported in the Gateway log.

An incorrectly configured RMA can prevent the SWIFTNet module from functioning.

Scope of RMA checking

RMA user exit checking entails searching for a valid authorization in the Gateway RMA authorizations data store. Transfers are not sent or received when valid authorizations are not found.

RMA user exit checking occurs when sending and receiving data to verify whether traffic is authorized:

- When sending to a partner, RMA checking verifies whether there is a valid authorization-to-send a document to the partner.
- When receiving from a partner, RMA checking verifies whether there is a valid authorization-to-receive a document from the partner.

When the service is in RMA Trial Period (as specified in the ASP) and the stop unauthorized transfers in Trial Period global configurations flag is not set, processing continues even if the transfer fails RMA authorization checking.

There is a `peldsp display_trans` parameter for reporting the result of the RMA check. The parameter is `x_asprma_check_result`. Use it for displaying the check result for ASP and RMA validation. There are the following values in the context of RMA checking:

- **RMA Performed.** RMA authorization checking was performed. This does not indicate whether authorization is valid.
- **RMA Successful .** RMA authorization for the message was successful.
- **RMA Trial.** RMA validation was performed in trial period for the specified service.

SWIFTNet HA overview

Standard Gateway installations cannot be scaled horizontally due to limitations of Gateway itself: when correlating an acknowledgment with a previously sent transfer, Gateway must have the transfer in its mailbox. If the transfer was sent by another Gateway or it has been deleted from its mailbox, the acknowledgment is rejected.

The High Availability solution removes this limitation, storing instead necessary transfer information in a shared correlation database. This way, SWIFTNet MRR can be used to distribute the incoming traffic on several Gateway installations running in parallel.

An incoming acknowledgment or notification system message will be correlated against the shared correlation database, instead of the local Gateway mailbox. Information contained in such acknowledgment or notification messages can be routed to a back-end application like Integrator.

Note SWIFTNet MRR" refers to "SWIFTNet Message Reception Registry". This is the registry of rules that are set by any SWIFT Customer defining the way he expects SWIFTNet to send messages. These MRRs can define nominal and backup connection points inside the Customer's Information System, as seen from a SWIFTNet Network point of view. These connection points refer either to "SWIFTNet Link" instances or to "StoreAndForward queues". By defining nominal and backup SWIFTNet Link instances, the SWIFT Network is able to switch in case of failure of the nominal connection. For more information about MRR, refer to the official SWIFT documentation.

SWIFTNet HA overview

Standard Gateway installations cannot be scaled horizontally due to limitations of Gateway itself: when correlating an acknowledgment with a previously sent transfer, Gateway must have the transfer in its mailbox. If the transfer was sent by another Gateway or it has been deleted from its mailbox, the acknowledgment is rejected.

The High Availability solution removes this limitation, storing instead necessary transfer information in a shared correlation database. This way, SWIFTNet MRR can be used to distribute the incoming traffic on several Gateway installations running in parallel.

An incoming acknowledgment or notification system message will be correlated against the shared correlation database, instead of the local Gateway mailbox. Information contained in such acknowledgment or notification messages can be routed to a back-end application like Integrator.

Note SWIFTNet MRR" refers to "SWIFTNet Message Reception Registry". This is the registry of rules that are set by any SWIFT Customer defining the way he expects SWIFTNet to send messages. These MRRs can define nominal and backup connection points inside the Customer's Information System, as seen from a SWIFTNet Network point of view. These connection points refer either to "SWIFTNet Link" instances or to "StoreAndForward queues". By defining nominal and backup SWIFTNet Link instances, the SWIFT Network is able to switch in case of failure of the nominal connection. For more information about MRR, refer to the official SWIFT documentation.

SWIFTNet HA flow changes

[RECEIPT transfers routing to Integrator](#)

[Extension of SWIFTNet dump to XML](#)

[Forced retries on non-correlated Acknowledgments](#)

[Limitations](#)

The following changes were introduced when implementing Gateway High Availability:

RECEIPT transfers routing to Integrator

In order for the backend application to centralize and correlate the incoming Acknowledgment messages, it is now possible to define decision rules that route these Acknowledgment messages to Integrator. For more information, see [Axway Integrator connector](#).

Extension of SWIFTNet dump to XML

To make Notification-type system messages content available to the back-end application, Gateway dumps the content for Authorization, Refusal, Delivery Notification and Failed Delivery Notification system messages to XML files.

For a complete list of information that will be dumped to XML files see

[SWIFTNet dump to XML](#).

Forced retries on non-correlated Acknowledgments

If the Gateway High Availability solution cannot correlate an incoming Acknowledgment or Notification system message against the High Availability database, it returns an empty response. In effect, a retry could be triggered. For Store-and-Forward flows, this is done automatically by the SWIFTNet infrastructure (up to 10 retries). For Real Time flows, retries are performed by the remote partner.

Limitations

The following limitations apply to a Gateway High Availability solution:

FileAct RealTime Get flows do not work in a High Availability configuration

Do not use FileAct RealTime Get in a High Availability configuration. You will need to configure a separate, non-High Availability Gateway in order to use them.

A file can be put at disposal by a Gateway node, and the request to get the file might be received by another Gateway node, because in High Availability mode, any Gateway node might be replaced on the fly by any other Gateway node.

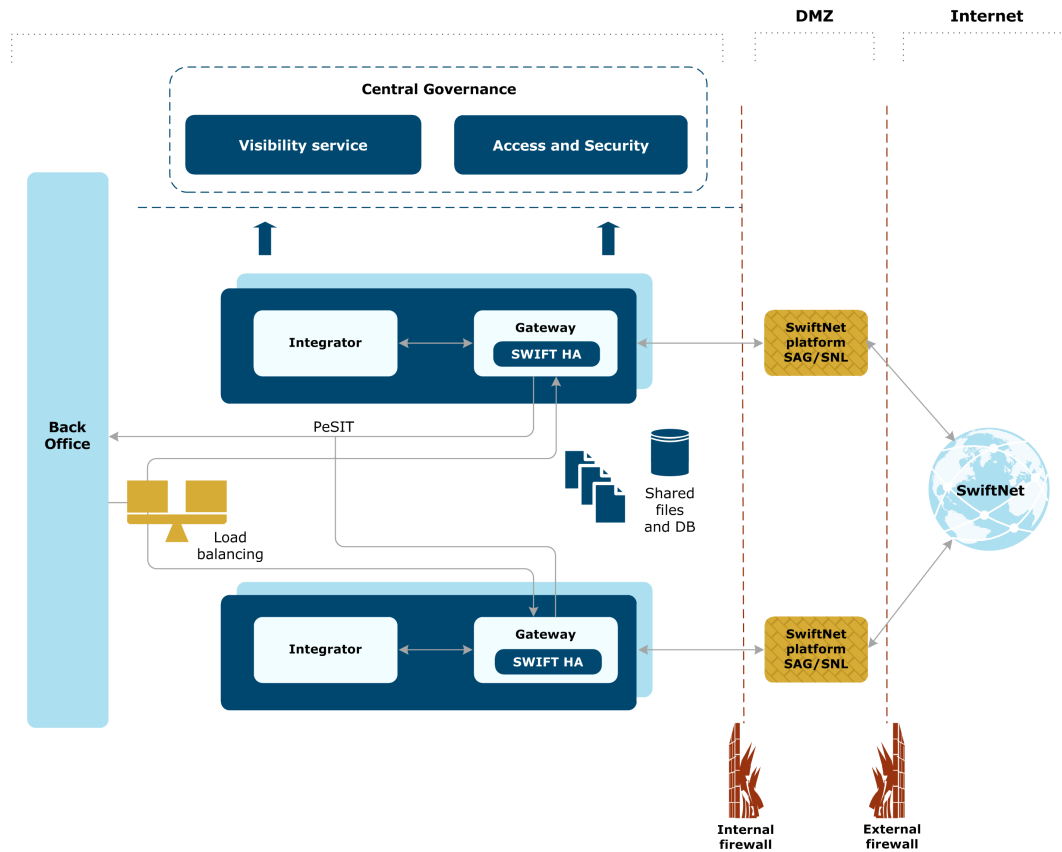
Incomplete end state transitions for Acknowledged transfers

Because in a High Availability configuration an Acknowledgment for a transfer can be received by a different Gateway to the transfer's sender, the transfer will not be updated to the Acknowledged state - instead remaining in Ended-to-Ack.

However, the Gateway receiving the Acknowledgment still sends Ack notifications to Sentinel (if Sentinel is enabled in the configuration). Note that the Tracking Object used in the context of SWIFTNet High Availability must have the "null management" option set.

SWIFTNet HA architecture

This following diagram provides an example of the architecture of a high-availability implementation in a bank for the SWIFT channel:



This example of a SWIFTNet high-availability architecture relies on several features:

- The SWIFT features called “distribution of real-time traffic over several SWIFTNet Links” and “shared Store-and-Forward (SnF) queues”.
- The SWIFTNet high-availability mechanism integrated in Gateway (from version 6.15) and providing a delivery notification pass-through service. This service enables a Gateway instance to receive a SWIFTNet delivery notification and associate it with the initial SWIFTNet transfer issued by another Gateway.
- A farm of active nodes where Integrator and Gateway are running.

Inbound flows

- The inbound SWIFTNet flows are balanced between the two Gateways through the “distribution of real-time traffic over several SWIFTNet Links” feature (configurable for real-time flows in the SWIFTNet Message Routing Rules) and through the sharing of store-and-forward queues. Each Gateway runs on one node of the farm.
- If one of the incoming flows is a delivery notification, then the pass-through service of the Gateway receiving the flows associates the notification with the initial transfer.
- After they have been processed by the Integrator running on the same node as the Gateway instance which received the flows, the inbound flows are pushed to the back-office through a Transfer Bus implementation to which the Gateway is connected.

Outbound flows

- The flows generated by the applications are distributed between the Gateway instances. Each instance runs on one node of the farm.
- After they have been processed by Integrator, the outbound flows are sent to the SWIFTNet network by Gateway. Gateway stores the transfer information in an RDBMS that will later make the association with the delivery notifications received.

Note In case of failure, each node is recovered by a backup node instance through a mechanism which is similar to the active/passive clustering mechanism.

Purge operation

For purging the HA database transfer entries, use the `pelpur` subcommand [pelpur clean swha db](#).

SWIFTNet HA user exit Configuration

[Prerequisites](#)

[Enabling the SWIFTNet High Availability feature](#)

[High availability user exit load and initialization](#)

Prerequisites

The prerequisites for activating the SWIFTNet High availability feature are:

- The High Availability Database (Oracle 11g or MySQL 5.6.5 or later) must be available. An ODBC connection to the database must be configured.
- Each Gateway node installation from a High Availability solution must connect to SWIFT through its own SAG.
- When installing Gateway, the High Availability option must be used.
- Each Gateway from the High Availability environment must be assigned a unique node identifier (See [About Gateway Node ID](#)).
- For monitoring Gateway from Sentinel you must use the file "`XFB_ALL_EN_HA.xml`", not "`XFB_ALL_EN.xml`", located on the installation DVD in the `/extras/sentinel` folder.

Note: The difference between the two files is determined by the "`nullMngt`" option, which for SWIFTNet High Availability has the value "Last", instead of "Default": the event updates will be done relative to the last update. In cases where you are generating the Tracking object, set the "null management" option.

Note The SWIFTNet High Availability feature requires a valid license key in order to function.

Enabling SWIFTNet High Availability without a proper license key will result in Gateway SWIFTNet processes being unable to start.

About Gateway Node ID

In environments where multiple Gateways are installed, for example connected to the same back-end application, it is desirable to have unique transfer identifiers for each Gateway installation. This functionality can be enabled by assigning each Gateway a unique node identifier. By default, this functionality is disabled.

A node identifier can have values from 0 to 15, to a total of 16 nodes. If set, the transfer identifier allocation changes as follows:

- Transfer identifiers are no longer allocated contiguously. Instead, a new transfer identifier will be the next number multiple of 16 + the node identifier (thus incremented by 16). This ensures that each Gateway allocates its own subset of transfer identifiers.
- There is no change for transfers already in the mailbox; so it is possible to have old transfers not conforming with the above rule. However, activating the node identifier feature should be done on a new installation.

Setting a Gateway node identifier

The Gateway node identifier can be set using the `peluconf` command:

```
peluconf set -s monitor node_id <node identifier>
```

where the node identifier ranges from 0 to 15.

There is no equivalent command to unset the node identifier; effectively, this means that once activated, the node identifier feature cannot be disabled. A Gateway can be set to a different node identifier.

Ensure that each Gateway is assigned a unique node identifier.

Enabling the SWIFTNet High Availability feature

You must configure global parameters before using the SWIFTNet High Availability feature. This can be partially done using the Graphical User Interface, and/or the command line configuration.

Use the following steps to enable SWIFTNet High Availability and set the paths to the HA user exit library. If not properly defined, the Gateway SWIFTNet client and server processes will be unable to start.

1. Set the `node_id` parameter to a value between 0 and 15.
2. In the left pane of the main window in the user interface, right-click the Gateway server to enable for High Availability and select **Configure**.
The Configuration window is displayed.
3. In the left pane of the Configuration window, go to **Connectivity > Axway connectivity > SWIFTNet** and select the **HA** tab.
4. From the **High Availability Exit** drop-down list, select `Use correlation exit` to enable the *High Availability with User Exit* feature.

Alternatively, enable the High Availability feature from the command line configuration:

```
peluconf set -s swnet swnet_ack_userexit_option 1
```

5. Set the path to the High Availability User Exit library, if changed from the default.

Alternatively, set the path to the High Availability User Exit library from the command line configuration:

```
peluconf set -s swnet swnet_ack_userexit_library <path to the HA User Exit library>
```

6. Select the database type (MYSQL or ORACLE).

Alternatively, select the database type from the command line configuration:

```
peluconf set -s swnet swnet_ack_userexit_db_type <database_type>
```

7. Select the database DSN name and user.

Alternatively, select the database DSN name and user:

```
peluconf set -s swnet swnet_ack_userexit_db_data_source_name <ODBC data source name>
```

```
peluconf set -s swnet swnet_ack_userexit_db_user <database user>
```

8. Using the `pelencpass` command, set and store the database password in an encrypted storage. For details about storing an encrypted password, see the documentation for the `pelencpass` command.

9. Set the resulting password files using the command line configuration:

```
peluconf set -s swnet swnet_ack_userexit_db_salt_file <password salt file>
```

```
peluconf set -s swnet swnet_ack_userexit_db_dk_file <password dk file>
```

```
peluconf set -s swnet swnet_ack_userexit_db_password_file <password file>
```

10. Set the path to the DB High Availability User Exit library from the command line configuration, if changed from the default:

```
peluconf set -s swnet swnet_ack_userexit_db_library <path to the DB User Exit Library>
```

11. Restart Gateway for the change in configuration to take effect.

High Availability User Exit load and initialization

When the High Availability feature is enabled, SWIFTNet client and server processes load the High Availability and DB user exit libraries at Gateway startup according to the preconfigured settings.

If errors occur during High Availability or DB user exit libraries loading or initialization, the SWIFTNet processes performing the operation are terminated. Correct the errors reported in the Gateway log.

An incorrectly configured High Availability feature can prevent the SWIFTNet module from functioning.

PeSIT Duplicate Checking in the context of SWIFTNet High Availability

[About PeSIT High Availability](#)

[Limitations](#)

[Prerequisites](#)

[Enabling the PeSIT Duplicate Checking](#)

About PeSIT High Availability

The PeSIT Duplicate Checking in the context of SWIFTNet High Availability feature is designed to detect duplicate PeSIT incoming transfers and to reject such transfers.

The duplicate check function is performed using the following PeSIT transfer parameters:

- PI3 (originator)
- PI4 (destination)
- PI12 (logical file name)
- PI13 (transfer ID)
- PI51 (file creation date)

The parameters are stored in the High Availability Database when the transfer is completed.

When receiving a new transfer/file CREATE request, Gateway performs a search in the High Availability Database based on the 5 parameters mentioned above. If a match is found, the transfer is considered a duplicate and the transfer is aborted. In such a case, Gateway responds with a 204 "File already exists" message.

If Gateway cannot find a match, the transfer proceeds normally.

Note PeSIT duplicate checking is performed only on incoming transfers.

Duplicate Check for incoming PeSIT transfers

If the Duplicate Check function is active, and a duplicate PeSIT transfer is detected, Gateway responds with a 204 "File already exists" code and the transfer is aborted.

The client software sending the transfer should not attempt to resend it.

Limitations

- PeSIT E HS

Furthermore, the following functions, though not disabled, may not work as expected in a SWIFTNet High Availability configuration:

Setting of synchronization points

The data sender can initiate sequentially numbered synchronization points during the transfer. The data receiver can acknowledge the synchronization points to confirm that the data transferred up to that point has been correctly received and saved. This mechanism is used to restart a transfer after an incident.

When you establish the logical connection between partner Sites, the two Sites can negotiate the following:

- Synchronization acknowledgment window
- Interval between synchronization points
- Whether resynchronization is allowed

Restarting interrupted transfers

Transfer restart enables you to restart a transfer interrupted before its completion. The interruption closes the file and disables the protocol and network connections. It is always the Initiator who initiates the restart and the Receiver who decides from which synchronization point the transfer is to be restarted.

Dynamic synchronization

Dynamic synchronization allows you to ask the partner to restart the transfer from a previous synchronization point, if an incident was encountered during data transfer. The file remains open during this operation, which can be initiated by either Initiator or Responder.

Prerequisites

The PeSIT Duplicate Checking in the context of the SWIFTNet High Availability feature uses the same configuration parameters as the SWIFTNet High Availability. To activate the PeSIT Duplicate Checking in the context of SWIFTNet High Availability, the SWIFTNet High Availability feature must be configured (See [High Availability User Exit Configuration](#))

Note The PeSIT Duplicate Checking in the context of the SWIFTNet High Availability feature needs a valid license key in order to function.

Enabling PeSIT Duplicate Checking in the context of SWIFTNet High Availability without a proper license key will result in Gateway PeSIT processes being unable to start.

The license key is the same as for SWIFTNet High Availability.

Enabling the PeSIT Duplicate Checking

The PeSIT Duplicate Checking can be individually enabled for PeSIT D and PeSIT E protocols, using the command line configuration tool, as follows:

1. Enable the PeSIT Duplicate Checking in the context of SWIFTNet High Availability feature from the command line configuration:

```
peluconf set -s ft_phsd duplicate_userexit_option 1
```

or

```
peluconf set -s ft_phse duplicate_userexit_option 1
```

2. Restart Gateway for the change in configuration to take effect.

Except for the activation parameters, the SWIFTNet High Availability feature configuration is shared between PeSIT and SWIFTNet, and performed in the SWIFTNet module (See [High Availability User Exit Configuration](#)). Ensure all the parameters are properly configured before activating the SWIFTNet High Availability feature.

Note While the PeSIT Duplicate Checking in the context of SWIFTNet High Availability was designed to function together with SWIFTNet High Availability as a combined solution, it is not necessary to have SWIFTNet High Availability activated in order to use PeSIT Duplicate Checking in the context of High Availability, but it is necessary to have the parameters for the connectivity with database set in the SWIFTNet configuration section.

SWIFTNet High Availability with External Correlation

Prerequisites

The prerequisites for activating the SWIFTNet *High Availability with External Correlation* feature are:

- Each Gateway node installation from a High Availability solution must connect to SWIFT through its own SAG.
- When installing Gateway, the High Availability option must be used.
- Each Gateway from the High Availability environment must be assigned a unique node identifier (See About Gateway Node ID).

Note The SWIFTNet *High Availability with External Correlation* feature requires a valid license key in order to function. Enabling SWIFTNet High Availability with External Correlation without a proper license key will result in Gateway SWIFTNet processes being **unable to start**.

About Gateway Node IDs

In environments where multiple Gateways are installed -for example connected to the same back-end application - it is desirable to have unique transfer identifiers for each Gateway installation. This functionality can be enabled by assigning each Gateway a unique node identifier. By default, this functionality is disabled. A node identifier can have values from 0 to 15, for a total of 16 nodes. If set, the transfer identifier allocation changes as follows:

- Transfer identifiers are no longer allocated contiguously. Instead, a new transfer identifier will be the next number multiple of 16 + the node identifier (thus incremented by 16). This ensures that each Gateway allocates its own subset of transfer identifiers.
- There is no change for transfers already in the mailbox; therefore it is possible to have old transfers not conforming with the above rule. However, activating the node identifier feature should be done on a new installation.

Setting a Gateway node identifier

The Gateway node identifier can be set using the `peluconf` command:

```
peluconf set -s monitor node_id <node identifier>
```

where the node identifier ranges from 0 to 15.

Note There is no equivalent command to unset the node identifier. This implies that **once activated, the node identifier feature cannot be disabled**. A Gateway node can be set to a different node identifier.
Ensure that each Gateway is assigned a unique node identifier.

Enabling the SWIFTNet High Availability with External Correlation feature

You must configure some global parameters before using the SWIFTNet High Availability with External Correlation feature. This can be partially done using the Graphical User Interface, and/or the command line configuration.

Use the following steps to enable SWIFTNet High Availability with External Correlation. If not properly defined, the Gateway SWIFTNet client and server processes will be unable to start.

1. Set the `node_id` parameter to a value between 0 and 15.
2. In the left pane of the main window in the user interface, right-click the Gateway server to enable High Availability and select **Configure**. The Configuration window appears.
3. In the left pane of the Configuration window, go to **Connectivity > Axway connectivity > SWIFTNet** and select the **HA** tab.
4. From the **High Availability Exit** drop-down list, select **Use external correlation** to enable the **High Availability with External Correlation** feature.
5. Alternatively, enable the High Availability with External Correlation feature from the command line configuration:

```
peluconf set -s swnet swnet_ack_userexit_option 2
```
6. Restart Gateway for the change in configuration to take effect

Axway Gateway: Connectors

JMS

About Java Message Service (JMS)

The Java Message Service™ (JMS) is a Java API for exchanging messages between two or more clients through a Message-Oriented Middleware (MOM). The JMS API defines a common set of interfaces and associated semantics that allow programs written in the Java programming language to communicate with any JMS-compatible MOM implementation. The JMS API is part of the Java 2 Platform, Enterprise Edition (J2EE). It is provided in the Java package `javax.jms`.

JMS messages

JMS messages are the objects that communicate information between JMS clients. The messages are asynchronous requests, reports, or events that are consumed by enterprise applications, not humans. They contain precisely-formatted data that describes specific business actions. Through the exchange of these messages, each application tracks the progress of the enterprise.

JMS message structure

A JMS message has three main parts:

- **Message header** (required): Contains operational settings to identify and route messages
- **Message properties** (optional): Contains additional properties to support compatibility with other providers or users. These properties can be used to create custom fields or filters.
- **Message body** (optional): Allows users to create different types of message (text message, bytes message, and so on).

The message interface is extremely flexible and provides many ways to customize the contents of a message.

JMS messaging models

The JMS API supports two types of messaging model:

- **Queuing model**
- **Topic model**

Queuing model (point-to-point)

In the *queuing model*, a producer posts messages to a specific message queue and a consumer reads messages from the queue(s) established to hold their messages. Here, the producer knows the destination of the message and posts the message directly to the consumer's queue.

In this model:

- Only one consumer receives the message
- The producer does not have to be running when the receiver consumes the message, nor does the receiver need to be running when the message is sent
- Queues retain all messages sent to them until the messages are consumed or until the messages expire
- Every message successfully processed is acknowledged by the receiver

Topic model (publish and subscribe)

The *topic model* supports publishing messages to a particular message topic. Zero or more subscribers may register interest in receiving messages on a particular message topic. In this model, neither the publisher nor the subscriber know about each other.

In this model:

- Multiple consumers can receive the message
- There is a timing dependency between publishers and subscribers. The publisher has to create a subscription in order for clients to be able to subscribe. The subscriber has to remain continuously active to receive messages, unless it has established a durable subscription. In that case, messages published while the subscriber is not connected will be redistributed whenever it reconnects.

JMS API Architecture

JMS consists of several elements:

- **JMS provider:** An implementation of the JMS interface for a Message-Oriented Middleware (MOM). Providers are implemented as either a Java JMS implementation or an adapter to a non-Java MOM.
- **JMS client:** A Java-based application or object that produces and/or consumes messages.
- **JMS producer:** A JMS client that creates and sends messages.
- **JMS consumer:** A JMS client that receives messages.
- **JMS message:** An object that contains the data being transferred between JMS clients.
- **JMS queue:** A staging area that contains messages that have been sent and are waiting to be read. The messages are delivered in the order sent. A message is removed from the queue once it has been read.
- **JMS topic:** A distribution mechanism for publishing messages that are delivered to multiple subscribers.

Java Naming and Directory Interface (JNDI)

The JMS API, like all other Java Enterprise APIs, uses the *Java Naming and Directory Interface* (JNDI) API for administration. The JMS API defines *ConnectionFactory*s and *Destinations* as administered objects that are configured and placed in a JNDI naming context. JMS clients then look up and use these preconfigured objects. This ensures that JMS applications are easy to deploy and administer.

JMS Providers

In order to use JMS, you must have a JMS provider that can manage the sessions and queues.

Examples of JMS providers:

- Axway Messaging (XMS)
- TIBCO EMS
- IBM WebSphere MQ (formerly MQSeries)
- Oracle Enterprise Messaging Services (OEMS)
- Sun Java System Message Queue
- Open Message Queue (Sun Microsystems)

[JMS connector](#)

[Configuring the Axway Gateway/JMS environment](#)

[Working with Axway Gateway/JMS](#)

More information

Refer to the Oracle website: <http://www.oracle.com/technetwork/java/jms/index.html>

Axway Gateway: Connectors

JMS connector

[JMS and Gateway](#)

[Required Gateway objects for JMS](#)

JMS and Gateway

The Gateway JMS connector enables Gateway to communicate with other applications using JMS messages. This enables you to interface with JMS-compatible Message-Oriented Middleware (MOM) such as Axway Messaging or IBM WebSphere MQ.

You can:

- Route messages received by a JMS MOM to Gateway
- Route files received by Gateway to a JMS MOM

The Gateway JMS connector calls the JMS API. You need to have a JMS provider and [configure the connector](#) for this provider.

Supported versions

This version of Gateway supports JMS V1.1 and all earlier versions.

Supported message types

The Gateway JMS connector supports the following JMS message types:

- **TextMessage**: contains a character string
- **BytesMessage**: contains an array of bytes
- **Message**: no data content (contains header and properties only)

Gateway rejects all other JMS message types.

Commit and Rollback

For each message transfer, the JMS connector within Gateway decides at a certain moment in time whether to *commit* (validate) the transfer or to carry out a *rollback* (cancel). Gateway carries out a commit only when all internal operations have been completed successfully. If an error occurs during a transfer, Gateway carries out an automatic rollback.

Overview of JMS message transfer using Gateway

From Gateway to JMS

Gateway can convert any file to JMS message format and send it to a JMS MOM. For example, you can route files received by Gateway (using PeSIT or any other supported protocol) to a JMS MOM.

From JMS to Gateway

Gateway detects the arrival of a JMS message on the message queue in the JMS MOM. If a message arrives, Gateway automatically receives it.

Required Gateway objects for JMS

You need to create the following Gateway objects to be able to exchange and/or route JMS messages.

Remote Site object

Here you define how to communicate with your JMS provider, for example:

- Direction: send and/or receive
- Mode: queue (point-to-point) or topic (publish and subscribe)
- Queue manager/JMS provider
- Initialization properties

You can create several Remote Sites, depending on your requirements.

Transfer Request object

A Transfer Request is the object that describes how a message is sent or received through JMS. Amongst other parameters, it specifies:

- The JMS Remote Site
- The JMS properties
- The payload location (**File component**)

There are various ways to create a Transfer Request, for example:

- Through a request issued by an application using the Gateway Application interface (API, command line)
- By a user using the *New Transfer Request* window of the Gateway GUI
- Through the automated routing service provided by Gateway

Decision Rule object and Model object

Here you specify the criteria that will activate the linked routing Model. Through the routing Model you define the parameters of the outgoing transfer. These parameters can be set with the value of the JMS incoming transfer parameters, such as the JMS properties.

Via the Decision Rule object you define the selection criteria for the routing Model:

- For *incoming* JMS messages:
 - **Decision Rule:** selection criteria, such as property name = property value
 - **Model:** Transfer parameter = property value
- For *outgoing* JMS messages:
 - **Decision Rule:** Transfer parameters (for example, PeSIT protocol, application, file name)
 - **Model:** parameter = property value (incoming transfer parameters)

Implementation Note

The `DateFormat` class is instantiated locally in each thread. Thus it is used as described in the Java documentation: "Date formats are not synchronized. It is recommended to create separate format instances for each thread." Each thread of Jms Connector has a specific and unique `DateFormat` instance.

Gateway uses:

- `JmsConnector` which is threadable.
- `JmsConnector` which instantiates a `LogWriter`.
- `LogWriter` which instantiates a `DateFormat`.

From the `MessageFormat` class, Gateway uses `MessageFormat.format()`, which for Java 1.6 is thread safe:

`http://docs.oracle.com/javase/6/docs/api/java/text/MessageFormat.html#format(java.lang.String,%20java.lang.Object...)`.

[About JMS](#)

[Configuring the Axway Gateway/JMS environment](#)

[Working with Axway Gateway/JMS](#)

Axway Gateway: Connectors

Configuring the Axway Gateway/JMS environment

[Prerequisites](#)

[Configuring the JMS connector](#)

[JMS static configuration](#)

[JMS connector logging](#)

[Required objects](#)

[Creating objects for JMS transfers](#)

Prerequisites

Before you can use Gateway to send and receive JMS messages:

- Check that Java Runtime Environment 1.4.2 or higher is installed on your machine.
- You need to have a JMS provider.
- You need access to the JMS provider from the machine running Gateway.
- The `jar` files for the JMS provider must be installed on the machine running Gateway.
- At least one queue must be defined on the JMS provider machine.
- Ensure that the MOM user has the necessary access rights (user name + password).

Configuring the JMS connector

To configure Gateway for use with JMS, set up the JMS connector in the Gateway [configuration menu](#):

- In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
- Select **Connectivity > Axway Connectivity > JMS**.

- Click **Activate** to enable the connector.
- Click **OK**.

JMS static configuration

The JMS java connector process is started by Gateway processes using a java command line. The configuration of the start JMS command line is done through the `[monitor]jms_command` parameter (default value is provided with Gateway installation):

```
[monitor]jms_command='java -Xms64M -Xmx128M -DcallbackAddress=$h_jms_
listen -Dtrace_dir_path=$p_tmp_dir -DjmsInLogProp=$p_runtime_
dir/jms/logJMSIN.properties -DjmsOutLogProp=$p_runtime_
dir/jms/logJMSOUT.properties -classpath $p_runtime_
dir/jms/jmsctor.jar:$p_runtime_dir/jms/javax.jms.jar:$p_runtime_
dir/jms/xerces-1.4.4.jar:$p_runtime_dir/jms/log4j-core-2.10.0.jar:$p_
runtime_dir/jms/log4j-api-2.10.0.jar com.axway.gateway.jms.Gtw2Jms'
```

JMS connector logging

The JMS connector process logs information using the log4j2 apache library. The logging information produced is configured (levels, rolling, policies etc) using two properties files:

- one for the inbound connector (`logJMSIN.properties`)
- one for the outbound connector (`logJMSOUT.properties`).

Two sample properties files are provided with the Gateway installation, but you can change them at will using the syntax described at <https://logging.apache.org/log4j/2.x/manual/configuration.html>.

You can also change the name and location of the properties files; to do so, update the `[monitor]jms_command` parameter.

Required objects

To carry out a JMS transfer to or from a remote business partner, you require as a minimum the following Gateway objects:

- Remote Site object

You may also need to create other objects for routing. For more information, refer to [Working with the JMS connector](#).

Creating objects for JMS transfers

This section describes how to create the following objects for JMS transfers:

- Remote Site objects

Creating a Remote Site object for JMS transfers

In the Remote Site object you specify the JMS queue initialization properties and the connection parameters to your provider. You can also specify the transfer direction (Initiator/Sender or Responder/Receiver).

To create a Remote Site object:

1. In the left pane of the GUI main window, click  to expand the nodes:
Partner Management > Sites
2. Right-click the **Remote Site** sub-node, then select **New** to display the *New Remote Site* window.
3. Complete the fields on the [General](#) tab. In the **Protocol** field, make sure you select **JMS**.
4. Complete the fields on the [JMS](#) tab.
5. Click **OK**.

Gateway creates a new Remote Site object with your specified characteristics.

Once the Remote Site has been configured, the connection between Gateway and the JMS provider is made automatically.

Now that you have configured the JMS environment in Gateway, you are ready to [work with JMS](#).

Using command line

Alternatively, you can create a Remote Site object using the [peladm create site](#) online command.

[About JMS](#)

[JMS connector](#)

[Working with Axway Gateway/JMS](#)

Axway Gateway: Connectors

Working with Axway Gateway/JMS

[Working with JMS on your Gateway](#)

[Submitting a JMS Transfer Request](#)

[Receiving JMS messages](#)

[Routing](#)

[After the transfer](#)

Working with JMS on your Gateway

Before transferring JMS messages you need to [configure Gateway and create various objects](#).

When you have completed these tasks, you can:

- submit a Transfer Request to transfer a JMS message, optionally requesting a receipt
- receive JMS messages from a JMS MOM

Gateway also enables you to route messages automatically to other users.

Message properties

A JMS message can contain *properties*. Often, the message already contains certain properties. You can also add your own properties.

Gateway can test message properties to decide what action to take for a given message, for example for routing.

Submitting a JMS Transfer Request

Transferring a message (from Gateway to JMS)

To execute a file transfer using Gateway, you create and submit a Transfer Request.

You can enter message properties in the Transfer Request object. Gateway adds these properties to the message.

Model objects

Optionally, you can also create one or more Model objects. Models are useful when you need to repeatedly reuse the same information to accomplish a series of tasks. You also use the Model object for [Routing](#).

You can enter message properties in the Model object. Gateway adds these properties to the message.

To help you complete the properties in the Model, first create a Property List.

Simple transfer example using the GUI

The following example illustrates how to send a file. For full details, refer to the relevant sections of this documentation.

1. [Create a New Transfer Request](#).

2. In the [General](#) tab, enter the following parameters:
 - **Type** = TRANS
 - **Mode** = Initiator
 - **Direction** = Send
 - **Destination alias** = <Name of Remote Site that you have already created>
3. In the [Attributes](#) tab, enter the following parameters:
 - **File component** = <name of file to send>
4. In the [JMS](#) tab, add your required Transfer properties.
5. Click OK.

Gateway sends the file to the client defined in the Remote Site. The message arrives in the queue specified by the Remote Site.

In the [Mailbox](#), you can view any properties that were added to the message.

Simple transfer example using an online command

Alternatively, you can use the `peltrans` online command to send a JMS message. The following example transfers the file `file.txt` via a Remote Site named `jms_out` which uses the JMS protocol in the OUT direction:

```
peltrans -dest_alias jms_out
        -appli DEFAULT_A
        -file_component $HOME/file.txt
        -property "PropName1=PropValue1"
        -property "PropName2=PropValue2"
```

Receiving JMS messages

Gateway listens to the JMS queue and automatically transfers incoming JMS messages. (You do not need to create a Transfer Request object.)

You do not need to create a Model object if only one or very few messages are being received and the messages do not require routing. In this case, messages arrive at the local machine. The files are written to the incoming directory specified in the Remote Site (**Msg In directory**).

Transfer attributes

You can view transfer details in the [Mailbox](#). Here you can see message properties as well as the path and name of the file.

The file ID is unique - it is derived from the message ID. You can see header fields and properties (for example, type of message and time stamp) as well as any user-defined properties.

Renaming Incoming JMS files

Incoming JMS files can be automatically renamed, if requested, through an application object selected using a property of the JMS message. To activate the renaming for incoming JMS files, you must update the configuration options `jms_rename_files` and `jms_inbound_appli_property` in the command line as follows:

- `peluconf set -s monitor jms_rename_files 1`

The default value is 0, which means no file renaming.

- `peluconf set -s monitor jms_inbound_appli_property %application_name%`

Sets the incoming message JMS property of string type that defines which application to use for file renaming.

The following transfer properties are accessible using symbolic variables for file renaming/moving:

- `x_local_ident`
- `x_date_begin`
- `x_date_end`
- `x_originator`
- `x_destination`

JMS properties from the incoming message can also be used. To access a JMS property, the syntax `&(x_property_[JMS_PROPERTY_NAME])` should be used.

Routing received messages

For more realistic situations, when many messages are being received, you can test for message properties in a Decision Rule. Gateway examines the properties to determine what action to take with the message.

For example, you can route the message to another JMS client or transform the message in order to route it to a client using any other protocol supported by Gateway, such as EDIINT AS2.

Routing

In Gateway, you automate routing by creating and managing Decision Rule objects. Via the Decision Rule, you define a set of conditions to match against the current transfer. For example:

- Protocol used for the current transfer = JMS
- Destination = Site A
- Destination is not the same as the originating Site
- and so on...

When all the Decision Rule conditions match those of the current transfer, Gateway applies the execution type specified for this Decision Rule to the transfer.

Example: to redirect an incoming transfer to a remote site, create a Decision Rule that applies a Model to the incoming transfer type. When Gateway detects an incoming transfer of the specified type, it applies the Model, which in turn deposits a new Transfer Request towards the desired site.

You can restrict routing to a limited set of transfers. For example, you could restrict the routing to a certain Site to transfers with the following characteristics:

- Transfer type: TRANS
- Transfer direction: IN
- Transfer status: ENDED or ACKED

After the transfer

To check that transfers have been made successfully, or to help you analyze problems, you can view the Mailbox and the Log file.

Using the Mailbox

For each JMS transfer you make using Gateway, Gateway records a transfer record in the Mailbox.

To view the Mailbox:

1. In the left pane of the GUI main window, click  to expand the **Transfer Management** node.
2. Click the **Transfers** node.

Gateway displays all currently stored Mailbox records.

3. To obtain additional information about a Transfer Request, double-click the row representing the Transfer Request record.

Using the Log file

By default, Gateway generates traces of transfer activities that it records to the log file.

[About JMS](#)

[JMS connector](#)

[About Models](#)

[Configuring the Gateway/JMS environment](#)

[Working with File Transfer Requests](#)

[Working with Property Lists](#)

[Working with Rule Tables and Decision Rules](#)

[Viewing and managing Mailbox contents](#)

[Working with Logs](#)

X.400

[X.400](#)

[Addressing](#)

[X.400 components](#)

[X.400 protocols](#)

This topic introduces the X.400 protocol.

About X.400

X.400 is a messaging standard, based on a set of ITU-T Recommendations, each in the 400-number range. These Recommendations define standards for Data Communication Networks for Message Handling Systems (MHS).

X.400 is an alternative to the more common SMTP email protocol. It is often used for the transmission of EDI messages, for example in the European food trade.

Addressing

Originators and recipients of an X.400 message are identified by their X.400 Originator/Recipient addresses (O/R addresses). X.400 specifies a number of possible address characteristics that do not exist in SMTP.

An X.400 address consists of several parts. In the short form (string form) each part has a tag, for example:

```
"C=SE;A=400NET;P=FRONTEC;O=AMT;OU1=X420UA;CN=EDI;"
```

X.400 address elements

The following table lists the most-commonly-used X.400 address elements.

Tag	Meaning	Max length
C	Country name	3 characters
A	ADMD, Administrative Management Domain name. Usually a public mail service provider.	16 characters
P	PRMD, Private Management Domain name	16 characters

O	Organization name	64 characters
OU1-4	Organizational Unit names	34 characters
S	Surname	40 characters
G	Given name	16 characters
I	Initials	5 characters
CN	Common name	64 characters
DDAT	Domain-defined Attribute Type, specific to a messaging domain and used to pass non-standard information.	8 characters
DDAV	Domain-defined Attribute Value. Always set together with, and following, the DDAT.	
Q	General Qualifier	3 characters

Example

The SMTP email address:

```
joe.smith@dev.axway.fr
```

might look like this in an X.400 email message:

```
G=joe; S=smith; O=axway; OU=dev; PRMD=sopranet; ADMD=sopranet; C=fr
```

X.400 components

Message handling is a distributed information processing task that integrates two related sub-tasks: *message transfer* and *message storage*.

The main components used in X.400 communication are:

- Message Transfer Agents (MTA)
- Message Stores (MS)
- User Agents (UA)

X.400 protocols

The protocols defined by the standard are:

Protocol	Usage	Description

P1	MTA - MTA	End-to-end message transfer service.
P3	MTA - UA	Communication between UAs and MTAs.
P7	UA - MS	An increasingly widely used protocol for communication between the UA and the MS. Typical P7 operations: • Bind to Message Store • UnBind from Message Store • List Messages held in Message Store • Fetch Messages/Notifications/Reports • Submit Messages/Notifications/Reports • Delete message from Message Store
P2/P22	Interpersonal messaging (IPM)	Defines the format of a structured memo-like document having two main components: the heading and the body part(s). The heading specifies things like the originator and recipient of the message, copy and blind copy recipients of the message, whether the originator expects to receive a notification for the message and in what circumstances etc. The body part may be one of many forms of data - those specified by the standard include Ia5 text, G3Fax, Images, Audio data etc. Messages may also be encapsulated within messages, providing a clean and structured forwarding mechanism.

[X.400 connector](#)

[Configuring Gateway for X.400](#)

[Managing X.400 partners](#)

[Working with X.400](#)

[X.400 O/R address](#)

[X.400 log messages](#)

More information

For details of ITU-T Recommendations, go to:

<http://www.itu.int/ITU-T/dbase/index.html>

X.400 connector

[Introduction](#)

[This version](#)

[X.400 and Gateway](#)

[Required Gateway objects for X.400](#)

This topic introduces the X.400 connector in Gateway.

Introduction

The X.400 connector enables Gateway to function as an X.400 host. Gateway can connect simultaneously to any number of Message Transfer Agents (MTA) and Message Stores (MS).

You can use the X.420 protocol to send and receive EDI messages. The Gateway X.400 connector is a communication server that uses this protocol.

The user has access to a system that handles sending, receiving, and acknowledgment of messages.

This version

This version of Gateway supports X.420 on TCP/IP.

This version has been designed to work with Axway Integrator.

Limitations

Routing and Sentinel tracking do not include X.400-specific fields.

X.400 and Gateway

The Gateway X.400 connector enables you to perform the following roles:

- Transport and route X.400 messages between partners and platforms
- Manage X.400 message transfers

You can also use Integrator in conjunction with Gateway for advanced message processing.

Required Gateway objects for X.400

You need to create the following Gateway objects to be able to exchange and/or route X.400 messages.

Remote Site object

In the Remote Site object you specify the remote MTA and MS retrieve methods. You can create as many Remote Sites as required.

Transfer Request object

In the Transfer Request object you describe how a message is sent or received using the X.420 protocol.

[About X.400](#)

[Configuring Gateway for X.400](#)

[Managing X.400 partners](#)

[Working with X.400](#)

[X.400 log messages](#)

Axway Gateway: Connectors

Configuring Gateway for X.400

[Prerequisites](#)

[Configuring Gateway for X.400](#)

[Configuring Axway PassPort](#)

[Creating objects for X.400 transfers](#)

[Configuring partners](#)

Prerequisites

In addition to installing Gateway (and any available patches) you must install PassPort PM V3.4.4 (or later) and the latest patch(es).

Configuring Gateway for X.400

To configure Gateway for use with X.400, set up the X.400 connector in the Gateway [configuration menu](#):

1. In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.

2. Select **Connectivity > Axway Connectivity > X.400**.
3. Complete the [X.400 parameters](#).
4. Click **Activate** to enable the connector.
5. Click **OK**.

Configuring Axway PassPort

To configure Gateway for use with PassPort, set up the PassPort connector in the Gateway [configuration menu](#):

1. In the left pane of the Gateway GUI, right-click the Gateway server that you want to configure and select **Configure**.
2. Select **Connectivity > Axway Connectivity > PassPort**.
3. Complete the [PassPort parameters](#).
4. Click **OK**.

Creating objects for X.400 transfers

This section describes how to create Remote Site objects for X.400 transfers.

Creating a Remote Site object for X.400 transfers

In the Remote Site object you specify the remote MTA and MS retrieve methods.

To create a Remote Site object:

1. In the left pane of the GUI main window, click  to expand the nodes:

Partner Management > Sites

2. Right-click the **Remote Site** sub-node, then select **New** to display the *New Remote Site* window.
3. Complete the fields on the [General](#) tab. In the **Protocol** field, make sure you select **X.400**.
4. Complete the fields on the [Options](#) tab.
5. Complete the fields on the [X.400](#) tab.
6. Click **OK**.

Gateway creates a new Remote Site object with the specified characteristics.
You can create as many Remote Sites as required.

Using command line

Alternatively, you can create a Remote Site object using the [peladm create site](#) online command.

Configuring partners

Configure X.420 partners using PassPort PM. Refer to [Managing X.400 partners](#).

[About X.400](#)

[X.400 connector](#)

[Managing X.400 partners](#)

[Working with X.400](#)

[X.400 log messages](#)

Axway Gateway: Connectors

Managing X.400 partners

[About X.400 partner management](#)

[Prerequisites](#)

[Managing X.420 partners in PassPort PM](#)

About X.400 partner management

For the X.400 protocol, Gateway makes use of the partner management function provided by Axway PassPort. You use PassPort PM to view, create, modify or delete X.400 users, or partners. In this version of Gateway, X.420 partners are supported.

This topic describes the X.400-specific parameters in PassPort PM.

Prerequisites

In addition to installing Gateway (and any available patches) you must install PassPort PM V3.4.4 (or later) and the latest patch(es).

Managing X.420 partners in PassPort PM

1. Log in to PassPort PM.
2. Click the **PassPort Administration** tab.
3. In the left menu, select **Profile Management > Communication Protocol > Open Communication Protocol**.
4. On the **General** tab, in **Protocol Type**, select **X420-Users**.

5. Click **Search**.
PassPort PM displays a list of all available X.420 partners.
6. To view or edit partner details, click the eye icon at the right of the line for that partner.
PassPort PM displays general information about the partner.
7. Click the **Protocol** tab to display X.400-specific parameters related to this partner.
8. Edit the parameters as required.
9. Click **OK** to save your changes or **Cancel** to quit without making changes.

X.420 partner parameters

The following table lists X.400-specific parameters.

Name (prefixed by X400_)	Format and value	Category (1)	Remarks
X420:			
USER_ALIAS	PrintableString 1..32		The alias name of the X.420 partner.
UA_NAME	PrintableString 1..16	Local	The name of the UA attached to this user. The name is only used locally in the system: it is not in any of the X.400 protocols. This name is used by the UA servers when they register themselves in the local MTA. The defined UA name value must be the same value as the value defined for the UA processes (x420s) u start option. You can add multiple entries with the same UA_NAME. This makes it possible to correlate multiple O/R addresses to a specific UA, which is the case when multiple local partners have been defined. The UA name value used must contain the substring 420.
MTA_NAME	IA5-string 1..32	Remote	This optional parameter is used to explicitly correlate a remote X.420 partner with an external MTA. When the local MTA gets a message that is addressed to this partner, the message is routed to this MTA. The MTA name must match a remote MTA defined in the MTA Remote Site.
MS_NAME	PrintableString 1..32	Local	The Message Store attached to this user (if MS used).

MESSAGE_STORE_USED	Boolean { True False }	Local	This parameter indicates if this UA entry is connected to a Message Store (True) or to the local MTA (False). If this parameter is set to True, there is an MSP7 Remote Site definition that is correlated to this UA entry. The Remote Site entry contains information about the remote mailbox.
SUBSET_ADDR_USED	Boolean { True False }	Local	Match OR_ADDRESS exactly for incoming message or not. If this parameter is set to True, match UA_OR_ADDR address attributes exactly. In other words, messages are put in the UA in-queue only if the recipient address contains address attributes that exactly match this address. If set to False, match at least these UA_OR_ADDR: address attributes. In other words messages are put in the UA in-queue if the recipient address contains at least these attributes.
DELIVERY_TIMEOUT	Decimal digits	Local	Timeout in minutes. Default = 1440 (24 hours). The parameter defines the number of minutes that a delivered message can stay in the UA in-queue. If the UA does not read the message and accepts the responsibility for it, the MTA will remove the message from the in-queue and send a non-delivery notification to the message originator. If the value is set to 0, it means that the UA has a retrieval queue. This implies that the responsibility for the message is imposed on the UA as soon as the message is placed in its in-queue, and the MTA does then not supervise this queue.
OR_ADDRESS:			The OR_ADDRESS is mandatory. It contains the X.400 O/R address information of the X.420 partner.
CN	PrintableString 2..2		Country Name identifies the country.
ADMD	PrintableString 1..16		Administration Domain Name. Identifies an ADMD relative to the country specified by the Country Name. An ADMD provides a public messaging service within a single country. Several ADMDs may exist within a country.

PRMD	PrintableString 1..16		Private Domain Name. Identifies a PRMD relative to the ADMD or the Country. A PRMD is a messaging service within an organization. The ADMD operator must uniquely assign the Name of the Private Domain if the PRMD is connected to an ADMD service.
ORG_NAME	PrintableString 1..64		Organization name address attribute.
ORG_UNIT_NAME_1	PrintableString 1..32		Organization unit name address attributes
ORG_UNIT_NAME_2	PrintableString 1..32		
ORG_UNIT_NAME_3	PrintableString 1..32		
ORG_UNIT_NAME_4	PrintableString 1..32		
COMMON_NAME	PrintableString 1..64		An RFC1685 encoding format of the O/R address.
FREE_FORM_NAME	PrintableString 1..64		Free-form name address attribute of the O/R descriptor.
TELEPHONE_NUMBER	PrintableString 1..32		Telephone number address attribute of the O/R descriptor.
DOMAIN_DEF_ATTR_1:			Optional domain-defined attribute.
DDA_TYPE_1	PrintableString 1..8		Domain-defined attribute type address attribute.
DDA_VALUE_1	PrintableString 1..128		Domain-defined attribute value address attribute.
DOMAIN_DEF_ATTR_2:			Optional domain-defined attribute.
DDA_TYPE_2	PrintableString 1..8		Domain-defined attribute type address attribute.
DDA_VALUE_2	PrintableString 1..128		Domain-defined attribute value address attribute.
DOMAIN_DEF_ATTR_3:			Optional domain-defined attribute.
DDA_TYPE_3	PrintableString 1..8		Domain-defined attribute type address attribute.
DDA_VALUE_3	PrintableString 1..128		Domain-defined attribute value address attribute.
DOMAIN_DEF_ATTR_4:			Optional domain-defined attribute.
DDA_TYPE_4	PrintableString 1..8		Domain-defined attribute type address attribute.

DDA_VALUE_4	PrintableString 1..128		Domain-defined attribute value address attribute.
PERSONAL_NAME:			Optional subgroup within OR_ADDRESS containing the personal address attributes: SURNAME, GIVENNAME, INITIALS and GQ.
SURNAME	PrintableString 1..40		Mandatory if PERSONAL_NAME is used. Surname address attribute.
GIVENNAME	PrintableString 1..16		Given name address attribute.
INITIALS	PrintableString 1..5		Initials address attribute.
GQ	PrintableString 1..3		Generation Qualifier address attribute.

Notes:

(1) All parameters apply to both categories (Local and Remote) unless otherwise stated.

(2) Mandatory parameters are indicated by an asterisk (*) in the PassPort PM GUI.

*String definitions***PrintableString:**

a-z

A-Z

0-9

Space

() + , . / - : = ?

IA5-String:

a-z

A-Z

0-9

Space

] ! " \$ % & ' () * + , . / : < = > ? @ [_ ` { | } ~ ^ # ; \ -

[About X.400](#)

[X.400 connector](#)

[Configuring Gateway for X.400](#)

[Working with X.400](#)

[X.400 O/R Address](#)

More information

For full details about partner management with PassPort PM, refer to the PassPort PM documentation.

Axway Gateway: Connectors

Working with X.400

[Working with X.400 on Gateway](#)

[Submitting an X.400 Transfer Request](#)

[After the transfer](#)

This topic describes the parameters in Gateway that enable X.420 transfers.

Working with X.400 on Gateway

Before transferring messages using X.400 you need to [configure Gateway and create various objects](#).

After completing these tasks, you can:

- Send X.400 messages to remote partners
- Receive X.400 messages from remote partners

Submitting an X.400 Transfer Request

You can use any of the following methods to create an X.400 (X.420) Transfer Request:

- [Create and submit a Transfer Request](#) using the Gateway GUI.
- Use the `peltrans` online command to create a Transfer Request. Refer to [Additional Transfer Request parameters for X.400](#) for a list of `peltrans` parameters specifically for X.420 transfers.
- Use GikAPI commands. Refer to [C API Transfer and Selection parameters](#) for specific X.420 parameters.
- Use Axway Integrator to initiate X.420 transfers (not described in this documentation)

After the transfer

To check that transfers have been made successfully, or to help you analyze problems, you can view the Mailbox and the Log file.

Using the Mailbox

For each X.400 transfer you make using Gateway, Gateway records a transfer record in the Mailbox.

Using the Log file

By default, Gateway generates traces of network activities that it records to the log file.

[Working with File Transfer Requests](#)

[Working with Transfers \(command line\)](#)

[Viewing and managing Mailbox contents](#)

[Working with Logs](#)

[X.400 log messages](#)

[About X.400](#)

[X.400 connector](#)

[Configuring Gateway for X.400](#)

[Managing X.400 partners](#)

[C API Transfer and Selection parameters](#)

TARGET2

About TARGET2

TARGET2 is the European Real Time Gross Settlement (RTGS) system which, since November 2007, replaces the TARGET system. TARGET2 is based on the Single Shared Platform (SSP), developed by Banca d'Italia, Banque de France, and Deutsche Bundesbank. These three banks operate the system on behalf of the Eurosystem. TARGET2 participants access the SSP using the SWIFT messaging services: FIN, FIN Copy, InterAct, FileAct and Browse.

About QT2

Query Target2 (QT2) is the mechanism that enables a client application to access a TARGET2 platform. It enables the client application to handle the XML messages specifically adapted to TARGET2 exchanges.

SWIFT documentation

Before you configure Axway Financial Exchange to handle TARGET2 exchanges you should be familiar with the SWIFT TARGET2 documentation. We specifically recommend the following SWIFT documents that you can obtain from www.swift.com:

- General documentation:
 - SWIFTNet Service Description
 - SWIFTNet Connectivity Packs
 - SWIFTNet Resilience Guide
 - SWIFTNet Naming and Addressing Summary
 - SWIFT Network Access Control Guidelines
 - SWIFTNet Certificate Administration Guide
- TARGET2 documentation:
 - SWIFTNet for TARGET2 Getting Started
 - TARGET2 User Detailed Functional Specification (UDFS) Book 1
 - TARGET2 User Detailed Functional Specification (UDFS) Book 2
 - TARGET2 User Detailed Functional Specification (UDFS) Book 4_tcm14-86388 - Version 2.4 - 6 June 2008
 - TARGET2 User Detailed Functional Specification (UDFS) Book 4 - Version 3.0 - 13 March 2009

TARGET2

About TARGET2

TARGET2 is the European Real Time Gross Settlement (RTGS) system which, since November 2007, replaces the TARGET system. TARGET2 is based on the Single Shared Platform (SSP), developed by Banca d'Italia, Banque de France, and Deutsche Bundesbank. These three banks operate the system on behalf of the Eurosystem. TARGET2 participants access the SSP using the SWIFT messaging services: FIN, FIN Copy, InterAct, FileAct and Browse.

About QT2

Query Target2 (QT2) is the mechanism that enables a client application to access a TARGET2 platform. It enables the client application to handle the XML messages specifically adapted to TARGET2 exchanges.

SWIFT documentation

Before you configure Axway Financial Exchange to handle TARGET2 exchanges you should be familiar with the SWIFT TARGET2 documentation. We specifically recommend the following SWIFT documents that you can obtain from www.swift.com:

- General documentation:
 - SWIFTNet Service Description
 - SWIFTNet Connectivity Packs
 - SWIFTNet Resilience Guide
 - SWIFTNet Naming and Addressing Summary
 - SWIFT Network Access Control Guidelines
 - SWIFTNet Certificate Administration Guide
- TARGET2 documentation:
 - SWIFTNet for TARGET2 Getting Started
 - TARGET2 User Detailed Functional Specification (UDFS) Book 1
 - TARGET2 User Detailed Functional Specification (UDFS) Book 2
 - TARGET2 User Detailed Functional Specification (UDFS) Book 4_tcm14-86388 - Version 2.4 - 6 June 2008
 - TARGET2 User Detailed Functional Specification (UDFS) Book 4 - Version 3.0 - 13 March 2009

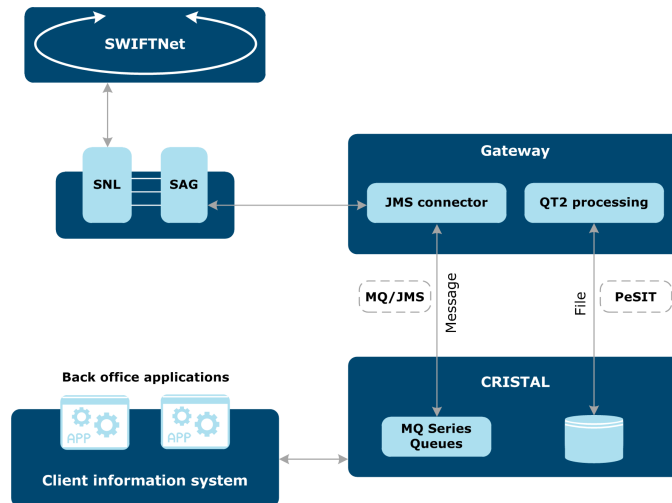
TARGET2 exchange architecture

Required components

To use Axway Financial Exchange to perform TARGET2 exchanges, you require:

- Axway Gateway
- CRISTAL
- A connection to SWIFTNet via SAG/SNL

The following figure illustrates the TARGET2 exchange architecture:



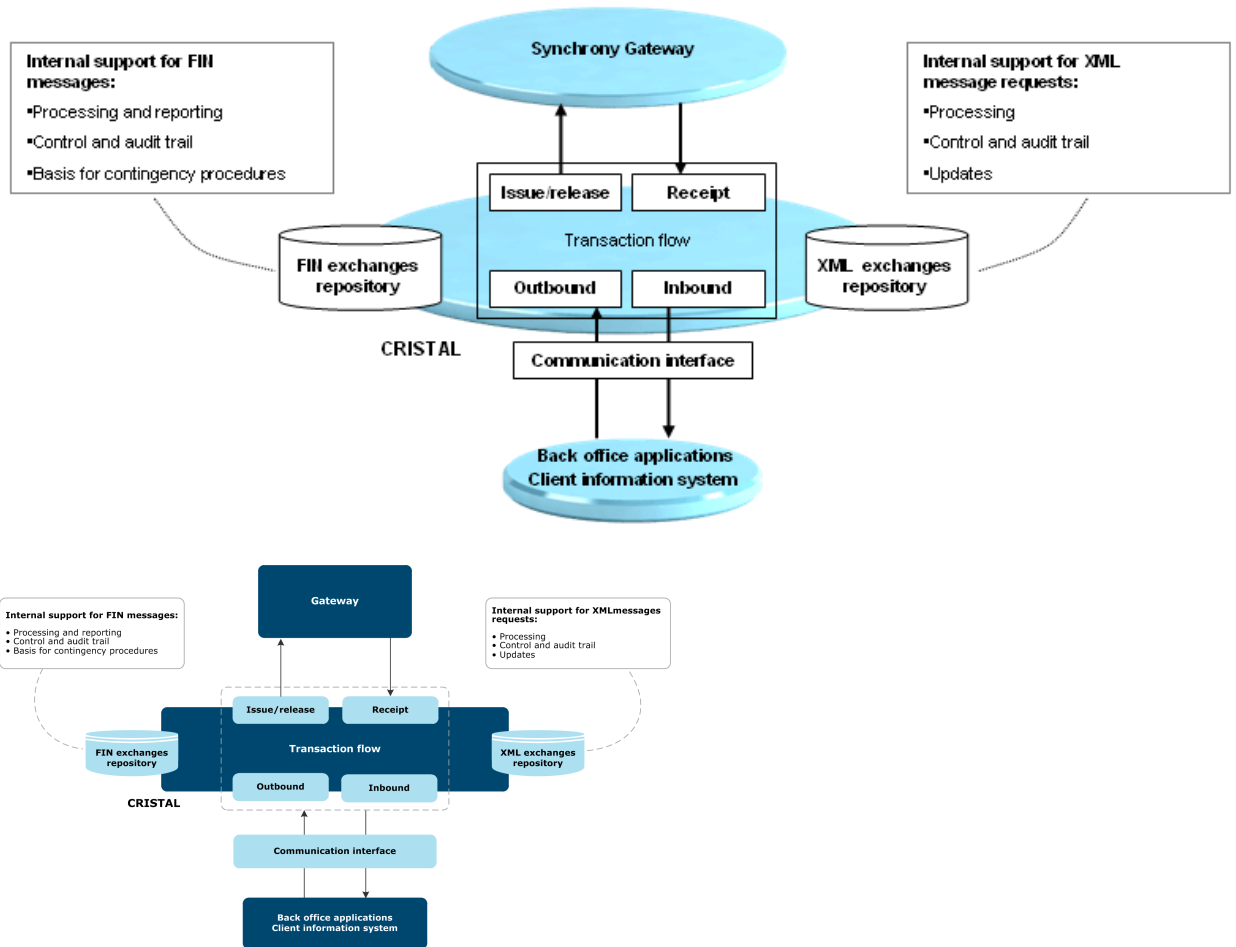
Communication between CRISTAL and Gateway

- CRISTAL and Gateway communicate through:
 - MQ/JMS for messages
 - PeSIT file transfers for files
- The TARGET2 protocol is handled by scripts that are launched on reception of the answers sent by the SSP (message or file) and also when a Gateway-issued SWIFTNet request fails.
- CRISTAL and Gateway exchange parameters which:
 - Enable Gateway to build the InterAct or FileAct requests
 - Return information about the execution of Gateway requests to CRISTAL Report

About CRISTAL

CRISTAL is a payment and liquidity management application developed by Diamis for the banking sector. CRISTAL handles all SWIFTNet exchanges that use TARGET2:

- Payment exchanges via FIN
- Information and control exchanges via InterAct and FileAct



TARGET2 parameters exchanged between Gateway and CRISTAL

The TARGET2 protocol parameters described in the following table are exchanged as JMS properties:

Parameter name	Description	Set by CRISTAL for sending routing	Set by Gateway as a report on the request execution
LocalPartner	Sender of the request. Defined in Gateway as the alias of a Local Trading Partner.	YES	YES
RemotePartner	Receiver of the request. Defined in Gateway as the alias of a Remote Trading Partner.	YES	YES
RequestRef	Request reference generated by CRISTAL. This parameter is sent to the partner.	YES	YES
RequestType	Request Type as defined in the TARGET2 documentation.	YES	NO
Service	SWIFTNet service name • trgt.papss (production) • trgt.papss!p (test)	YES	NO
ReportStatus	Gateway error number. Concatenation of the error, diagnosis and error codes.	NO	YES (only in the case of Gateway error)

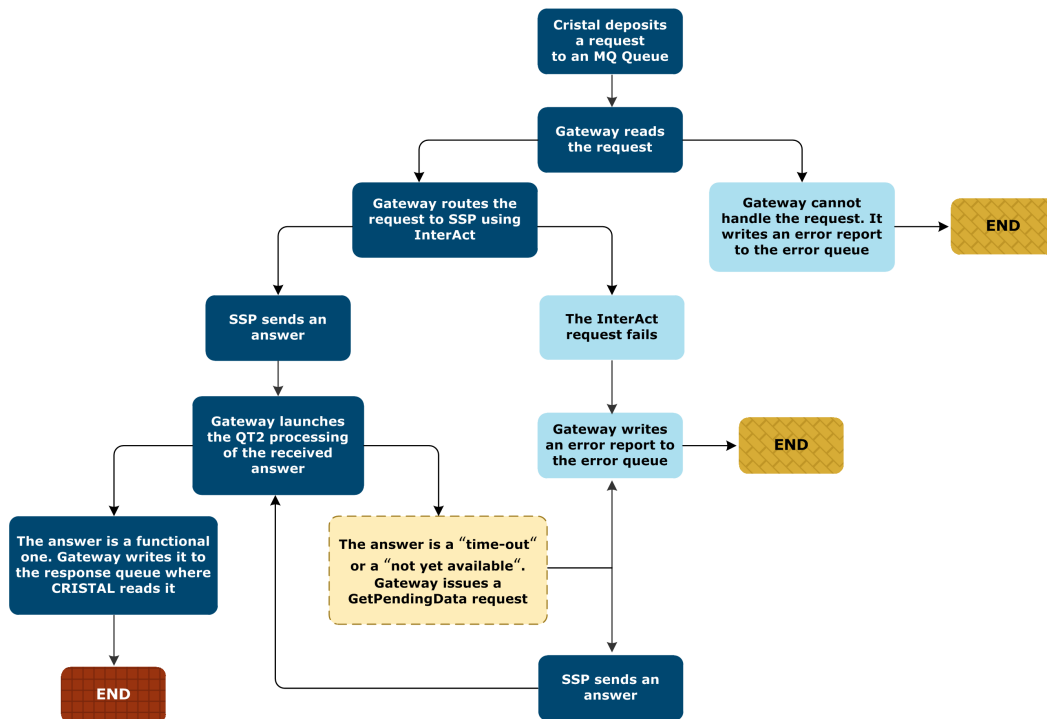
To retrieve files from the TARGET2 directory, CRISTAL sends the following JMS parameters to Gateway:

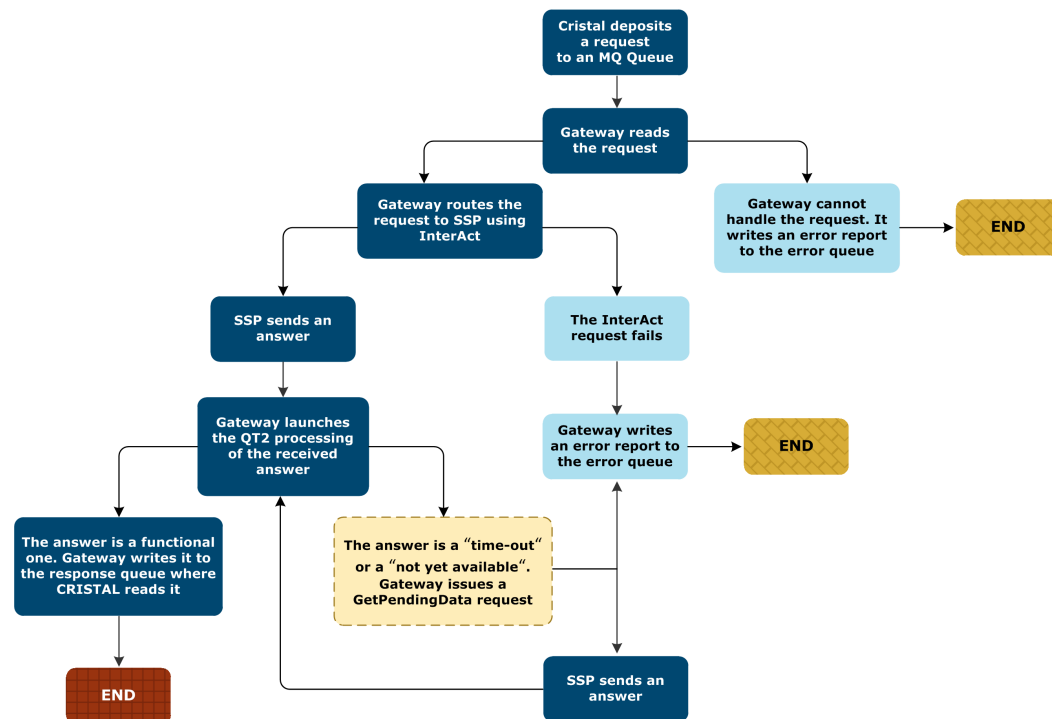
- **LocalPartner:** Alias of the Local Trading Partner
- **RemotePartner:** Alias of the Remote Trading Partner corresponding to the FileAct DN of the SSP
- **RequestRef:** Request reference
- **Service:** `getFileActDirectory`, (trgt.papss)
- **RequestType:**
 - `reda.xxx.target2.dirupdate` to get the delta TARGET2 Directory
 - `reda.xxx.target2.dirfull` to get the full TARGET2 Directory
- **FileName:** Name of the required file. For TARGET2 directories, the following syntax is required:
T2DIR<TTTT><Z><YYYYMMDD><SS>

Where:

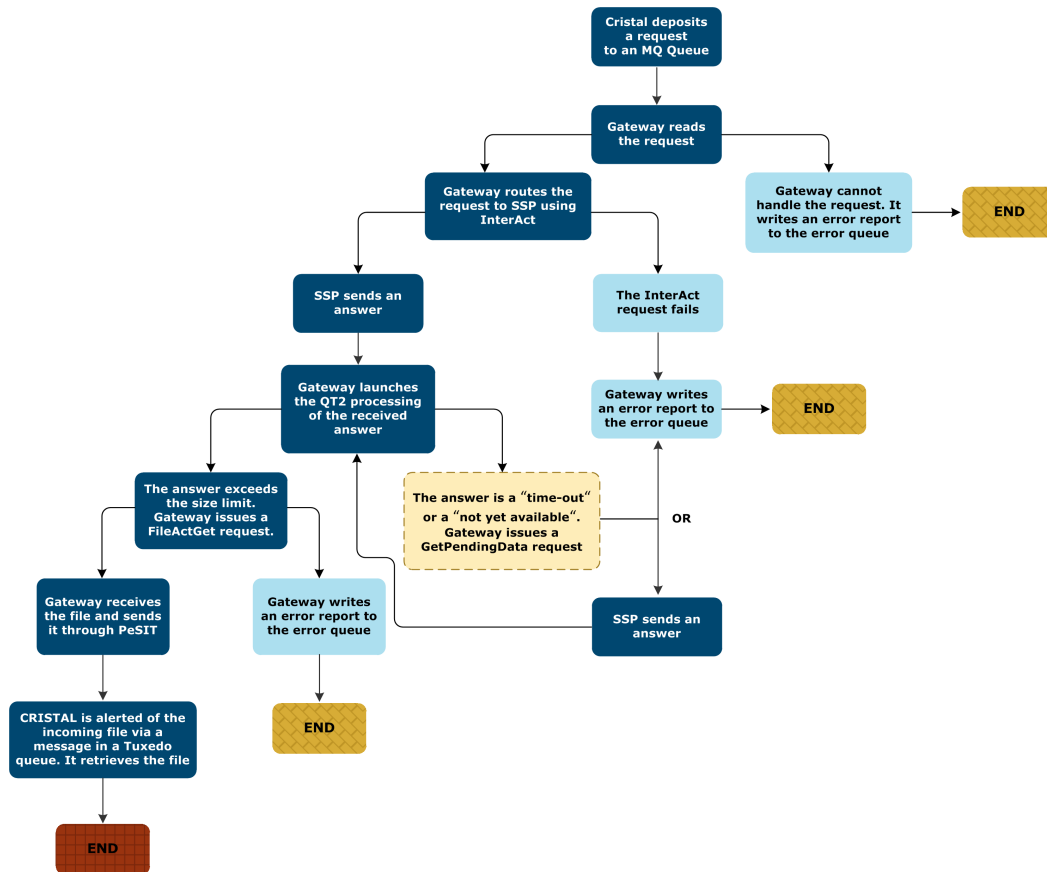
TTTT = FULL or UPDATE

Workflow diagrams: SSP answer in a message



Workflow diagrams: SSP answer in a message

Workflow diagrams: SSP answer in a file



Workflow diagrams: Gateway exchange handling

Message handling diagrams

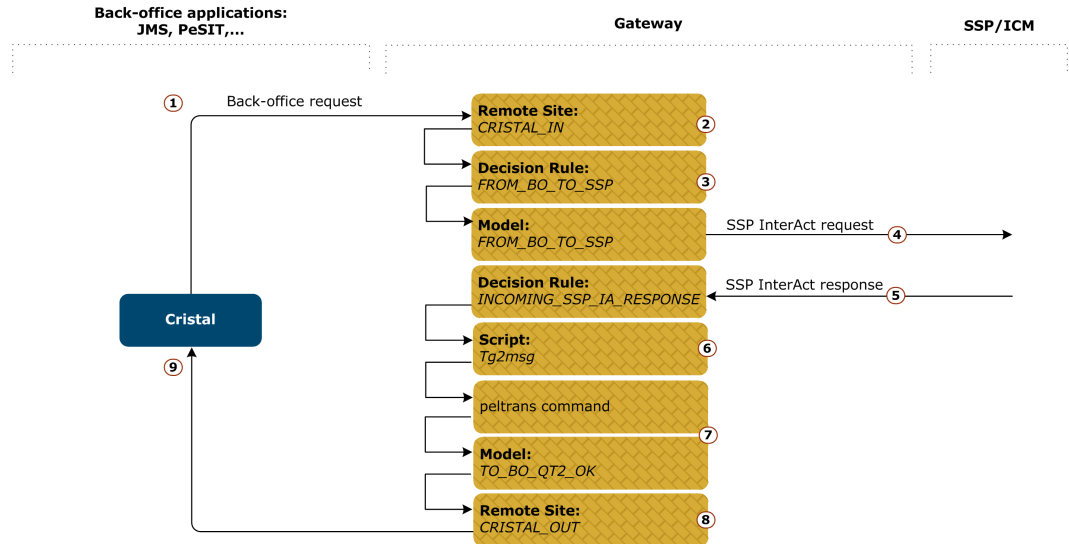
This topic contains flow diagrams and sequential descriptions that illustrate Gateway handling of QT2 exchanges between back-end applications and SWIFTNet.

The following transfer cases are illustrated:

- Case 1: [Gateway handling of successful TARGET2 InterAct transfer](#)
- Case 2: [Gateway handling of a transfer when InterAct request fails locally](#)
- Case 3: [Gateway handling of a transfer with InterAct request response error](#)
- Case 4: [Gateway handling of a transfer when no response is received to InterAct request \(timeout or not yet available\)](#)
- Case 5: [Successful Gateway handling of a transfer when file exceeds InterAct maximum size limit](#)

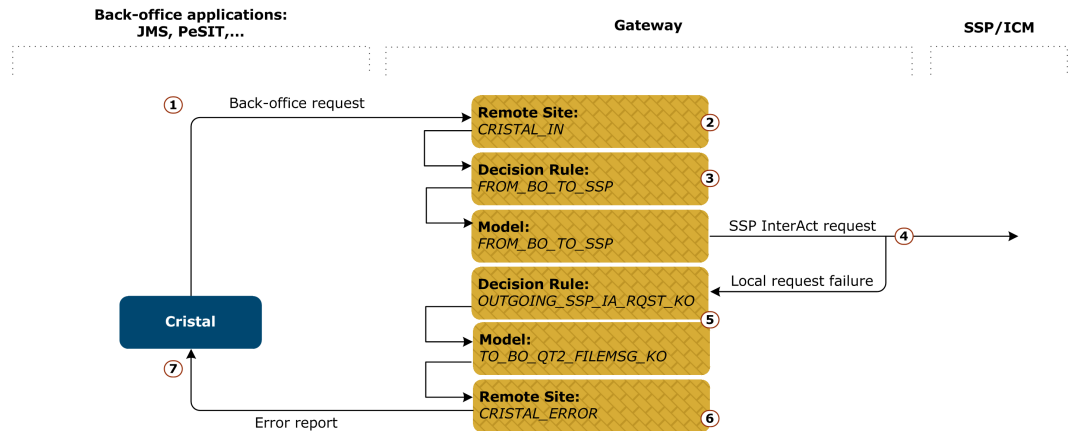
- Case 6: [Unsuccessful Gateway handling of a transfer when the file exceeds the InterAct maximum size](#)

Case 1: Gateway handling of successful TARGET2 InterAct transfer



1. CRISTAL deposits the TARGET2 XML request in the MQSeries queue.
2. Gateway reads the XML request through the CRISTAL_IN Remote Site.
3. Gateway channels the XML request through the Decision Rule FROM_BO_TO_SSP.
4. This Decision Rule triggers the transfer Model FROM_BO_TO_SSP, creating an InterAct transfer request to the SSP.
5. The SSP sends a response before the SWIFTNet time-out.
6. Gateway receives the response through the Decision Rule INCOMING_SSP_IA_RESPONSE. The Decision Rule starts the script tg2msg.
7. If the SSP response is the final response (business answer or technical answer with errcode=0000) then the tg2msg script forwards this answer to CRISTAL via a Gateway peltrans command, using the Model TO_BO_QT2_OK.
8. The peltrans command generates a transfer that sends the answer to the CRISTAL return queue through the CRISTAL_OUT Remote Site.
9. CRISTAL reads the answer in the return queue.

Case 2: Gateway handling of a transfer when InterAct request fails locally



1. CRISTAL deposits the TARGET2 XML request in the MQSeries queue.
2. Gateway reads the XML request through the CRISTAL_IN Remote Site.
3. Gateway channels the XML request through the Decision Rule FROM_BO_TO_SSP.
4. This Decision Rule triggers the transfer Model FROM_BO_TO_SSP, creating an InterAct transfer request to the SSP.
5. The InterAct request fails locally. This event is caught by the Decision Rule OUTGOING_SSP_IA_REQUEST_KO, which automatically launches the Model TO_BO_QT2_FILEMSG.
6. Gateway deposits the InterAct request failure error report to the CRISTAL error queue, via the CRISTAL_ERR Remote Site.
7. CRISTAL reads the error report in the error return queue.

Case 3: Gateway handling of a transfer with InterAct request response error



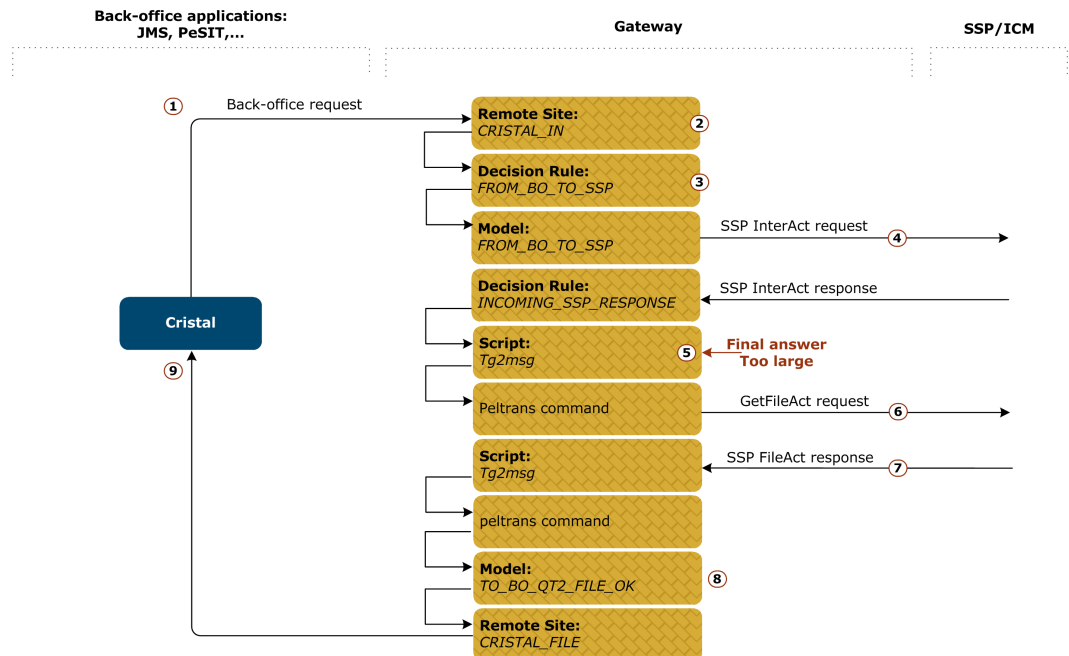
1. CRISTAL deposits the TARGET2 XML request in the MQSeries queue.
2. Gateway reads the XML request through the CRISTAL_IN Remote Site.
3. Gateway channels the XML request through the Decision Rule FROM_BO_TO_SSP.
4. This Decision Rule triggers the transfer Model FROM_BO_TO_SSP, creating an InterAct transfer request to the SSP.
5. The SSP sends a response before the SWIFTNet time-out. The received message is wrong (corrupted, not-compliant with standard, ...), therefore the tg2msg script sends an error report to CRISTAL through a peltrans command using the Model TO_BO_QT2_KO.
6. The peltrans command generates a transfer that deposits the error report to the CRISTAL error queue through the CRISTAL_ERR Remote Site.
7. CRISTAL reads the answer in the return queue.

Case 4: Gateway handling of a transfer when no response is received to InterAct request (timeout or not yet available)



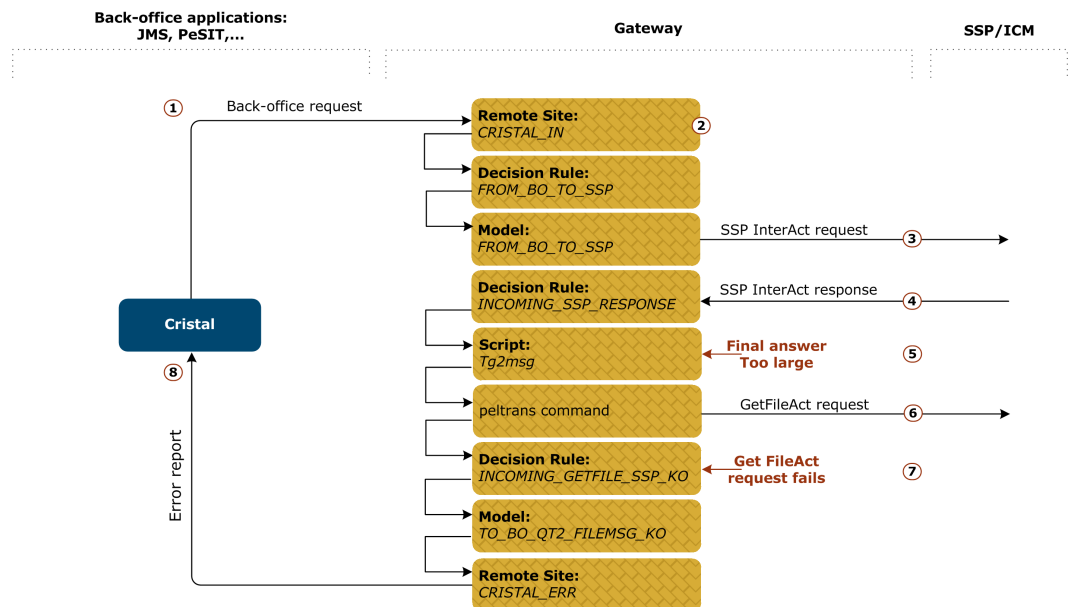
1. CRISTAL deposits the TARGET2 XML request in the MQSeries queue.
2. Gateway reads the XML request through the CRISTAL_IN Remote Site.
3. Gateway channels the XML request through the Decision Rule FROM_BO_TO_SSP.
4. This Decision Rule triggers the transfer Model FROM_BO_TO_SSP, creating an InterAct transfer request to the SSP.
5. The parsing of the response generates a return code that indicates the final answer is not available due to time-out or not_yet_available.
6. The tg2msg script sends a GetPendingData InterAct request, with the correct request reference, via a peltrans command.
7. The Gateway action depends on the request result:
 - If the response is OK, Gateway completes the exchange as in case 1.
 - If the new request fails locally, Gateway deposits an InterAct request failure error report to the CRISTAL error queue, via the CRISTAL_ERR Remote Site.
 - If the message returned from the SSP is incorrect, Gateway deposits an InterAct request failure error report to the CRISTAL error queue, via the CRISTAL_ERR Remote Site.

Case 5: Successful Gateway handling of a transfer when file exceeds InterAct maximum size limit



1. CRISTAL deposits the TARGET2 XML request in the MQSeries queue.
2. Gateway reads the XML request through the CRISTAL_IN Remote Site.
3. Gateway channels the XML request through the Decision Rule FROM_BO_TO_SSP.
4. This Decision Rule triggers the transfer Model FROM_BO_TO_SSP, creating an InterAct transfer request to the SSP.
5. Gateway tg2msg script parses the answer and generates a return code indicating that the final answer is too large. SSP makes the answer available through a Get FileAct.
6. The tg2msg script sends a Get FileAct request containing the correct logical file name via a peltrans command.
7. The tg2msg script processes the the GetFILE response.
8. The tg2file script sends the answer to CRISTAL through a peltrans command using the Model TO_BO_QT2_FILE_OK. This model generates a PeSIT transfer to the CRISTAL_FILE Remote Site.
9. The file is received by the file transfer monitor running on the CRISTAL platform, and is acquired by CRISTAL through an end of transfer script.

Case 6: Unsuccessful Gateway handling of a transfer when the file exceeds the InterAct maximum size



1. CRISTAL deposits the TARGET2 XML request in the MQSeries queue.
2. Gateway reads the XML request through the CRISTAL_IN Remote Site.
3. Gateway channels the XML request through the Decision Rule FROM_BO_TO_SSP.
4. This Decision Rule triggers the transfer Model FROM_BO_TO_SSP, creating an InterAct transfer request to the SSP.

5. Gateway tg2msg script parses the answer and generates a return code indicating that the final answer is too large. SSP makes the answer available through a Get FileAct.
6. The tg2msg script sends a Get FileAct request containing the correct logical file name via a peltrans command.
7. The Get FileAct request fails. The failure is caught by the INCOMING_GETFILE_SSP_KO decision rule. This Decision Rule automatically launches the Model TO_BO_QT2_FILEMSG_KO Model.
8. The error report about the InterAct request failure is deposited to the CRISTAL error queue through the CRISTAL_ERR Remote Site.

SSP Cash Management message-handling evolutions

Overview

Cash Management (CAMT) is a set of financial exchange message formats for business and technical XML messages that complies with the message standards used by the TARGET2 Single Shared Platform (SSP). As far as possible, the TARGET2 SSP conforms to the SWIFTNet message standards. The XML schema files provided by the TARGET2 SSP are SWIFT standard compliant.

From version 2.4, the TARGET2 SSP supports concurrent versions of schema files supplied by the following two Cash Management standards releases:

- CAMT version 3.0 – former version
- CAMT version 4.0 – updated November 2008 release version

CAMT message version handling

Concurrent CAMT version handling

SWIFT standards specify that all SSP responses to customer requests are returned in the same standard as the original request. For example, a request sent in CAMT V3.0 requires a response in CAMT V3.0 and a request sent in CAMT V4.0 requires a response in CAMT V4.0.

When two concurrent versions of a request exist, the SSP attributes the lower message number to the CAMT V3.0 formatted message and the higher number the CAMT V4.0 formatted message.

CAMT V3.0 examples:

- `<AppHdr xmlns="urn:swift:xsd:$ahV10">`
- `<Document xmlns="urn:swift:xsd:$camt.003.001.03">`

CAMT V4.0 examples:

- `<AppHdr xmlns="urn:swift:xsd:$ahV10">`
- `<Document xmlns="urn:swift:xsd:camt.003.001.04">`

In the preceding examples, the first element of each line sets the XML namespace as defined in the XML schema. In this case, the initial elements set namespaces for the application header and the business document using the tags <AppHdr> and <Document>. The namespace declaration is an attribute of these tags.

CAMT message XML tag evolutions

The following examples enable you to compare the alignment of XML tag names in the structure of CAMT V3.0 and V4.0 XML message tags:

- V3.0 message structure
 - Schema filename: \$camt.004.001.03.xsd
 - Request Type: camt.004.001.03
 - URN: :urn:swift:xsd:\$camt.004.001.03
 - Document Content <doc><camt.004.001.03>....
- V4.0 message structure
 - Schema filename: camt.004.001.04.xsd
 - Request Type: camt.004.001.04
 - URN: :urn:swift:xsd:camt.004.001.04
 - Document Content <doc><RtrAcct>

SSP 3.0 compliance

The FEX TARGET2 package is compliant with the current version of the SSP: version 3.0.

Checking the TARGET2 package installation prerequisites

Refer to *TARGET2 package installation prerequisites* in the Gateway [Installation Guide](#).

Checking the TARGET2 package installation prerequisites

Refer to *TARGET2 package installation prerequisites* in the Gateway [Installation Guide](#).

Checking the Gateway installation

After you install Gateway:

1. Start Gateway.
2. Create at least one SWIFT Remote Site. For details, refer to the SWIFTNet connector sections of the Axway Gateway documentation.
3. Check that the `$p_home_dir` variable does not contain a trailing slash "/" character.

For UNIX, for example:

```
/> echo $p_home_dir
/home/user/Gateway      # GOOD
/home/user/Gateway/    # BAD
```

4. If the variable contains a trailing slash character:

- Go to: `<Gateway install directory>/run_time/etc/`
- Edit the `profile` file to change the `$p_home_dir` path.
- Run the profile by entering the following command:

```
[UNIX]

/> gatestop

/> . ./run_time/etc/profile

/> gatestart
```

Installing the TARGET2 package

Refer to the *Gateway Installation Guide* > [Installing the TARGET2 package](#) for more information.

TARGET2

Configuration overview

The Axway Financial Exchange TARGET2 package is delivered in the FEX installation package. Alternatively download it from the Axway Support website: <https://support.axway.com>

There are two typical contexts for TARGET2 configuration:

- No version of TARGET2 is installed:
 - Unpack the Axway Financial Exchange TARGET2 package, and configure as required.
- TARGET2 v1.0.1 or v1.0.2 is already installed:
 - Delete the existing QT2 folder.
 - Unpack TARGET2 v1.0.3 last release.
 - Delete all objects (Trading Partners, Models, Decision Rules and Remote Sites) using the script file `cl_delobjCRISTAL.sh`
 - Import a new set of objects from the `objExports` folder using the script file `cl_`

```
objCRISTAL.sh
```

- To update a previous V103 to the latest release, download and apply the latest patch.

Editing TARGET2 files and objects

Prerequisite: Install the TARGET2 package as described in the *Installation Guide* > [Installing the TARGET2 package to Gateway](#).

Configure Axway Gateway for TARGET2 as follows:

1. Edit the `cl_objCRISTAL` script file which is located in the directory `qt2/objExport/`. In this file, set values for the following parameters that correspond to your network configuration:
 - LOCAL_SITE - alias of the Local Remote Site
 - SWIFT_REMOTE_SITE - alias of the SWIFTNet Remote Site
 - LOCAL_DN
 - COPYDIR - the directory you choose to use to locally copy the received files, in ROUTE mode
2. Edit the `qt2/config` file. In this file, set values for the following parameters:
 - MODE
 - FILE_ROUTING
 - FULLCPENGINE (FALSE for Gateway 6.10.1 and Gateway 6.11.2 Patch <10 ; TRUE for Gateway 6.11.2 Patch >=10 and later versions)
3. Run the script `cl_objCRISTAL`.
This script imports the set of objects that you downloaded and unzipped on your Gateway supporting machine to Gateway.
4. Manually configure the following Remote Site template objects:
 - CRISTAL_IN
 - CRISTAL_OUT
 - CRISTAL_ERR
 - CRISTAL_FILE

For configuration details, refer to [Customizing Gateway Remote Sites for TARGET2](#).

For a summary of the Gateway objects you manage for TARGET2, refer to [Gateway objects for TARGET2 exchanges](#).

For examples of configuring Gateway objects for multiple CRISTAL applications, refer to [Multiple CRISTAL configurations](#).

Imported objects

When you run the `qt2/objExport/cl_objCRISTAL` script, the following objects are imported from the `objExports` folder into Gateway:

- [Decision Rules](#)
- [Models](#)
- [Trading Partners](#)
- [Remote Sites](#)

Imported Decision Rules

The following figure illustrates the resulting set of imported Decision Rules.

DEFAULT			
Name	Type	Exec type	Comments
 FROM_BO_TO_SSP1	Decision Rule	Model: FROM_BO_TO_SSP1	Request
 TG2DIR_GETFILE_SSP1	Decision Rule	Model: TG2DIR_GETFILE_SSP1	Route
 INCOMING_SSP_IA_RESPONSE1	Decision Rule	Batch command: /swiftreq/users/swiftrdq/Axway/Synchrony_V4.2.0...	Launch
 INCOMING_GETFILE_SSP1	Decision Rule	Batch command: /swiftreq/users/swiftrdq/Axway/Synchrony_V4.2.0...	Launch
 OUTGOING_SSP_IA_REQ_KO1	Decision Rule	Model: TO_BO_QT2_FILEMSG_KO1	Route
 INCOMING_GETFILE_SSP_KO1	Decision Rule	Model: TO_BO_QT2_GETFILE_KO1	Route
 INCOMING_SSP_IA_REQUEST	Decision Rule	Batch command: /swiftreq/users/swiftrdq/Axway/Synchrony_V4.2.0...	Only f

Imported Models

The following figure illustrates the resulting set of imported Models.

Model	Arguments	Comments
 FROM_BO_TO_SSP1	destination alias, originator alias, direction, type, application, file component, param1...	Request
 TG2DIR_GETFILE_SSP1	destination alias, originator alias, direction, type, application, param1, mode, file nam...	used to
 TO_BO_QT2_FILE_OK1	destination alias, originator alias, direction, type, application, file component, snd ms...	Routes
 TO_BO_QT2_FILEMSG_KO1	destination alias, originator alias, direction, type, application, file component, mode, ...	Routes
 TO_BO_QT2_FILEMSG_KO...	destination alias, originator alias, direction, type, application, file component, mode, ...	Routes
 TO_BO_QT2_GETFILE_KO1	destination alias, originator alias, direction, type, application, file component, mode, ...	Routes
 TO_BO_QT2_OK1	destination alias, originator alias, direction, type, application, file component, mode, ...	Routes

Imported Trading Partners

The following figure illustrates the resulting set of imported Trading Partners.

Alias	Format	Name	Is Local Partner	Comments
 QT2_TP_LOCAL1	SWIFTNet	o=ptsafrrk,o=swift	Yes	Local SWIFT
 QT2_TP_REMOTE_FILE_SSP	SWIFTNet	o=ptsafrrk,o=swift	No	SSP FileAct
 QT2_TP_REMOTE_FILE_SSP_TET	SWIFTNet	cn=fileact,ou=tet,o=trgtxepm,o=swift	No	SSP FileAct
 QT2_TP_REMOTE_SSP	SWIFTNet	o=ptsafrrk,o=swift	No	SSP InterAct
 QT2_TP_REMOTE_SSP_TET	SWIFTNet	cn=interact,ou=tet,o=trgtxepm,o=swift	No	SSP InterAct

Imported Sites

The following figure illustrates the resulting set of imported Remote Sites.

Alias	Proto.ID	Sta...	Init	Resp	Protocol	Comments
 CRISTAL_OUT1	CRISTAL_OUT1	T-R	--	R-	JMS	Linked to the Queue where Gateway put the OK response
 CRISTAL_IN1	CRISTAL_IN1	TI-	-E	--	JMS	Linked to the Queue where Cristal put the request
 CRISTAL_FILE1	CRISTAL_FILE1	T-R	R-	RE	PeSIT E	The destination of the SSP answers when it is sent as a file
 CRISTAL_ERR1	CRISTAL_ERR1	T-R	--	R-	JMS	Linked to the Queue where Gateway put its internal Error

Customizing Gateway Remote Sites for TARGET2

After importing the TARGET2 object set, you must modify the configuration of the Gateway Remote Site objects to correspond to your network environment.

The TARGET2 object set includes the following four Remote Sites objects for use in communication with a CRISTAL instance. You require each of the four Remote Sites to enable communication with a single CRISTAL instance. If you have more than one CRISTAL instance, you define four Remote Site objects for each instance.

You can modify the names attributed to the Remote Sites in the following examples, depending on the internal naming rules of your installation.

Site name	Description
CRISTAL_IN1	JMS Remote Site Sender only Links to the Queue where CRISTAL deposits the request
CRISTAL_OUT1	JMS Remote Site Receiver only Links to the Queue where Gateway deposits the OK response received from SSP
CRISTAL_ERR1	JMS Remote Site Receiver only Links to the Queue where Gateway deposits internal error reports
CRISTAL_FILE1	PeSIT Remote Site Receiver only The destination of the SSP answers when it is sent as a file.

Open each CRISTAL Remote Site object in the Gateway GUI and check the settings in the **General** and **JMS** tabs as illustrated in the examples below. Adapt the settings to correspond to your network requirements.

SFE : New Transfer Request

SWIFTNet		JMS		OFTP		X.420	
General	Attributes	Details	PeSIT	Etebac5	HTTP/FTP	SMTP	Trading partner

General information

Model:

Type: Mode: Direction:

Application:

Ack option: Purge option: Permanent:

Sites

☒ Originator alias:

☐ Destination alias:

☐ Local agent alias:

☐ Remote agent alias:

Originator:

Destination:

The screenshot shows the 'SFE : New Transfer Request' dialog box with the 'JMS' tab selected. The 'Local attributes' section includes a 'File component' field with the path 'dq/users/swiftrdg/Axway/Synchrony_V4.1.1/qt2/xml/InitialRequest.xml' and a 'Record format' dropdown. The 'Transfer attributes' section includes 'Directory name' and 'Directory alias' fields. The 'File name' and 'File type' fields are also present. The 'Format' section includes 'Padding character (Hexa)' and 'Truncating allowed' dropdowns. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

The preceding JMS tab illustration is intended only as an example. The values you enter in this tab vary depending on your network configuration.

Open the Remote Site object CRISTAL_FILE1 in the Gateway GUI. Check the settings of the following tabs:

- General
- Options
- Network address
- PeSIT
- Net security

Adapt the tab settings to correspond to you network requirements.

SFE : CRISTAL_FILE1 - Remote Site

General Options Network address PeSIT Net security

Alias CRISTAL_FILE1

Protocol PeSIT E

Protocol identifier CRISTAL_FILE1

Group name GDEFAULT

Transfer owner

Partner local alias

Comments The destination of the SSP answers when it is sent as a file.

Directory for received files

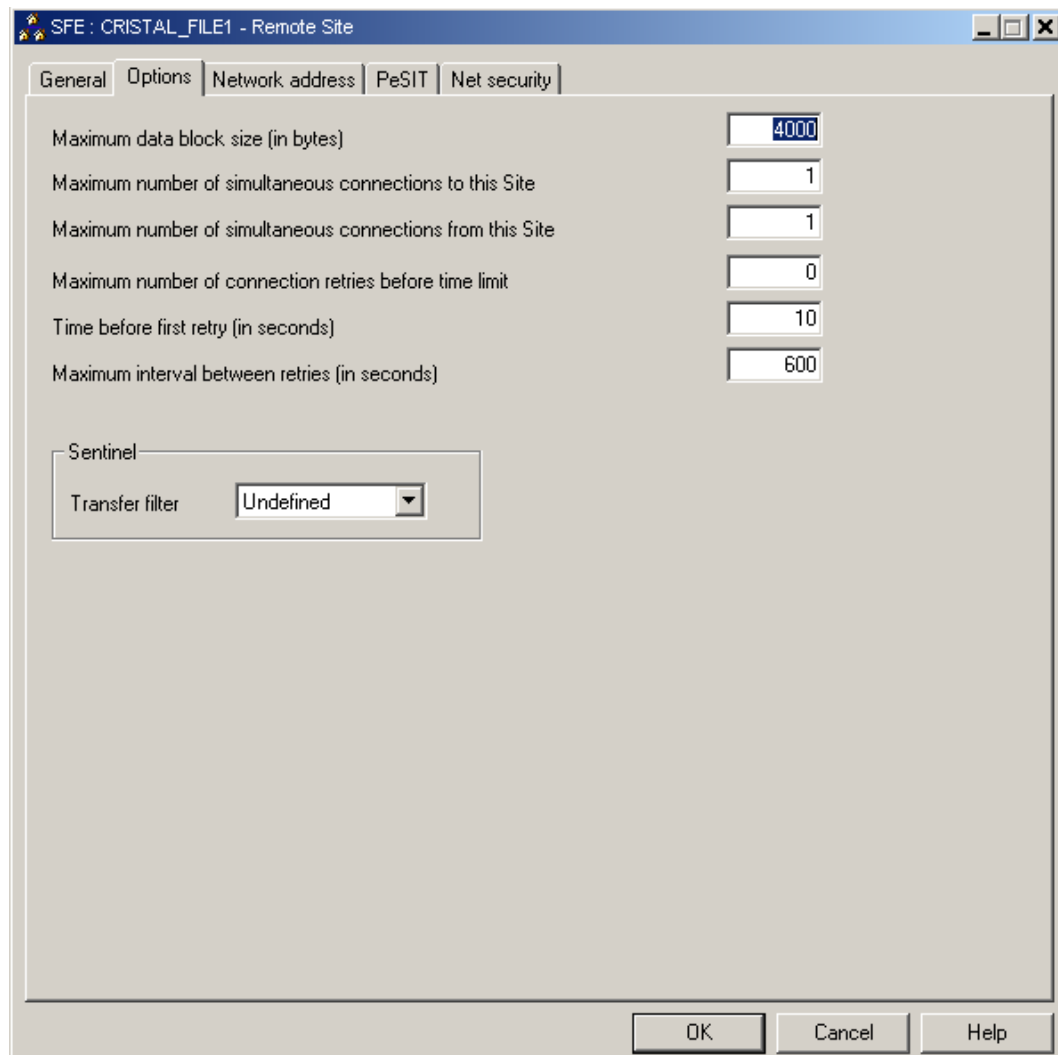
Role

☐ Initiator/Sender	☒ Responder/Sender
☒ Initiator/Receiver	☒ Responder/Receiver

Initialization mode

☐ Initiating Site	☒ Trace
☒ Responding Site	

OK Cancel Help



The screenshot shows a Windows-style dialog box titled "SFE : CRISTAL_FILE1 - Remote Site". It has five tabs: "General", "Options", "Network address" (which is selected), "PeSIT", and "Net security".

Under the "Network address" tab, there are two main sections. The left section contains four radio buttons: "Main address" (which is selected), "Alternative address n°1", "Alternative address n°2", and "Alternative address n°3". Below these are two dropdown menus: "Network type" set to "TCP/IP" and "Initial communication path" set to "Main address".

The right section is a larger area containing four input fields:

- "IP address or machine name" with the value "172.17.2.26".
- "Port number" with the value "6330".
- "Calling address" with the value "172.17.2.26".
- "Network Proxy" with an empty dropdown menu.

At the bottom right of the dialog box are three buttons: "OK", "Cancel", and "Help".

SFE : CRISTAL_FILE1 - Remote Site

General | Options | Network address | PeSIT | Net security

Pre-connection phase

ID/Password to send to partner

ID/Password to receive from partner

Connection phase

Client password to send / to check

Server password to send / to check

☒ Restart allowed ☒ Resynchronization allowed

Synchronization acknowledgement window: 4

Interval between synchronization points (KBytes): 100

Acknowledgement option: Undefined

☐ Negative acknowledgement option

Message

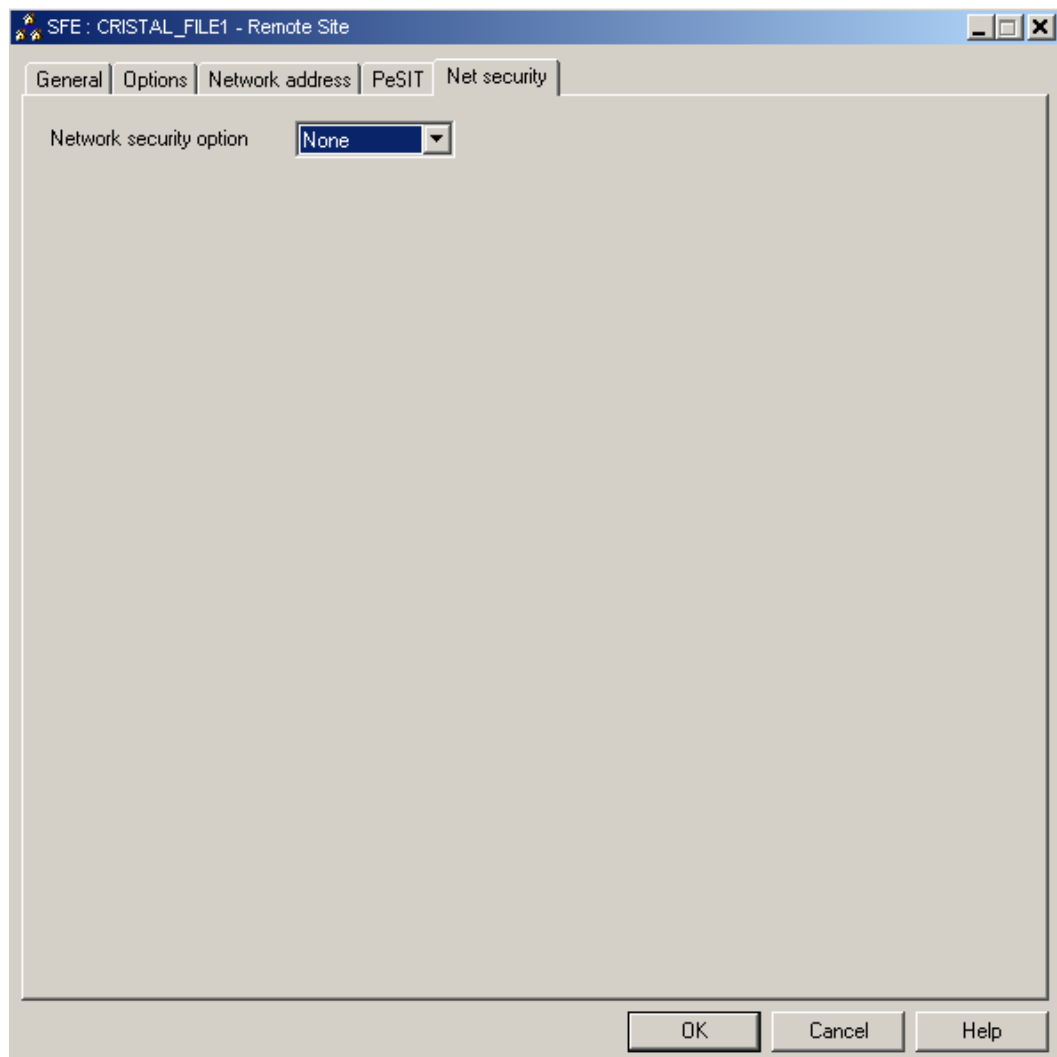
☐ Change direction support

Polling interval (in minutes): 0 Maximum connections: 0 ☐ Activated

☐ Permanent connection

Connection hold delay: 0 Heartbeat rate: 0

OK Cancel Help



Activate the PeSIT protocol in the Gateway Configuration menu. Deactivate SSL/TLS options in TCP Advanced options.

Be sure that a CGate matching the CRISTAL_FILE1 site is available and enabled. Check the IP address, listening port and protocol.

GPHSE				
Name	State	User login	Client ID	Server ID
PHSE	Enabled	*	*	*
CRISTAL_FILE1	Enabled	*	CRISTAL_FILE1	*

SFE : CRISTAL_FILE1 - CGate

General | Security | Output parameters | VFD Rights | Calling Address Controls

Name: CRISTAL_FILE1

Parent CGateGroup: GPHSE

Comments:

State: Enabled

HTTP / FTP / SFTP / PeSIT pre-connection

Identification

Login user name: * Change password: Never

Checking

Login password: *

PHSE

Identification

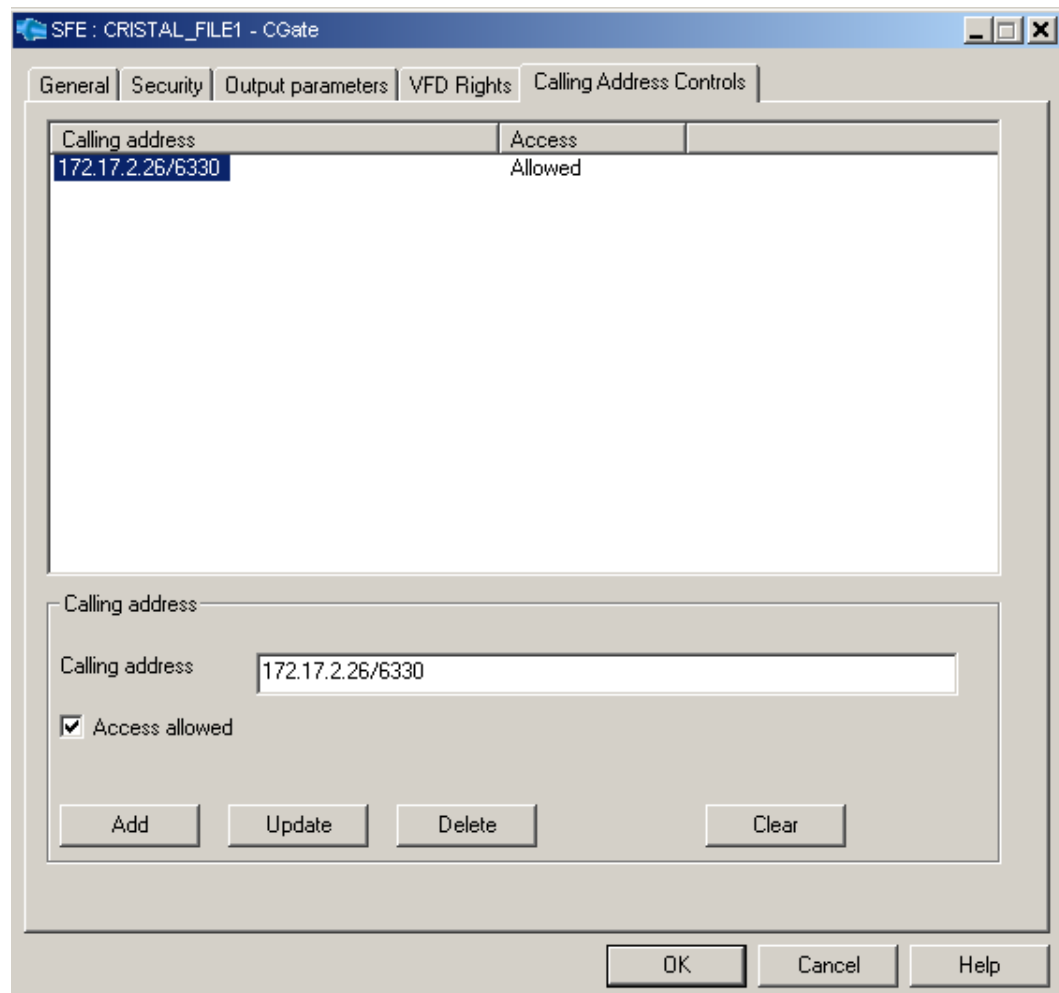
Client ID: CRISTAL_FILE1

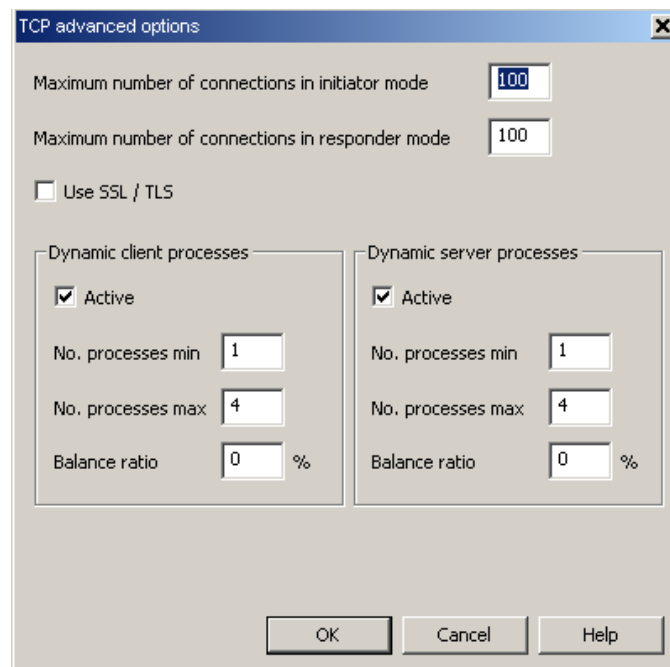
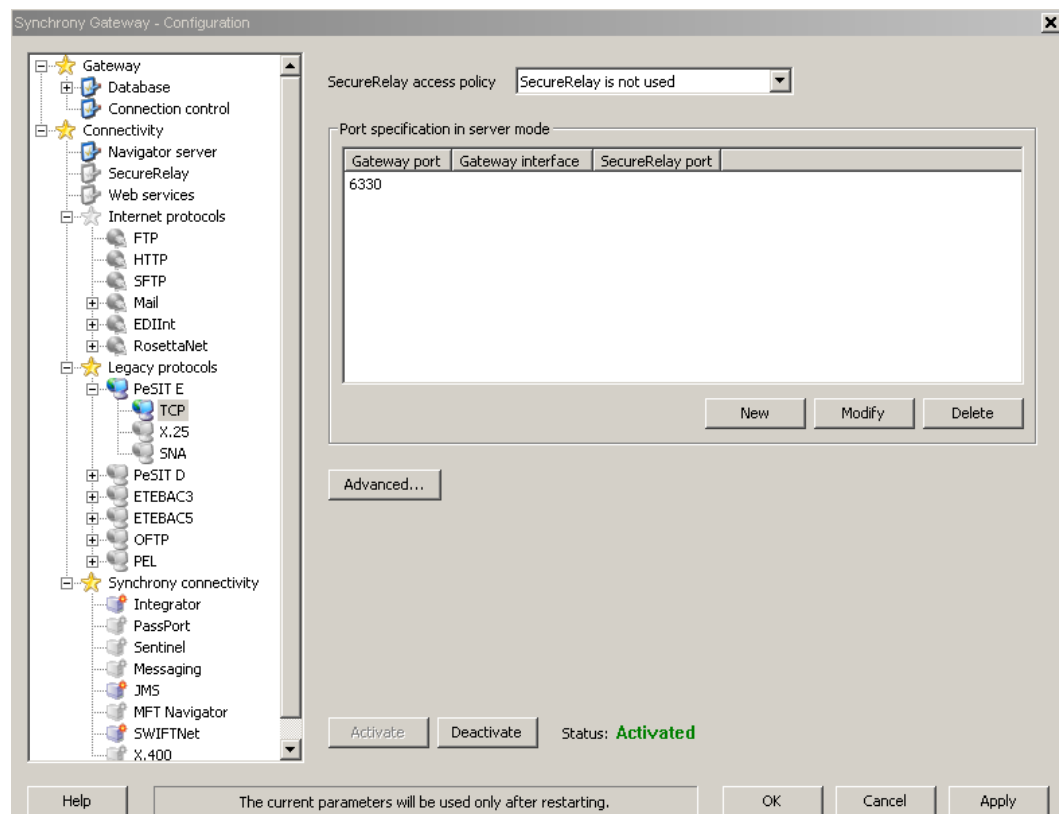
Server ID: *

Checking

Client password: *

OK Cancel Help





Checking the SWIFT Remote Site

In addition to the CRISTAL Remote Site objects, you need a Remote Site object that represents the SWIFT Network. This is the same object that you use for other interchanges with SWIFTNet via the SAG/SNL using the SWIFT Remote API (RA). The following figures illustrate an example configuration of the SWIFT Remote Site object.

The screenshot shows a configuration window titled "SFE : RSS - Remote Site". It has three tabs: "General", "Options", and "SWIFTNet". The "General" tab is active. The fields are as follows:

- Alias: RSS
- Protocol: SWIFTNet (dropdown menu)
- Protocol identifier: (empty text field)
- Group name: GDEFAULT
- Transfer owner: (empty text field)
- Partner local alias: (empty text field)
- Comments: (empty text area)
- Directory for received files: (empty text field with a folder icon button)

Below these fields are two sections:

- Role**
 - ☒ Initiator/Sender
 - ☒ Initiator/Receiver
 - ☒ Responder/Sender
 - ☒ Responder/Receiver
- Initialization mode**
 - ☒ Initiating Site
 - ☒ Responding Site
 - ☒ Trace

At the bottom right are three buttons: "OK", "Cancel", and "Help".

SFE : RSS - Remote Site

General Options SWIFTNet

Login Password

RA instance

Message partner

Event end point

Store and Forward only

Queues

SNL end point

Alternate RA installation pathname

Alternate configuration file

Maximum number of client processes

Maximum number of server processes

Sign ident list

Decrypt ident list

Backup site alias

OK Cancel Help

Customizing Gateway Decision Rules for TARGET2

After importing the TARGET2 object set, you must modify the configuration of the Decision Rule objects to correspond to your network environment.

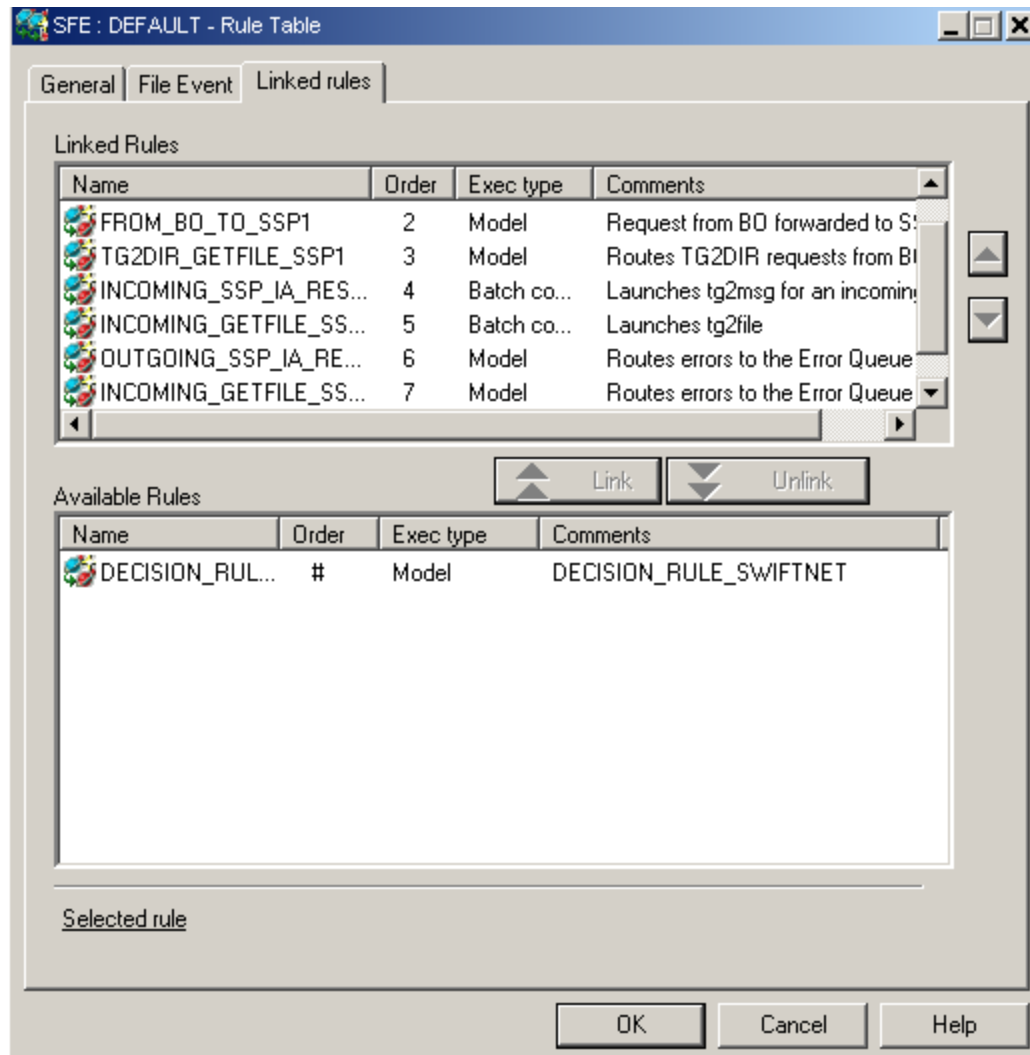
After you import Decision Rules:

1. In Gateway Navigator, right-click **Event Management\Transfer Change State\Default**.
2. Click **Properties**.
3. In the **General** tab:
 - Select **Active**
 - For Default execution type select **No action**
 - Select **Link input and output transfers**
 - For Log level select **Full**

The screenshot shows a Windows-style dialog box titled "SFE : DEFAULT - Rule Table". It has three tabs: "General", "File Event", and "Linked rules". The "General" tab is selected. Inside the tab, there are several fields: "Name" with the value "DEFAULT", "Type" with the value "Transfer Change State", and a "Comments" text box. Below these is a checked checkbox labeled "Active". Underneath is a "Default execution type" dropdown menu currently set to "No action". At the bottom of the tab is a "Log level" dropdown menu set to "Full". At the very bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

4. In the **Linked rules** tab:

- Move Decision Rules from Available rules to Linked rules using the **Link** button.
- Click **OK**.



5. Customize parameters according to your configuration requirements.

Gateway objects for TARGET2 exchanges

The following table lists the Gateway objects you require for TARGET2 exchanges, and indicates if they are provided in the installation package or if you must create them.

Object	Description	Comments

Remote Sites

You import the four Remote Site objects listed in this section: You then modify the objects, depending on your network configuration.

Object	Description	Comments
CRISTAL_IN1	JMS Remote Site. Sender only Links to the Queue where CRISTAL put the request	Provided in the package. To be configured.
CRISTAL_OUT1	JMS Remote Site. Receiver only Links to the Queue where Gateway put the OK response received from SSP	Provided in the package. To be configured.
CRISTAL_ERR1	JMS Remote Site. Receiver only Links to the Queue where Gateway put its internal Error report	Provided in the package. To be configured.
CRISTAL_FILE1	PeSIT remote Site. Receiver only The destination of the SSP answer when it is sent as a file.	Provided in the package. To be configured.

Local and Remote Trading Partners

You import the five Trading Partner objects listed in this section: You then modify the objects, setting the appropriate name, service and request type.

QT2_TP_LOCAL1	The local SWIFTNet Trading Partner. - Alias: Use the BIC Code as alias. - Name: Use the DN name. - Direction allowed: update_tradepart -a [use the BIC Code as the alias] -dir_all B - Mode allowed: update_tradepart -a [use the BIC Code as the alias] -mod_all B	Provided in the packaging. To be modified. Use the BIC Code as the alias.

Object	Description	Comments
QT2_TP_REMOTE_SSP	Remote SWIFTNet Trading Partner for InterAct. The Payload_Type Option must be set to embedded for the InterAct Trading Partner. - Alias: Use the BIC Code as alias. - Name: use the DN name. - Service name: Use the appropriate service name InterAct. - Request type: Use the appropriate Request type. - Direction allowed: update_tradepart -a [use the BIC Code as the alias] -dir_all B - Mode allowed: update_tradepart -a [use the BIC Code as the alias] -mod_all B - Payload type option: peladm update_tradepart -a [use the BIC Code as alias]-pto embedded - Receipt request synchronized: update_tradepart -a [use the BIC Code as the alias] -rrs Y ... Note: Make sure the Remote SSP partner is in pull-mode - if not: update_tradepart -a [your SWIFTNet remote Trading Partner -rrs Y:].	Provided in the package. To be modified. Use the BIC Code as the alias.

Object	Description	Comments
QT2_TP_REMOTE_FILE_SSP	Remote SWIFTNet Trading Partner for FileAct. - Alias: Use the BIC Code as alias - Name: Use the DN name - Service name: Use the appropriate FileAct service name - Request type: Use the appropriate Request type - Receipt DN: Use the appropriate Receipt DN - Receipt request type: Use the appropriate request type. Note: Make sure that the Request type of the Remote FILE SSP partner is set.	Provided in the package. To be modified.
QT2_TP_REMOTE_SSP_TET	SSP InterAct - Customer Test Environment	Provided in the package. To be modified.
QT2_TP_REMOTE_FILE_SSP_TET	SSP FileAct - Customer Test Environment	Provided in the package. To be modified.
Decision Rules You can optionally import any of the Decision Rule objects listed in this section: You then modify the objects, depending on you production requirements.		
FROM_BO_TO_SSP1	Selects incoming requests received through the CRISTAL_IN Site to route them to the SSP via the Model FROM_BO_TO_SSP. Selection criteria: - mode = R (Responder) - direction =I (input) - originator alias = CRISTAL_IN1 - property_RequestType # *xxx*	Provided in the package. To be modified.

Object	Description	Comments
INCOMING_GETFILE_SSP1	Selects the incoming TARGET2 files received from SSP through a Get FileAct, and routes them to CRISTAL via the script tg2file.sh/tg2file.bat. Uses TO_BO_QT2_FILE_OK1 TO_BO_QT2_FILEMSG_KO_P1 /home/swiftrdq/cristal/ Selection criteria: • direction=I (input) • protocol=SWIFTNET • trade destination alias=QT2_TP_LOCAL1 • trade service= trgt.* • type=TRANS (FileAct) • State= E or W (Ended or waiting for Ack) • userProcess=N	Provided in the package. To be modified.
INCOMING_GETFILE_SSP_KO1	Selects TARGET2 FileAct Get requests that failed and sends an error report to CRISTAL_err by directly invoking the Model TO_BO_QT2_FILEMSG_KO Selection criteria: • direction=I (input) • protocol=SWIFTNET • State=C or F or X (cancelled, frozen, NACKed) • trade destination alias=QT2_TP_LOCAL1 • trade service= trgt.* • type=TRANS (FileAct) • userProcess=N	Not provided in the package.

Object	Description	Comments
INCOMING_SSP_IA_REQUEST1	This is used only for Simulation. Selects requests and routes them to the dedicated script <code>ssp_pull_s.sh</code> and launches the Model <code>FROM_BO_TO_SSP1</code>. Selection criteria: • <code>direction=I</code> (Input) • <code>mode=I</code> (Responder) • <code>protocol=SWIFTNET</code> • <code>state=W</code> (Ended_to_ack) • <code>type=MSG</code> (InterAct) • <code>userProcess=N</code>	Provided in the package.
INCOMING_SSP_IA_RESPONSE1	Selects the incoming answers received via an InterAct reception from the SSP, and routes them to the dedicated script <code>tg2msg</code> by using <code>TO_BO_QT2_OK TO_BO_QT2_FILEMSG_KO</code> Selection criteria: • <code>direction=I</code> (input) • <code>mode=R</code> (Responder) • <code>protocol=SWIFTNET</code> • <code>trade originator alias=QT2_TP_LOCAL1</code> • <code>trade service= trg*. *</code> • <code>type=MSG</code> (InterAct) • <code>state= E or C</code> (Ended or Cancelled) • <code>userProcess=N</code>	Provided in the package.

Object	Description	Comments
OUTGOING_SSP_IA_REQ_KO1	Selects outgoing InterAct requests that fail locally and automatically sends CRISTAL a report error via JMS. It launches the Model TO_BO_QT2_FILEMSG_KO. Selection criteria: • direction=O (output) • mode=I (Initiator) • protocol=SWIFTNET • State=C (cancelled) • trade originator alias=QT2_TP_LOCAL1 • trade service= trgt.* • type=MSG (InterAct) • userProcess=N	Provided in the package.
TG2DIR_GETFILE_SSP1	Selects the requests sent by CRISTAL to get the TARGET2 directory file and routes them to SSP through the TG2DIR_GETFILE_SSP Model, called by the tg2file script Selection criteria: • mode = R (Responder) • direction =I (input) • originator alias = CRISTAL_IN1 • state=E • service=*xxx*	Provided in the package. For retrieving files in the TARGET2 directory.

Models

You can change the name of the Remote Site, and use the routing of your choice. Seven Models are available. They are provided in the package but must be modified.

Object	Description	Comments
FROM_BO_TO_SSP1	Sends the Get Request received from CRISTAL to the SSP via InterAct. This Model sets the SWIFTNet transfer parameters by using the JMS transfer parameters including the JMS properties. This Model is automatically activated by Gateway on reception of the CRISTAL JMS request Model parameters: • appli=DEFAULT_A • dest_alias=[SWIFTNet Remote site] • direction=O (output) • file_component=&(path_name) • mode=I (initiator) • org_alias=[Local Site] • param1=&(property_RequestRef) • trade_dest_alias=&(property_RemotePartner) • trade_org_alias=&(property_LocalPartner) • trade_request_type=&(property_RequestType) • trade_service=&(property_Service) • type=MSG (InterACT)	Provided in the package. To be modified.

Object	Description	Comments
TG2DIR_ GETFILE_SSP1	Gets the TARGET2 directory file via FileAct. This Model sets the SWIFTNet transfer parameters by using the JMS transf parameters. Model parameters: • appli=DEFAULT_A • dest_alias=[SWIFTNet Remote site] • direction=I (Input) • File_Name=&(property_FileName) • mode=I (initiator) • org_alias=[Local Site] • param1=&(property_RequestRef) • trade_dest_alias=&(property_LocalPartner) • trade_org_alias=&(property_RemotePartner) • trade_request_type=&(property_RequestType) • trade_service=&(property_Service) • type=TRANS (FileAct)	Provided in the package For retrieving files in the TARGET2 directory.

Object	Description	Comments
TO_BO_QT2_FILE_OK1	Sends CRISTAL the final answer received from the SSP via a Get FileAct. It is called by the tg2file script through a peltrans command. The Model name and other parameters are passed in the peltrans command to set the transfer parameters that depend on the parameters of the initial Get FileAct request. Model parameters: • appli=DEFAULT • dest_alias=CRISTAL_FILE1 • direction=O • property=BodyType=text • file_component= &(path_name) • file name=QT2ANSW • mode=I • org_alias=[Local Site] • snd msg= &(param1) • type=TRANS	Provided in the package. To be modified.

Object	Description	Comments
TO_BO_QT2_FILEMSG_KO1	Sends CRISTAL error reports related to failure of a TARGET2 Get request, or related to a local failure of a TARGET2 InterAct request. This Model sets the JMS transfer parameters by using the SWIFTNet transfer parameters. Model parameters: • appli=DEFAULT • dest_alias=CRISTAL_ERR1 • direction=O (Output) • file_component=[/swiftreq/users/swiftreq/Axway/qtt2/objExports\..\aupver] • property=LocalPartner=&(trade_dest_alias) • property=RemotePartner=&(trade_org_alias) • property=RequestType=&(trade_request_type) • property=Service=&(trade_service) • property=RequestRef=&(param1) • property_ReportStatus=&(last_end_diag)-&(last_end_err)-&(last_end_reason) • property=bodyType=string • mode=I • org_alias=[Local Site] • type=TRANS	Provided in the package

Object	Description	Comments
TO_BO_QT2_FILEMSG_KO_P1	Sends CRISTAL error reports related to failure of a TARGET2 Get request, or related to a local failure of a TARGET2 InterAct request. This Model sets the JMS transfer parameters by using the SWIFTNet transfer parameters. Model parameters: • appli=DEFAULT • dest_alias=CRISTAL_ERR1 • direction=O (Output) • file_component [/swiftrdq/users/swiftrdq/Axway/qt2/objExports\..\aupver] • property=bodyType=string • mode=I • org_alias=[Local Site] • type=TRANS	Provided in the package

Object	Description	Comments
TO_BO_QT2_GETFILE_KO1	Sends CRISTAL the final answer received from the SSP via a Get FileAct. It is called by the tg2file script through a peltrans command. The Model name and other parameters are passed in the peltrans command to set the transfer parameters that depend on the parameters of the initial Get FileAct request. Model parameters: • appli=DEFAULT • dest_alias=CRISTAL_ERR1 • direction=O (Output) • file_component=[/swiftreq/users/swiftreq/Axway/qt2/objExports\..\aupver] • property=LocalPartner=&(trade_dest_alias) • property=RemotePartner=&(trade_org_alias) • property=RequestType=&(trade_request_type) • property=Service=&(trade_service) • property=RequestRef=&(param1) • property_ReportStatus=&(last_end_diag)-&(last_end_err)-&(last_end_reason) • property=bodyType=string • mode=I • org_alias=[Local Site] • type=TRANS	Provided in the package.

Object	Description	Comments
TO_BO_QT2_OK1	Sends to CRISTAL the final answer received from the SSP through InterAct. This model sets the default JMS transfer parameters. It is called by the tg2msg script through a peltrans command. Model name and other parameters are passed in the peltrans command in order to set the transfer parameters that depend on the parameters of the initial Get InterAct request. Model parameters: • appli=DEFAULT • dest_alias=CRISTAL_OUT1 • direction=O (Output) • &(path_name) • property=bodyType=string • mode=I • org_alias=[Local Site] • type=TRANS	Provided in the package. To be modified.
Scripts The following scripts are provided in the package and must be modified.		
tg2msg	Launched by the INCOMING_SSP_IA_RESPONSE Decision Rule to process the answers returned by the SSP via a Get InterAct.	Provided in the package To modify, add the Initial transfer parameters to the peltrans commands
Tg2file	Launched by the Decision Rule INCOMING_GETFILE_SSP to process the answers returned by the SS via a Get FileAct.	Provided in the package To modify, add the Initial transfer parameters to the peltrans commands

Object	Description	Comments
ssp_pull	Launched by the Decision Rule INCOMING_SSP_IA_REQUEST to simulate a request to the SSP by using the Model FROM_BO_TO_SSP.	Provided in the package To modify, add the Initial transfer parameters to the peltrans commands

Multiple CRISTAL configurations

You can adapt the TARGET2 object configuration to support multiple JMS/CRISTAL back-end applications, each one working for a specific institution and using a specific BIC code and Local Trading Partner.

In this case, you use Decision Rules to filter incoming flows from TARGET2 depending on the Local Trading Partner alias. Decision Rules use Models to forward the messages to the respective Remote JMS Sites.

Requests coming from the different back-office JMS Sites can be forwarded to different SWIFTNet Remote Sites by duplicating the FROM_BO_TO_SSP and TG2DIR_GETFILE_SSP Decision Rules and their associated Models.

Multiple CRISTAL configuration: example 1

Remote Sites

- SWIFT1 - SWIFTNet
- CRISTAL_IN1 - JMS
- CRISTAL_OUT1 - JMS
- CRISTAL_ERR1 - JMS
- CRISTAL_FILE1 – PeSIT E

Trading Partner

- QT2_TP_LOCAL1

Models

- FROM_BO_TO_SSP1 - linked to the SWIFTNet Remote Site
- TG2DIR_GETFILE_SSP1 - linked to the SWIFTNet Remote Site
- TO_BO_QT2_FILE_OK1 - linked to CRISTAL_FILE1 Remote Site
- TO_BO_QT2_FILEMSG_KO1 - linked to CRISTAL_ERR1 Remote Site
- TO_BO_QT2_FILEMSG_KO_P1 - linked to CRISTAL_ERR1 Remote Site
- TO_BO_QT2_GETFILE_KO1 - linked to CRISTAL_ERR1 Remote Site
- TO_BO_QT2_OK1 - linked to CRISTAL_OUT1 Remote Site

Decision Rules

- FROM_BO_TO_SSP1 - Model FROM_BO_TO_SSP1
- INCOMING_GETFILE_SSP1 - batch call: tg2file TO_BO_QT2_FILE_OK1 TO_BO_QT2_FILEMSG_KO_P1 &(local_copy_directory1)
- INCOMING_GETFILE_SSP_KO1 - Model TO_BO_QT2_GETFILE_KO1
- INCOMING_SSP_IA_REQUEST – batch call: ssp_pull_s.sh
- INCOMING_SSP_IA_RESPONSE1 - batch call: tg2msg TO_BO_QT2_OK1 TO_BO_QT2_FILEMSG_KO_P1
- OUTGOING_SSP_IA_REQ_KO1 - Model TO_BO_QT2_FILEMSG_KO1
- TG2DIR_GETFILE_SSP1 - Model TG2DIR_GETFILE_SSP1

Multiple CRISTAL configuration: example 2

Remote Sites

- SWIFT2 - SWIFTNet
- CRISTAL_IN2 - JMS
- CRISTAL_OUT2 - JMS
- CRISTAL_ERR2 - JMS
- CRISTAL_FILE2 – PeSIT E

Trading Partner

- QT2_TP_LOCAL2

Models

- FROM_BO_TO_SSP2 - linked to the SWIFTNet Remote Site
- TG2DIR_GETFILE_SSP2 - linked to the SWIFTNet Remote Site
- TO_BO_QT2_FILE_OK2 - linked to CRISTAL_FILE2 Remote site
- TO_BO_QT2_FILEMSG_KO2 - linked to CRISTAL_ERR2 Remote Site
- TO_BO_QT2_FILEMSG_KO_P2 - linked to CRISTAL_ERR2 Remote Site
- TO_BO_QT2_GETFILE_KO2 - linked to CRISTAL_ERR2 Remote Site
- TO_BO_QT2_OK2 - linked to CRISTAL_OUT2 Remote Site

Decision Rules

- FROM_BO_TO_SSP2 - Model FROM_BO_TO_SSP2
- INCOMING_GETFILE_SSP2 - batch call: tg2file TO_BO_QT2_FILE_OK2 TO_BO_QT2_FILEMSG_KO_P2 &(local_copy_directory2)
- INCOMING_GETFILE_SSP_KO2 - Model TO_BO_QT2_GETFILE_KO2
- INCOMING_SSP_IA_REQUEST – batch call: ssp_pull_s.sh
- INCOMING_SSP_IA_RESPONSE2 - batch call: tg2msg TO_BO_QT2_OK2 TO_BO_QT2_FILEMSG_KO_P2
- OUTGOING_SSP_IA_REQ_KO2 - Model TO_BO_QT2_FILEMSG_KO2
- TG2DIR_GETFILE_SSP2 - Model TG2DIR_GETFILE_SSP2

Simulation mode message processing

The following diagram summarizes the simulation mode processing sequence.



Simulation mode message processing

The following diagram summarizes the simulation mode processing sequence.



Simulation mode configuration requirements

To configure Gateway for TARGET2 simulation mode (for example, to set up a test environment) you manage the following Gateway objects:

- Local Site
- Remote Site SWIFT
- Remote Trading Partner (for InterAct service)
- Remote Trading Partner (for FileAct service)
- Local Trading Partner
- Models
- Decision Rules

Configuring Gateway objects for simulation mode

If you have not already done so, import the Remote Sites, Trading Partners, Decision Rules and Models from the TARGET2 package and update the configuration file as described in:

- [Imported objects](#)
- [Editing TARGET2 files and objects](#)

After you import the objects, you must then configure them using either Gateway Navigator or the command line mode.

Configuring the TARGET2 Trading Partners

To set up the loop configuration, enter values for the parameters name, service, and request type in the following objects:

- QT2_TP_LOCAL
- QT2_TP_REMOTE_FILE_SSP
- QT2_TP_REMOTE_SPP

To make sure that the QT2_TP_REMOTE_SSP Partner is in pull-mode (receipt request and request receipt synchronized), execute the command:

```
peladm update_tradepart -a QT2_TP_REMOTE_SSP -rr signed -rrs Y
```

Configuring Decision Rules

Modify and configure the imported Decision Rules for routing to the back-end applications. All Decision Rules, if not specified otherwise, must be linked to an active Rule Table.

Attach the following Decision Rules to the default Rule Table. In each Decision Rule, modify the criteria trade service= trgt.* to trade service= *:

- INCOMING_GETFILE_SSP
- INCOMING_GETFILE_SSP_KO1
- INCOMING_SSP_IA_RESPONSE1
- OUTGOING_SSP_IA_REQ_KO1

Updating the configuration file

In the TARGET2 configuration file located at `../qt2/config`, set the execution mode to simulation (SIM):

```
set MODE=SIM
```

Transfers and message type

To run the TARGET2 module in simulation mode, a SWIFT loop configuration must be possible in which you send messages/files to yourself. Users must have access to a SWIFT InterAct service and a SWIFT identity that allows them to send InterAct to themselves.

The file sent to the simulation (InitialRequest.xml in the example) can have any content. The only requirements are:

- The file must be embedded:
 - File content must be contained between the tags `<SwInt:RequestPayload>` and `</SwInt:RequestPayload>`
 - File content must not conflict with XML parsing
- The file must be an XML file. For example, you could use the `InitialRequest.xml` file provided in the QT2 directory

Pre-built test cases

The Gateway TARGET2 package includes a set of pre-built test case files that you can use to test TARGET2 exchanges in either Real or Simulation mode.

You can launch test cases either from the GUI or via line commands.

The following test cases are provided to enable you to analyze your configuration:

Test case number	Case result
0000	Returns a technical final response, not compressed
0001	Returns a technical final response, compressed
1001	Returns the response "Execution timeout limit exceeded".
1002	Returns the response "Data not yet available".
1003	Returns the response "Message size limit exceeded". [Uncompressed file put at disposal.]

Test case number	Case result
1003b	Returns the response "Message size limit exceeded". [Compressed file put at disposal.]
1111	Returns a response with an unhandled SSP error code.
7777	Simulation will send a functional response, not compressed

Pre-built test cases

The Gateway TARGET2 package includes a set of pre-built test case files that you can use to test TARGET2 exchanges in either Real or Simulation mode.

You can launch test cases either from the GUI or via line commands.

The following test cases are provided to enable you to analyze your configuration:

Test case number	Case result
0000	Returns a technical final response, not compressed
0001	Returns a technical final response, compressed
1001	Returns the response "Execution timeout limit exceeded".
1002	Returns the response "Data not yet available".
1003	Returns the response "Message size limit exceeded". [Uncompressed file put at disposal.]
1003b	Returns the response "Message size limit exceeded". [Compressed file put at disposal.]
1111	Returns a response with an unhandled SSP error code.
7777	Simulation will send a functional response, not compressed

Launching tests via line commands

To launch a test case via a Gateway line command, use a command in the format of the following example:

```
.../QT2> peltrans -ml FROM_BO_TO_SSP -fc xml/InitialRequest.xml -param1 0000
```

Where:

- **-ml** is the name of the Model called to execute the test case
For the current set of test cases you always use the Model FROM_BO_TO_SSP.
- **-fc** is the path to the XML file that Gateway uses to generate the initial request to SSP.
Gateway requires the file `InitialRequest.xml` to generate the initial request for all test cases.
- **-param1** is the test case number
Select a test case number from the table in [Pre-built test cases](#).

Launching tests from the Gateway GUI

To launch a test case from the Gateway GUI you complete the following *New Transfer Request* screens:

General tab

SFE : New Transfer Request

SWIFTNet | JMS | OFTP | X.420

General | Attributes | Details | PeSIT | Etebac5 | HTTP/FTP | SMTP | Trading partner

General information

Model: FROM_BO_TO_SSP1

Type: INTERACT | Mode: Initiator | Direction: Send

Application: DEFAULT

Ack option: | Purge option: | Permanent:

Sites

☒ **Originator alias**: SFE | List Sites | List Lists

Destination alias: RSS | List Sites | List Lists

☐ **Local agent alias**: | List Sites

Remote agent alias: | List Sites

Originator:

Destination:

OK | Cancel | Help

In the **Model** field, enter the name of the model called by Gateway to execute the test case. For the current set of test cases you always use the Model FROM_BO_TO_SSP.

Attributes tab

SFE : New Transfer Request

SWIFTNet | JMS | OFTP | X.420

General | **Attributes** | Details | PeSIT | Etebac5 | HTTP/FTP | SMTP | Trading partner

Local attributes

File component:

Record format: Record length: Data code:

Transfer attributes

HTTP, FTP, SFTP only ☐

Directory name: Directory alias:

File name:

File type: (Pesit, Etebac5 only)

Record format: Record length: Data code:

Compression: Newline convention:

Format

Padding character (Hexa): Truncating allowed:

In the **File component** field, enter the path to the `InitialRequest.xml` file. By default, this file is located at: `C:\Axway\qt2\xml`

Details tab

SFE : New Transfer Request

SWIFTNet | JMS | OFTP | X.420

General | Attributes | **Details** | PeSIT | Etebac5 | HTTP/FTP | SMTP | Trading partner

User

Param 1: 0000

Param 2:

Message to send:

Retry and Timing

Interval:

Maximum number of retries: 10

Priority: Normal

Frozen state:

Earliest start date

Date: 22/10/2008

Time: 17:13:19

Latest start date

Date: 22/10/2008

Time: 17:13:19

Routing

Route from (Transfer ID):

Reply

Reply to (Transfer ID):

Sentinel

Transfer filter: Undefined

OK Cancel Help

In the **Param 1** field, enter the number of the test case you want to run. Select a test case from the table provided in [Pre-built test cases](#).

SWIFTNet tab

SFE : New Transfer Request

GeneralAttributesDetailsPeSITEtebac5HTTP/FTPSMTPTrading partner

SWIFTNetJMSOFTPX.420

Destination partnerQT2_TP_REMOTE_SSP

Delivery modeUndefined

Destination DN

Compress(empty)

Originator partnerQT2_TP_LOCAL1

Originator DN

Service name

Request type

ReceiptE2ESecurityOtherInterAct

Receipt requestUndefined

Receipt DN

Receipt request type

OK

Cancel

Help

In the **Destination partner** and **Originator partner** fields, enter the names of the appropriate Remote Sites.

Analyzing test results

To confirm the success or failure of the test transfer, first check the Mailbox screen in Gateway Navigator.

Transfers													
Loc...	Application	Type	State		Direct...	Mode	Origin...	Destination	File component	Para...	Model	Protocol	Row
797	DEFAULT_A	MSG	Ended (E)	5	Send	Initiator	PESIT	RSS	InitialRequest.xml	0001	FROM_BO_TO_SSP1	SWIFTNet	Not
798	FTP_B	MSG	Acknowledged (A)2		Receive	Responder	RSS	PESIT	F0000798	0001		SWIFTNet	
799	FTP_B	MSG	Ended (E)	5	Send	Initiator	PESIT	RSS	ResponseCompressed...	0001		SWIFTNet	Not
800	FTP_B	RECEIPT	Ended (E)	5	Send	Initiator	PESIT	RSS	ResponseCompressed...			SWIFTNet	Not
801	FTP_B	MSG	Ended (E)	7	Receive	Responder	RSS	PESIT	F0000801.dcm	0001		SWIFTNet	Rou
802	DEFAULT	TRANS	Ended (E)	5	Send	Initiator	PESIT	CRISTAL_OUT1	F0000801.dcm		TO_BO_QT2_OK1	JMS	Not

In the preceding example, the initial test case transfer is 797.

For detailed results of the test case transfers, open the trace file for the transfer in the directory:

```
[PATH]\Synchrony_V4.x.x\qt2\tmp
```

Trace file numbers correspond to the Local TransferID number. For the transfer 797 displayed in the Mailbox screen, the corresponding trace file is 797.trace.

The following is an example of a trace file 111.trace result for a 0001 type test case.

Gateway unknown. Assuming 6112.

+++++ TRACE from /swiftrdq/users/swiftrdq/Axway/Synchrony_
V4.1.1/qt2/scripts/tg2msg.sh TO_BO_QT2_OK1 TO_BO_QT2_FILEMSG_KO_P1

InterAct Response received from the SSP. Begin Processing

Tuesday, October 14, 2008 6:05:31 PM CEST

Original Request ID: 797

Current Transfer ID: 801

QT2ROOT=/swiftrdq/users/swiftrdq/Axway/Synchrony_V4.1.1/qt2

GATEWAY=6112

MODE=SIM

FULLCMPENGINE=TRUE

MODEL_OK=TO_BO_QT2_OK1

MODEL_ERR=TO_BO_QT2_FILEMSG_KO_P1

QT2_TP_REMOTE_FILE_SSP=QT2_TP_REMOTE_FILE_SSP

RQT_GPD=camt.998.001.01

P_TRY_NO=3

P_TRY_DELAY=5

VCAMT=3

Setting Variables

Start Unpacking Section

x_ftp_command=dt="ZIP-Base64"

No decompression necessary

File to be Xalan-ed to extract the technical information, if any

~~~~~ TGOUT=/swiftrdq/users/swiftrdq/Axway/Synchrony\_

V4.1.1/qt2/tmp/801.TGOUT ~~~~~

<SwInt:Gateway xmlns:SwInt="urn:swift:snl:ns.SwInt">

<SwInt:ResponsePayload dt="ZIP-Base64">

<AppHdr xmlns="urn:swift:xsd:\$ahV10">

<From>

<Type>NAME</Type>

<Id>PM</Id>

</From>

<MsgRef>ABCDEFGHJKLMNOPQRST1234567890</MsgRef>

<CrDate>2007-01-01T12:00:00</CrDate>

</AppHdr>

<Document xmlns="ssp.pm.ReturnLimit\$camt.010.001.03">

<camt.010.001.03>

<MsgId>

<Id>ABC123456</Id>

---

```
</MsgId>
<BizQryRef>
  <QryRef>ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghi</QryRef>
</BizQryRef>
<BizRpt>
  <CurLmt>
    <LmtId>
      <BilLmtCtrPtyId>
        <BIC>BANKNL2AXXX</BIC>
      </BilLmtCtrPtyId>
      <Tp>BILI</Tp>
      <AcctOwnr>BANKNL2AXXX</AcctOwnr>
    </LmtId>
    <Lmt>
      <Amt>
        <AmtWthtCcy>123456.12</AmtWthtCcy>
      </Amt>
      <CdtDbtInd>DBIT</CdtDbtInd>
    </Lmt>
  </CurLmt>
  <DfltLmt>
    <LmtId>
      <BilLmtCtrPtyId>
        <BIC>BANKNL2AXXX</BIC>
      </BilLmtCtrPtyId>
      <Tp>MULT</Tp>
      <AcctOwnr>BANKNL2AXXX</AcctOwnr>
    </LmtId>
    <Lmt>
      <Amt>
        <AmtWthtCcy>123456.12</AmtWthtCcy>
      </Amt>
      <CdtDbtInd>DBIT</CdtDbtInd>
    </Lmt>
  </DfltLmt>
</BizRpt>
</camt.010.001.03>
</Document>
</SwInt:ResponsePayload>
</SwInt:Gateway>
~~~~~ TGOUT=/swiftreq/users/swiftrdq/Axway/Synchrony_
V4.1.1/qt2/tmp/801.TGOUT ~~~~~
```

---

---

```
TG2_CODE=
TG2_DESC=Document received.
TG2_ALERT=FALSE
TG2_REF=ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghi
TG2_ID=ABC123456
Making Decision
Unknown status code. Calling RoutingData with ERROR_IA=FALSE
ENTER: routingdata
Variable ERROR_IA=FALSE
ENTER: dotrans
Try 1 of 3
peltrans -ml TO_BO_QT2_OK1 -fc /home/swiftrdq/Axway/Synchrony_
V4.1.1/Gateway_V6.12.1/run_time/tmp/F0000801.dcm -rfx 801 -prop
LocalPartner=QT2_TP_LOCAL1 -prop RemotePartner=QT2_TP_REMOTE_SSP -
prop RequestType=camt.998.001.02 -prop Service=swift.generic.ia!x -prop
RequestRef=0001
LEAVE: dotrans
LEAVE: routingdata
Tuesday, October 14, 2008 6:05:32 PM CEST
----- END TRACE from tg2msg
```

---

Axway Gateway: Customizing Gateway Processes

# Customizing Gateway processes

The following Gateway features and utilities enable you to customize and extend the standard processes provided by the product:

- [User exits](#)
- [C API](#)
- [Code conversion and File formatting](#)

Axway Gateway: Customizing Gateway Processes

## User exits

**Caution** Please note that care should be taken when **customizing user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

[User exits](#)

[Internal and External exits](#)

[Exit families](#)

[Exit-related files](#)

[Exit management overview](#)

## About User exits

A user exit is a point in a software program at which a user-developed program can take control of program processes.

Axway provides a set of C-language user exits with Gateway. You can find them in `<Gateway installation directory>/src`.

The exits are triggered by different types of Gateway processing event. They enable you to integrate your own custom processing routines with Gateway standard processing activity.

## Internal and External exits

Gateway provides two basic categories of user exit: internal and external.

### *Internal exits*

Internal user exits are routines that Gateway invokes directly in synchronous calls from normal internal processes. The internal process waits for the result of the exit call before continuing with the programmed series of processing events. Internal exit implementation has the following limitations:

- Gateway reliability can be affected by unexpected behavior of user exit code
- C user API routines cannot be called from user exit code
- Any load (CPU, IO, memory), overhead or latency due to user exit code may affect performances and trigger time-outs, due to file transfer and user exit code sharing monitor processes and related resources

You do not have to perform any particular procedure to activate internal exits in Gateway. Internal exit trigger points are automatically active in the processing code for each protocol.

### *External exits*

External user exits are routines that Gateway invokes in *asynchronous* calls from the normal internal processes. Gateway sends a notification message to a separate set of user processes called *notif* processes. On reception of the notification message, *notif* calls the user exit routine corresponding to the event.

Contrary to the case for internal exits, Gateway continues other tasks while *notif* manages exit processes. On reception of the expected response, Gateway re-synchronizes transfer processing from information saved in the associated exit context.

One set of *notif* processes handles a single exit routine at a given time. Several sets of *notif* processes can be active simultaneously, handling several exit routines.

### *How notif operates*

*Notif* uses a request/response protocol to communicate with the Gateway Supervisor process. This protocol is encapsulated into a notification API that is similar to the standard C user API. The notification API is not directly available to user applications.

A state engine based on the notification API gets notification events from the Gateway supervisor process and schedules exit routine calls according to event descriptors. This engine contains the main program entry point of executable files linked with external user exit routines.

In order to enable user applications based on exit routines to easily follow protocol exchanges, all notification messages related to a protocol session instance are sent to the same user notification process.

## Exit families

User exit routines are classified into exit families.

The exit families delivered with Gateway are:

|       |                                                                                                                    |
|-------|--------------------------------------------------------------------------------------------------------------------|
| INIT  | Initialization and end of the user notification routines                                                           |
| FT    | File Transfer protocol events: protocol session establishment/ release, file transfer beginning/ completion/ abort |
| XFER  | Transfer request state transition                                                                                  |
| SITE  | Dynamic Site state transition                                                                                      |
| LOG   | User selected log message                                                                                          |
| SECS  | TLS and SSH handshake events                                                                                       |
| EVENT | Scheduling and File Storage Rule Tables for Event Management                                                       |

## Exit-related files

To process an exit, Gateway reads from two files that provide information for the exit processes: the *header file* and the *source file*.

### *Header file*

The header file describes the structure of the information that the exit sends as output or returns as input to the Gateway process. A header file contains separate sections for each exit routine in a given exit family.

Header files have names with `.h` file extensions, for example, `exitsecu.h`.

Header files are located in the `<Gateway installation directory>/inc` directory.

*Do not modify the contents of this file.*

### *Source file*

The source file contains the customized user code that enables you to obtain the return values you require during exit processing.

The source files have names with `.c` file extensions, for example, `exitsecu.c`.

Gateway provides source file models in which you can insert your own code. You can find these models in the `<Gateway installation directory>/src` directory.

**Do not** move exit header files or exit source files from the directories where they are installed.

## Exit management overview

To manage a user exit:

1. Execute the `profile` batch file to set the environment variables for your platform.
- For UNIX: execute the `profile` batch file located in `<Gateway installation directory>/run_time/etc`
- For Windows: execute the `profile.bat` batch file located in `<Gateway installation directory>\run_time\etc`
2. Edit the source files that correspond to your exits, inserting the required code.  
After modifying a source file you must [compile the file](#).
3. For security-related exits:
  - Activate the `secs_exit_scheduling` option (in the [advanced options](#) of the configuration menu).
  - For TLS, activate the `exit_scheduling` parameter of the TLS Security Profile.

[Configuration: Gateway parameters](#)

[Internal user exits](#)

[User exits and Rule Tables](#)

[User exits in Perl](#)

[External user exits](#)

[User exits: Security functions](#)

[TLS user exits](#)

[SSH user exits](#)

[Compiling C user exits](#)

[SWIFTNet-specific user exits](#)

Axway Gateway: Customizing Gateway Processes

## Internal user exits

**Caution** Please note that care should be taken when **customizing user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

[Internal user exits](#)

[List of internal user exits](#)

[Details of internal user exits](#)

## Internal user exits

This topic describes the internal user exits supplied with Gateway.

Internal exits are implemented via synchronous calls from the Gateway Supervisor module processes to user routines defined in exit source files.

### Exit file locations

| File (extension)       | Location                                                   |
|------------------------|------------------------------------------------------------|
| Exit header files (.h) | <Gateway installation directory>/inc                       |
| Exit source files (.c) | sub-directories in<br><Gateway installation directory>/src |

## List of internal user exits

The following table lists the available internal exits and their associated header and source files.

Follow the links for details about using the exits.

| Access Files                 | User exit                                   | Called...                                                               |
|------------------------------|---------------------------------------------|-------------------------------------------------------------------------|
| exitsecu.c and<br>exitsecu.h | <a href="#">ExitCheckUserLogin</a>          | When a user logs on to Gateway                                          |
|                              | <a href="#">ExitCheckUserLogout</a>         | When a user logs out of Gateway                                         |
|                              | <a href="#">ExitCheckUserAccessOnObject</a> | When a user accesses a controlled resource                              |
|                              | <a href="#">ExitCheckFileAccess</a>         | When a user requests a file transfer                                    |
|                              | <a href="#">ExitCheckUserAccess</a>         | When a user requests access to a protected resource                     |
| exitcs.c and<br>exituser.h   | <a href="#">ExitCSLogin</a>                 | When a user attempts to connect to Gateway from a remote client station |



| Access Files                    | User exit                              | Called...                                                                                                                                                                                                                      |
|---------------------------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| exitpsit.c and<br>exituser.h    | <a href="#">ExitPesitMsg</a>           | On reception of FPDU_<br>MSG (PeSIT message<br>user exit routine)                                                                                                                                                              |
|                                 | <a href="#">ExitPesitPreConnection</a> | On receiving a PeSIT<br>pre-connection<br>message (PeSIT pre-<br>connection user exit<br>routine)                                                                                                                              |
|                                 | <a href="#">ExitPesitGetAppli</a>      | On receiving an<br>incoming PeSIT<br>Transfer Request                                                                                                                                                                          |
| et3exitc.c and<br>et3publi.h    | <a href="#">Et3EncodeParameters</a>    | In Responder/Sender<br>mode, when<br>representing a bank<br>exchange partner,<br>Gateway calls the<br>Et3EncodeParameters<br>exit function when it<br>receives the ETEBAC 3<br>parameter card.                                 |
| et3exitb.c<br>and<br>et3publi.h | <a href="#">Et3DecodeParameters</a>    | In Initiator/Sender<br>mode, when<br>representing the client<br>exchange partner,<br>Gateway calls the<br>Et3DecodeParameters<br>exit function with the<br>Gateway parameter<br>card corresponding to<br>the Transfer Request. |
| et3exitf.c<br>and<br>et3exitf.h | <a href="#">Et3ExitFProcessing</a>     | In Responder/Sender<br>mode, Gateway calls the<br>Et3ExitFProcessing exit<br>function on file<br>reception.                                                                                                                    |

| Access Files                    | User exit                          | Called...                                                            |
|---------------------------------|------------------------------------|----------------------------------------------------------------------|
| et5exitb.c<br>and<br>exituser.h | <a href="#">Et5ExitConnectInd</a>  | When Gateway operates in server mode and a CONNECT fpdu is received. |
|                                 | <a href="#">Et5ExitCreateInd</a>   | When Gateway operates in server mode and a CREATE fpdu is received   |
|                                 | <a href="#">Et5ExitDeselectInd</a> | When Gateway operates in server mode and a DESELECT fpdu is received |
|                                 | <a href="#">Et5ExitDtfEndInd</a>   | When Gateway operates in server mode and a DTF.END fpdu is received  |
|                                 | <a href="#">Et5ExitSelectInd</a>   | When Gateway operates in server mode and a SELECT fpdu is received   |
|                                 | <a href="#">Et5ExitTransEndInd</a> | When a TRANS.END fpdu is received                                    |
|                                 | <a href="#">Et5ExitOpenInd</a>     | When an OPEN fpdu is received                                        |

| Access Files                                  | User exit                            | Called...                                                                                    |
|-----------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------|
| <b>exitsds.c</b><br>and<br><b>exitsds.h</b>   | <a href="#">ExitSdsMemo</a>          | On transfer creation                                                                         |
|                                               | <a href="#">ExitSdsLogin</a>         | When a client connects to an SD5 server                                                      |
|                                               | <a href="#">ExitSdsSelect</a>        | When an SD5 server locates detects a selectable transfer                                     |
|                                               | <a href="#">ExitSdsSigPar</a>        | When a client requests a signature or paraphe (contextual authorization)                     |
|                                               | <a href="#">ExitSdsCancel</a>        | The SD5 server calls ExitSdsCancel on each cancel request originating at the client machine. |
| <b>exitodt.c</b><br>and<br><b>exituser.h</b>  | <a href="#">ExitOdtGetAppli</a>      | To retrieve a Gateway Application object with OFTP (Odette) protocol                         |
| <b>exitpop3.c</b><br>and<br><b>exituser.h</b> | <a href="#">ExitPop3GetAppli</a>     | To retrieve a Gateway Application object with POP3 protocol                                  |
| <b>exitftp.c</b><br>and<br><b>exituser.h</b>  | <a href="#">ExitFtpGetAppli</a>      | To retrieve a Gateway Application object with FTP protocol                                   |
|                                               | <a href="#">ExitFtpSetReturnCode</a> | For each FTP command received (except for <b>USER</b> , <b>PASS</b> and <b>SYST</b> )        |
|                                               | <a href="#">ExitFtpRespPasy</a>      | In Responder mode before formatting and sending <b>PASV</b> response                         |
|                                               | <a href="#">ExitFtpCmdPort</a>       | In Initiator mode before formatting and sending <b>PORT</b> command                          |

| Access Files                                  | User exit                                       | Called...                                                                                                                                                                  |
|-----------------------------------------------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>exithttp.c</b><br>and<br><b>exithttp.h</b> | <a href="#">ExitHttpGetAppli</a>                | In server mode, on reception of a PUT or POST HTTP request.                                                                                                                |
|                                               | <a href="#">ExitHttpPutBodyResponseReceived</a> | In client mode, on reception of a PUT or POST request response. Gateway calls the exit after sending the requested file, before it sends the end-of-transfer notification. |
|                                               | <a href="#">ExitHttpRequestReceived2</a>        | In server mode, on reception of a request, before processing CGate and pre-connection exits.                                                                               |
| <b>exitlog.c</b><br>and<br><b>exituser.h</b>  | <a href="#">ExitLogArchived</a>                 | When the log file is archived                                                                                                                                              |
|                                               | <a href="#">ExitLogMessage</a>                  | Before a record is written in the log file                                                                                                                                 |
| <b>exitstat.c</b><br>and<br><b>exituser.h</b> | <a href="#">ExitStatMessage</a>                 | Before a statistics record is formatted and written in the Gateway stat file                                                                                               |
| <b>exitstat.c</b><br>and<br><b>exitstat.h</b> | <a href="#">ExitStatArchived</a>                | When the statistic file is archived                                                                                                                                        |
| <b>exitlgin.c</b><br>and<br><b>exitclnt.h</b> | <a href="#">ExitLoginParam</a>                  | Before a user logs in to Gateway. It is executed in the current user process context and is intended for the user to supply the (default) login parameters.                |
| <b>exitckrq.c</b><br>and<br><b>exitclnt.h</b> | <a href="#">ExitCheckXferRequest</a>            | Before accessing a file (for a Transfer Request)                                                                                                                           |

| Access Files                                            | User exit                                 | Called...                                                  |
|---------------------------------------------------------|-------------------------------------------|------------------------------------------------------------|
| <code>exitfs.c</code><br>and<br><code>exitrte.h</code>  | <a href="#">ExitAfterFileStoringEvent</a> | Before scanning a<br>Directory Scanning type<br>Rule Table |
| <code>exit temp</code><br>and<br><code>exitrte.h</code> | <a href="#">ExitAfterTemporalEvent</a>    | Before scanning a<br>Scheduling type Rule<br>Table         |

## Details of internal user exits

### Supply login parameters: *ExitLoginParam*

#### Purpose

This exit is called before you log in to Gateway. It is executed in your process context, and enables you to supply the (default) login parameters.

#### Syntax

```
void ExitLoginParam (char *user_name, int szName, char *password,
int szPass, char *newpassword, int szNPass)
```

| ExitLoginParam    |                          |                                                |
|-------------------|--------------------------|------------------------------------------------|
| Input parameters  | <code>user_name</code>   | Login name for the user                        |
|                   | <code>szName</code>      | <code>user_name</code> buffer size             |
|                   | <code>password</code>    | Login password for that <code>user_name</code> |
|                   | <code>szPass</code>      | Password buffer size                           |
|                   | <code>newpassword</code> | New login password                             |
|                   | <code>szNPass</code>     | New password buffer size                       |
| Output parameters | <code>user_name</code>   | Login name                                     |
|                   | <code>password</code>    | Login password for that <code>user_name</code> |
|                   | <code>newpassword</code> | New login password                             |
| Return            | None                     |                                                |

## Login: *ExitCheckUserLogin*

### Purpose

This exit is called after `ExitLoginParam` when you log in to Gateway. `ExitCheckUserLogin` validates the login request parameters.

### Syntax

```
int ExitCheckUserLogin (void **login_token, char *user_name, char
*password)
```

| ExitCheckUserLogin |                           |                                         |
|--------------------|---------------------------|-----------------------------------------|
| Input parameters   | <code>login_token</code>  | Pointer to context for user             |
|                    | <code>user_name</code>    | User login name                         |
|                    | <code>password</code>     | User login password                     |
| Output parameters  | <code>login_token</code>  | Pointer to context for user             |
| Return             | <code>E_UUNDEFINED</code> | Continue with Gateway security checking |
|                    | <code>E_UDENIED</code>    | Login denied by user routine            |
|                    | <code>E_UALLOWED</code>   | Login accepted by user routine          |

## Logout: *ExitCheckUserLogout*

### Purpose

This exit is called when you log out of Gateway. It allows you to release the user context and liberate allocated space.

### Syntax

```
int ExitCheckUserLogout (void *login_token)
```

| ExitCheckUserLogout |                          |                             |
|---------------------|--------------------------|-----------------------------|
| Input parameters    | <code>login_token</code> | Pointer to context for user |

| ExitCheckUserLogout |              |                                              |
|---------------------|--------------|----------------------------------------------|
| Output parameters   | None         |                                              |
| Return              | E_UUNDEFINED | Return code from the exit routine is ignored |

### *Access resource: ExitCheckUserAccessOnObject*

#### **Purpose**

This exit is called when a user accesses a controlled resource, that is, a Gateway object.

Before the user performs a read, write, update, delete or admin operation on a specific Gateway object, such as a Site or an Application, this exit is called to check that the user has the required access rights to the object in question.

#### **Syntax**

```
int ExitCheckUserAccessOnObject (void *login_token, int resource, int
access ExitObjectRecord_t *object)
```

| ExitCheckUserAccessOnObject |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters            | login_token               | Pointer to context for user                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|                             | resource                  | Resource ID (refer to <code>exitsecu.h</code> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                             | access                    | Requested access type (refer to <code>exitsecu.h</code> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                             | object                    | <p>Data structure that describes the attributes for the object. This data structure is read only and contains object attributes that identify a unique object instance.</p> <p>Note that *object is null when:</p> <ul style="list-style-type: none"> <li>• The resource is USER_LOGFILE</li> <li>• A user access request affects a set of objects rather than a specific instance. For example <code>Select()</code>, <code>OpenSet()</code>, <code>CloseSet()</code>, <code>purge()</code>.</li> </ul> <p>When <b>*object</b> is null, return <code>E_UUNDEFINED</code> to pass access control to Gateway.</p> |
| Output parameters           | None                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Return                      | <code>E_UUNDEFINED</code> | Continue with Gateway security checking                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|                             | <code>E_UDENIED</code>    | Access denied by user routine                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|                             | <code>E_UALLOWED</code>   | Access accepted by user routine                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

## Request file transfer: *ExitCheckFileAccess*

### Purpose

When a user requests a file transfer, this exit is called to validate access to the required file. If the exit routine denies access, the Transfer Request is rejected. Otherwise, it is accepted.

The default action in the supplied sample codes always accepts it.

### Syntax

```
int ExitCheckFileAccess (void *login_token, char *file, int access)
```



| ExitCheckFileAccess |                           |                                                           |
|---------------------|---------------------------|-----------------------------------------------------------|
| Input parameters    | <code>login_token</code>  | Pointer to context for user                               |
|                     | <code>file</code>         | File to be transferred                                    |
|                     | <code>access</code>       | Requested access type (refer to <code>exitsecu.h</code> ) |
| Output parameters   | None                      |                                                           |
| Return              | <code>E_UUNDEFINED</code> | Continue with Gateway security checking                   |
|                     | <code>E_UDENIED</code>    | Access denied by user routine                             |
|                     | <code>E_UALLOWED</code>   | Access accepted by user routine                           |

## Access: *ExitCheckUserAccess*

### Purpose

When a user accesses (via read/write/update/delete/admin) a protected resource (Site, Application, Transfer Request, User Record, Access Rights Record, Model, and so on), Gateway calls `ExitCheckUserAccess()` to validate the access request. If the exit routine accepts the request, the access is honored. If the exit routine denies the request, the access is rejected. If the exit routine doesn't care (default action in the supplied sample codes), Gateway continues the validation with its own user database definition.

The request is accepted if, and only if, at least one of the following conditions exists:

- the user is an administrator, or
- the required access on the resource is granted in the associated access rights record

If the required resource is a transfer request, the user must either be the request creator or have the *access all transfers* privilege.

### Syntax

```
int ExitCheckUserAccess (void *login_token, int resource, int access)
```

| ExitCheckUserAccess |              |                                         |
|---------------------|--------------|-----------------------------------------|
| Input parameters    | login_token  | Pointer to user context                 |
|                     | resource     | Resource ID                             |
|                     | access       | Requested access type                   |
| Return              | E_UUNDEFINED | Continue with Gateway security checking |
|                     | E_UDENIED    | Access denied by user routine           |
|                     | E_UALLOWED   | Access accepted by user routine         |

### *Remote client login: ExitCSLogin*

#### **Purpose**

This exit enables you to check the identification of a user who attempts to connect to Gateway from a remote client station.

This exit function is linked with the client station server process.

#### **Syntax**

```
int ExitCSLogin(char * username, char * password)
```

| ExitCSLogin       |             |                                                                    |
|-------------------|-------------|--------------------------------------------------------------------|
| Input parameters  | user_name   | User login name                                                    |
|                   | password    | User login password                                                |
| Output parameters | login_token | Pointer to context for user                                        |
| Return            | 0           | The user is authorized                                             |
|                   | 1           | The password is not recognized. The connection request is rejected |

## Receive PeSIT message: *ExitPesitMsg*

### Purpose

This user exit routine is called on reception of an FPDU\_MSG. On return, FPDU\_ACKMSG is sent to the partner Site with diagnostic code (PI 2) equal to zero.

**Note:** The received `msg` field is converted into the local machine character set before passing to this EXIT.

### Syntax

```
int ExitPesitMsg (char *origin, char *destination, int file_type, char
*file_name, int xfer_id, unsigned char *msg, int len)
```

| ExitPesitMsg      |                                                   |               |
|-------------------|---------------------------------------------------|---------------|
| Input parameters  | origin                                            | PI 3          |
|                   | destination                                       | PI 4          |
|                   | file type                                         | PI 11         |
|                   | filename                                          | PI 12         |
|                   | transfer ID                                       | PI 13         |
|                   | msg                                               | PI 91         |
|                   | len                                               | length of msg |
| Output parameters | None                                              |               |
| Return            | The return code from the exit routine is ignored. |               |

## Pre-connect PeSIT: *ExitPesitPreConnection*

### Purpose

PeSIT pre-connection user exit routine. On reception of a PeSIT pre-connection message, this exit is called to validate the message content (user ID and password).

- If the return code is 1, the request is accepted and a PeSIT ACK message is sent.
- If the return code is -1, the request is rejected and a PeSIT NAK message is sent.

Other return code values are reserved for future use.

- If accepted, the "requestor" field can also be filled to indicate the expected requestor ID in the incoming FPDU\_CONNECT. In this case, the requestor ID (PI 3) in the incoming FPDU\_CONNECT must match the specified value.
- Otherwise the FPDU\_CONNECT is rejected with a diagnostic = 304 (unauthorized requestor ID). If the "requestor" is not filled, the PI 3 value is validated depending on the Gateway Site definition.

**Note:** In this exit, all character fields are converted to the character set for the local machine.

### Syntax

```
int ExitPesitPreConnection (char *userID, char *password, char
*requestor, int bufisz)
```

| ExitPesitPreConnection |           |                                                   |
|------------------------|-----------|---------------------------------------------------|
| Input parameters       | userID    | User ID from the received pre-connection message  |
|                        | password  | Password from the received pre-connection message |
|                        | requestor | Associated requestor ID to validate               |
|                        | bufisz    | Buffer size of "requestor" field                  |
| Output parameters      | None      |                                                   |
| Return                 | 1         | Accept the pre-connection request                 |
|                        | -1        | Refuse the pre-connection request                 |

### *Get Application object for PeSIT: ExitPesitGetAppli*

#### Purpose

This user exit routine is called when Gateway receives a Transfer Request. It enables users to determine the application to use, based on the protocol elements received.

#### Syntax

```
Int ExitPesitGetAppli (const char *origin, const char *destination,
const char *file_type, const char *file_name, const char *xfer_id, const
char *file_label, const char *msg, char *application, int len)
```

[illegible]

[illegible]

[illegible]

## *Get Application object for OFTP: ExitOdtGetAppli*

### **Purpose**

Use this OFTP protocol exit routine to retrieve the Gateway Application object to use for the file transfer. This exit allows you to use Gateway Application objects even if both fields of the SFID command are already being used for other purposes.

Gateway calls this exit after receiving an SFID command. The SFID fields are the arguments used to search for the Application name.

The results are stored in the `buffer` parameter whose size is `len` bytes. The Gateway internal Application allows you to receive any file. Application characteristics include:

- You cannot delete or modify this Application
- No code conversion or record padding is possible with this Application

### **Syntax**

```
int ExitOdtGetAppli (char *dest_alias, char *filename, time_t date, char
*user_field, char *destination, char *originator, char *buffer, int len)
```

| ExitOdtGetAppli   |                          |                                                                           |
|-------------------|--------------------------|---------------------------------------------------------------------------|
| Input parameters  | <code>dest_alias</code>  | Remote Site of connection                                                 |
|                   | <code>filename</code>    | File name set in SFID FPDU                                                |
|                   | <code>date</code>        | Date and time field of SFID FPDU, converted to <code>time_t</code> format |
|                   | <code>user_field</code>  | Content of user field in SFID FPDU                                        |
|                   | <code>destination</code> | Alias name of final destination                                           |
|                   | <code>originator</code>  | Alias name of origin                                                      |
|                   | <code>len</code>         | Max length of buffer parameter                                            |
| Output parameters | <code>buffer</code>      | Buffer to store Application name                                          |



| ExitOdtGetAppli        |                                                                                                                                                               |                                                          |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| Return                 | 0                                                                                                                                                             | Gateway Application found and stored in buffer parameter |
|                        | 1                                                                                                                                                             | No Application found: use internal Application           |
|                        | 2                                                                                                                                                             | No Application found: Transfer Request is rejected       |
| Example                | The sample function ( <code>exitodt.c</code> ) provides one example that uses a matching table. You can customize this example to suit your own requirements. |                                                          |
| Additional information | For more information on OFTP, refer to <a href="#">About OFTP</a> .                                                                                           |                                                          |

## Get Application object for FTP: *ExitFtpGetAppli*

### Purpose

This exit is called to retrieve the Application if the Remote Site parameter **Application method** (`get_appli_method`) is set to *User exit (AIE)*.

Gateway calls this exit immediately after receiving an FTP `STOR` or `STOU` command. The result must be stored in the `buffer` parameter whose size is `len` bytes.

### Syntax

```
int ExitFtpGetAppli(
 char *dest_alias,
 char *filename,
 short data_code,
 char *appli_name,
 char *destination,
 char *originator,
 char *buffer,
 int len)
```

| ExitFtpGetAppli        |                                                                                                                                                                                   |                                                                      |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Input parameters       | <code>dest_alias</code>                                                                                                                                                           | Remote Site of connection                                            |
|                        | <code>filename</code>                                                                                                                                                             | FTP filename                                                         |
|                        | <code>short_data_code</code>                                                                                                                                                      | Transfer data code                                                   |
|                        | <code>appli_name</code>                                                                                                                                                           | Content of <code>SITE P_APPLI</code> parameter                       |
|                        | <code>destination</code>                                                                                                                                                          | Alias of final destination                                           |
|                        | <code>originator</code>                                                                                                                                                           | Alias of origin                                                      |
|                        | <code>len</code>                                                                                                                                                                  | Max length of <code>buffer</code> parameter                          |
| Output parameters      | <code>buffer</code>                                                                                                                                                               | Buffer to store Application name                                     |
| Return                 | 0                                                                                                                                                                                 | Application is found and stored in the <code>buffer</code> parameter |
|                        | 1                                                                                                                                                                                 | No Application found and default FTP internal Application used       |
|                        | 2                                                                                                                                                                                 | No Application found and Transfer rejected                           |
| Example                | The sample function ( <code>exitftp.c</code> ) provides an example using a matching table from data code of transfer and destination alias name. Any other method can be applied. |                                                                      |
| Additional information | For more information on FTP, refer to <a href="#">About FTP</a> .                                                                                                                 |                                                                      |

## Customize FTP: ExitFtpSetReturnCode

### Purpose

This exit allows customized actions on reception of an FTP command on the server side. It is called for each FTP command received (except for `USER`, `PASS` and `SYST`). Use the exit function to:

- Do nothing: Gateway continues processing as usual.
- Supply a return code and message: Gateway skips the command processing and responds immediately with the code and message supplied by the exit. This feature is useful for two reasons:

- It simulates a positive completion return code for commands that are not supported.
- It forces a negative completion return code for commands that are not accepted.

Syntax

```
int ExitFtpSetReturnCode(

 int ftp_ref, /* current ftp reference */

 char *ftp_cmd, /* current ftp command */

 int *ftp_code, /* code to be returned */

 char *ftp_msg) /* message to return */
```

| ExitFtpSetReturnCode |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters     | ftp_ref                       | The reference for the current control connection for the command.                                                                                                                                                                                                                                                                                                                                                                                     |
|                      | ftp_cmd                       | The complete FTP command received, including all parameters except CRLF.                                                                                                                                                                                                                                                                                                                                                                              |
| Output parameters    | ftp_code                      | <p>This parameter is mandatory. It is the code that must be returned in response to this command.</p> <p>If the ftp_code is set to 0, the exit function does nothing and Gateway processes the command as usual.</p> <p>If the ftp_code is set to a non-0 value, the command is not processed and Gateway immediately responds with this ftp_code.</p>                                                                                                |
|                      | ftp_msg                       | <p>This parameter is optional and is only used if the returned ftp_code value is not 0.</p> <p>If ftp_code is a non-0 value, this is the message sent. If the ftp_code is not 0 and the ftp_msg contains no message (strlen(ftp_msg) is 0), Gateway uses the standard message associated with the ftp_code.</p> <p>This parameter should not be used unless there is a very specific reason to do so. The message must not exceed 200 characters.</p> |
| Return               | The return value is always 0. |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

### ExitFtpSetReturnCode

Additional  
information

For more information on FTP, refer to [About FTP](#).

#### ExitFtpSetReturnCode default code

This is the code supplied by default with Gateway. It does nothing and lets Gateway behave as if there were no exit function.

```
int ExitFtpSetReturnCode(int ftp_ref, char *ftp_cmd, int *ftp_code, char
*ftp_msg){ *ftp_code = 0; ftp_msg[0] = 0;if (*ftp_code > 0) LogPrintf
(LOG_INFO, "EXITFTP (%d) reply code set to %d for \ command \"%s\"\\n",
ftp_ref, *ftp_code, ftp_cmd);return(0);}
```

#### Example 1

##### Simulating a positive completion response

This example shows how to simulate a positive completion for commands that are not implemented in Gateway.

```
int ExitFtpSetReturnCode (int ftp_ref, char *ftp_cmd, int *ftp_code, char
*ftp_msg){ *ftp_code = 0; ftp_msg[0] = 0; if (memcmp(ftp_cmd, MKD, 3)
== 0) *ftp_code = 257; else if (memcmp(ftp_cmd, RMD, 3) == 0) *ftp_
code = 250; else if (memcmp(ftp_cmd, CWD, 3) == 0) *ftp_code = 250;
else if (memcmp(ftp_cmd, CDUP, 4) == 0) *ftp_code = 200; if (*ftp_code
> 0) LogPrintf(LOG_INFO, "EXITFTP (%d) reply code set to %d for \ command
\"%s\"\\n", ftp_ref, *ftp_code, ftp_cmd); return(0);}
```

#### Example 2

##### Forcing a negative completion response

This example shows how to force a negative completion for some commands. If you decide that the server should never receive files, you should intercept STOR and STOU commands and answer with a 502 (command not implemented) or 532 (need account for storing file) return code. In addition, the ftp\_msg parameter is used to reply with a non-standard message.

```
int ExitFtpSetReturnCode (int ftp_ref, char *ftp_cmd, int *ftp_code, char
*ftp_msg){ *ftp_code = 0; ftp_msg[0] = 0; if ((memcmp(ftp_cmd, STOR, 4)
== 0) || (memcmp(ftp_cmd, STOU, 4) == 0)) { *ftp_code = 502; strcpy
(ftp_msg, Sorry, but this server will refuse \ any store command); } if
(*ftp_code > 0) LogPrintf(LOG_INFO, "EXITFTP (%d) reply code set to %d
for \ command \"%s\"\\n", ftp_ref, *ftp_code, ftp_cmd);return(0);}
```

## *Customize FTP: ExitFtpRespPasv and ExitFtpCmdPort*

The signatures of user exit functions supports IPv6 addresses. Input and output parameters of the exit point to a generic container that is able to store IPv4 and IPv6 addresses. When customizing the exit functions, you need to inspect the `sa_family` member of the input parameters and cast this container according to its value (`AF_INET` for IPv4 and `AF_INET6` for IPv6).

The provided sample, which works on IPv4 addresses is updated to reflect the changes. For IPv6, the checks need to take into account the length of the address which is 128 bits and the CIDR notation used for defining the mask (number of bits in the address identifying the network -- `<address>/<mask>`)

### **Purpose**

Gateway is designed to be deployed on the front side of a DMZ (DeMilitarized Zone). It is likely to be hidden from peers by a device which translates TCP addresses. Such devices generally intercept and modify **FTP `PORT/EPRT`** commands and **`PASV/EPSV`** responses according to the translations applied, so that they remain transparent for peers. However, they cannot intercept and modify in FTP sessions that rely on an encrypted transport layer as with TLS.

In order to support such devices (in a very limited range of applications), the `ExitFtpRespPasv` and `ExitFtpCmdPort` exits enable users to translate the address to send to a peer, before encryption.

In Responder mode, Gateway calls `ExitFtpRespPasv` before formatting and sending a **`PASV/EPSV`** response.

In Initiator mode, Gateway calls `ExitFtpCmdPort` before formatting and sending a **`PORT/EPRT`** command.

The purpose of these two exits is to transform the TCP address that is sent to the FTP peer from the one that locally identifies the bound data port to the one that describes it from the other side of an address translation device (that is, from a LAN to an internet address, or from DMZ or private network to public network address).

Additionally, for informative purposes, `ExitFtpRespPasv` and `ExitFtpCmdPort` supply the local and remote addresses of the control session in the local area network. These addresses should be the only data used by the address translation algorithm, and thus should be sufficient to reproduce the translation.

### **Syntax**

```
void ExitFtpRespPasv (struct sockaddr_in localCtrlAddr, struct sockaddr_in remoteCtrlAddr, struct sockaddr_in *localDataAddr)
```

| ExitFtpRespPasv   |                                                                             |                                                                                                                                                                                                                                                                                                |
|-------------------|-----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | <code>localCtrlAddr</code>                                                  | The value of the local TCP address (IP address and port number) of the control connection, as seen by the local system                                                                                                                                                                         |
|                   | <code>remoteCtrlAddr</code>                                                 | The value of the remote TCP address (IP address and port number) of the control connection, as seen by the local system                                                                                                                                                                        |
|                   | <code>localDataAddr</code>                                                  | Pointer to the local TCP address (IP address and port number) of the data connection, as seen by the local system. This address will be used by FTP to generate the <b>PORT/EPRT</b> command / <b>PASV/EPSV</b> response. The function is free to change the content of the pointed structure. |
| Output parameters | <code>localDataAddr</code>                                                  | Pointer to the local TCP address (IP address + port number) of the data connection, as seen by the local system. This address is be used by FTP to generate the <b>PORT/EPRT</b> command / <b>PASV/EPSV</b> response. The function is free to change the content of the pointed structure.     |
| Return            | The return value is always 0. There is no error case support for this exit. |                                                                                                                                                                                                                                                                                                |

### Example

A sample code is provided. It translates the IP part of the given local TCP address of the data session according to a static array, named `ipTransTable`, and included in `exitftp.c`.

If the remote address of the control session is in the local area, the local address of the data session is not translated. Local addresses are defined by a second static array, `ipLanTable`, in `exitftp.c`, that defines IP patterns and sub-network masks that must match LAN IP addresses. By default, this implementation is surrounded by "#if 0" instructions, and thus is not included in the binary components of Gateway.

## Get Application object for HTTP: ExitHttpGetAppli

### Purpose

This exit is called to retrieve the Application if the Remote Site parameter **Application method** (`get_appli_method`) is set to *User exit (AIE)*.

When operating in server mode, Gateway calls this exit on reception of a PUT or POST request.

### Syntax

```
int ExitHttpGetAppli (char *dest_alias, char *filename, short data_code,
char *appli_name, char *destination, char *originator, char *buffer, int
len)
```

| ExitHttpGetAppli  |                          |                                                                      |
|-------------------|--------------------------|----------------------------------------------------------------------|
| Input parameters  | <code>dest_alias</code>  | Remote Site of the connection                                        |
|                   | <code>filename</code>    | Content of protocol filename                                         |
|                   | <code>data_code</code>   | Transfer data code                                                   |
|                   | <code>appli_name</code>  | Content of <code>SITE P_APPLI</code> parameter                       |
|                   | <code>destination</code> | Alias of the final destination                                       |
|                   | <code>originator</code>  | Alias of the transfer originator                                     |
|                   | <code>len</code>         | Max length of buffer parameter                                       |
| Output parameters | <code>buffer</code>      | Buffer to store Application name                                     |
| Return            | 0                        | Application is found and stored in the <code>buffer</code> parameter |
|                   | 1                        | No Application found and default HTTP internal Application used      |
|                   | 2                        | No Application found and Transfer rejected                           |

## Handling received HTML page: ExitHttpPutBodyResponseReceived

### Purpose

Use this exit to handle the HTML page received in response to a PUT or POST request.

In client mode, Gateway calls this exit on reception of a PUT or POST request. It calls the exit after sending the requested file, before it sends the end-of-transfer notification.

### Syntax

```
void ExitHttpPutBodyResponseReceived (NotifXferRecord_t *xfer, char
*request_url, HttpHeaders_t request[], int request_c, char *response_
status, HttpHeaders_t response[], int response_c, char *response_
body)
```

| ExitHttpPutBodyResponseReceived |                              |                                                                                              |
|---------------------------------|------------------------------|----------------------------------------------------------------------------------------------|
| Input parameters                | <code>xfer</code>            | Transfer file structure as defined in <code>exitxdef.h</code>                                |
|                                 | <code>request_url</code>     | URL of the transfer                                                                          |
|                                 | <code>request</code>         | Table containing the HTTP fields used to send the file                                       |
|                                 | <code>request_c</code>       | Number of elements contained in the table that is specified in the <i>request</i> parameter  |
|                                 | <code>response_status</code> | Status line from the HTTP response message                                                   |
|                                 | <code>response</code>        | Table containing the HTTP response message fields                                            |
|                                 | <code>response_c</code>      | Number of elements contained in the table that is specified in the <i>response</i> parameter |
|                                 | <code>response_body</code>   | HTML page identified in <i>response</i>                                                      |
| Output parameters               | None                         |                                                                                              |
| Return                          | None                         |                                                                                              |

### Collecting HTTP request information: *ExitHttpRequestReceived2*

#### Purpose

This exit provides the user with information about incoming HTTP requests.



In server mode, Gateway calls this exit on reception of a request, before processing CGate and pre-connection exits.

### Syntax

```
void ExitHttpRequestReceived2 (int ref_http, In_TlsSessionInfo_t *tls_
infos, char *sessionId, char *request_url, HttpHeaders_t request[],
int request_c, HttpCgiParam_t url_param[], int url_param_c, HttpCgiParam_
t post_param[], int post_param_c, char *buffer, int len)
```

| ExitHttpRequestReceived2 |              |                                                                                                                                        |
|--------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters         | ref_http     | Local HTTP transfer context number                                                                                                     |
|                          | tls_infos    | In_TLS_sessioninfo_t pre-connection exit structure, as defined in <code>exitxdef.h</code>                                              |
|                          | sessionId    | Identity of the session (only if this identification method is selected in the <a href="#">HTTP options</a> in the configuration menu) |
|                          | request_url  | URL of the transfer                                                                                                                    |
|                          | request      | Table containing the HTTP fields used to send the file                                                                                 |
|                          | request_c    | Number of elements contained in the table that is specified in the <code>request</code> parameter                                      |
|                          | url_param    | Table containing URL CGI fields for the transfer                                                                                       |
|                          | url_param_c  | Number of elements contained in the table that is specified in the <code>url_param</code> parameter                                    |
|                          | post_param   | Table containing URL CGI fields for POST requests                                                                                      |
|                          | post_param_c | Number of elements contained in the table that is specified in the <code>post_param</code> parameter                                   |
|                          | len          | Length of the <code>OutProtoSpecS</code> field                                                                                         |

| ExitHttpRequestReceived2 |                 |                                                                                                                                                          |
|--------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Output parameters        | OutProtoSpecStr | User field that can receive a string of characters. This string can be passed to the <code>proto_spec_string</code> parameter of the pre-connection exit |
| Return                   | None            |                                                                                                                                                          |

## Get Application object for POP3: ExitPop3GetAppli

### Purpose

This exit is called to retrieve the Application if the Remote Site parameter **Application method** (`get_appli_method`) is set to *User exit (AIE)*.

Gateway calls this exit immediately after receiving a mail header. The result must be stored in the `buffer` parameter whose size is `bufsize` bytes.

### Syntax

```
int ExitPop3GetAppli (char *originator, char *destination, char *from,
char *to, char *xfer_ident, char *subject, char *filename, char *appli_
name, char *buffer, int bufsize)
```

| ExitPop3GetAppli |                          |                                              |
|------------------|--------------------------|----------------------------------------------|
| Input parameters | <code>originator</code>  | POP3 server ID as defined in the Site object |
|                  | <code>destination</code> | Login USERID as defined in the Site object   |
|                  | <code>from</code>        | The from: field in mail header               |
|                  | <code>to</code>          | The to: field in mail header                 |
|                  | <code>xfer_ident</code>  | The message_id: in mail header               |
|                  | <code>subject</code>     | The subject: field in mail header            |
|                  | <code>appli_name</code>  | The proposed Application name (if any)       |
|                  | <code>file_name</code>   | Reserved                                     |
|                  | <code>bufsize</code>     | Max length of buffer parameter               |

| <b>ExitPop3GetAppli</b> |                                                                                                                                                    |                                                            |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| Output parameters       | <code>buffer</code>                                                                                                                                | Buffer to store the desired Application name               |
| Return                  | 0                                                                                                                                                  | Application is found and stored in the "buffer" parameter  |
|                         | 1                                                                                                                                                  | No Application found and default internal Application used |
|                         | 2                                                                                                                                                  | No Application found and Transfer rejected                 |
| Example                 | The sample function ( <code>exitpop3.c</code> ) provides an example using the proposed Application name (if any). Any other method can be applied. |                                                            |
| Additional information  | For more information on POP3, refer to <a href="#">About SMTP/POP3</a> .                                                                           |                                                            |

### *Process log file archive: ExitLogArchived*

#### **Purpose**

This exit is called after archiving the log file, if asynchronous batch processing program for log archiving is not used.

#### **Syntax**

```
int ExitLogArchived (char *path_name)
```

| <b>ExitLogArchived</b> |                                               |                                       |
|------------------------|-----------------------------------------------|---------------------------------------|
| Input parameters       | <code>path_name</code>                        | Pathname of the file that is archived |
| Output parameters      | None                                          |                                       |
| Return                 | The return value of this function is always 0 |                                       |

Create statistics file: *ExitStatMessage*

**Purpose**

This exit is called before a statistics record is formatted and written in the Gateway stat file. You can use this function to create a specific user STAT file.

**Syntax**

```
int ExitStatMessage (NotifXferRecord_t *param)
```

| ExitStatMessage   |       |                                                                                                                                         |
|-------------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | param | Pointer to the public structure of the Transfer parameters.<br>Refer to <code>exitxdef.h</code> for more details on the structure used. |
| Output parameters | None  |                                                                                                                                         |
| Return            | 0     | If no stat record is to be formatted by Gateway                                                                                         |
|                   | n     | Statistics record format number to be formatted by Gateway                                                                              |

Leave stats: *ExitStatArchived*

**Purpose**

This exit is called after archiving the stat file archiving, if asynchronous batch processing program for statistics archiving is not used.

**Syntax**

```
int ExitStatArchived (char *path_name)
```

| ExitStatArchived  |                                               |                                       |
|-------------------|-----------------------------------------------|---------------------------------------|
| Input parameters  | path_name                                     | Pathname of the file that is archived |
| Output parameters | None                                          |                                       |
| Return            | The return value of this function is always 0 |                                       |

## Check Transfer Request: *ExitCheckXferRequest*

### Purpose

This exit is called by a client program when a user issues a Transfer Request. Typical client programs are Gateway commands and operator interface.

Use this exit to check external security or to modify Transfer Requests depending on Gateway username or other external parameters (such as system user).

Note: Unlike other user access control exits which run on the server side, this routine is called from a client program and runs on the client side. Unless *ExitCheckXferRequest* is in a shared library (such as DLL), you must re-link all client programs when this routine is modified.

### Syntax

```
int ExitCheckXferRequest (char *itp_user_name, ItpXferRequest_t *req)
```

| ExitCheckXferRequest |                            |                                                                                                             |
|----------------------|----------------------------|-------------------------------------------------------------------------------------------------------------|
| Input parameters     | <code>itp_user_name</code> | Gateway user name (can be different from system user name)                                                  |
|                      | <code>req</code>           | Transfer Request to check or modify.<br>Refer to <code>itpdefs.h</code> for more details on structure used. |
| Output parameters    | <code>req</code>           | Transfer Request to check or modify.<br>Refer to <code>itpdefs.h</code> for more details on structure used. |
| Return               | <code>E_UUNDEFINED</code>  | Return No exit user action (access accepted)                                                                |
|                      | <code>E_UDENIED</code>     | Access denied by user routine                                                                               |
|                      | <code>E_UALLOWED</code>    | Access accepted by user routine                                                                             |

## Write log file record: *ExitLogMessage*

### Purpose

This exit is called before a record is written in the log file.

### Syntax

```
int ExitLogMessage(LogMessage *msg, char *text)
```

| ExitLogMessage    |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | msg           | Pointer to the message descriptor. The message descriptor contains the following parameters: <ul style="list-style-type: none"><li>time_t time_stampdate of the event corresponding to the message</li><li>char ident[20+1]message identification (example: SUP101W)</li><li>int param_countnumber of variable parameters in the message (information not supplied in this version)</li><li>char *param_value[20]values of variable parameters in the message (information not supplied in this version)</li></ul> |
|                   | text          | Pointer to the message text                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Output parameters | None          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Return            | EXIT_TO_NOTIF | Returns the value EXIT_TO_NOTIF if the external exit ExitLogMessageExt_C () must be notified for this log message.                                                                                                                                                                                                                                                                                                                                                                                                 |
|                   | 0             | If the user processing applied on log messages has a very short response time, this normal user routine ExitLogMessage() embedded with the monitor system process (p_sys for UNIX platforms) can be used directly. Whenever a very short response time cannot be guaranteed, it is recommended to implement the user processing code in the external exit ExitLogMessageExt_C().                                                                                                                                   |

User exits and Rule Tables

[Overview](#)

### [Activating the exit trigger](#)

[Rule Table exit routines](#) ([ExitAfterTemporalEvent](#) and [ExitAfterFileStoringEvent](#))

## Overview

Gateway provides two user exits related to the handling of Rule Tables. If activated, these exits enable you to create execution conditions based on one or more of the symbolic variables handled by the exit routine.

You can use these exits to:

- Execute controls that determine whether or not a Table is scanned
- Return user variables that can be reused in transfer models or in execution conditions
- Initiate various processes via the APIs that are available to you

## Activating the exit trigger

When you create a Rule Table you define the table as being **with** or **without** an associated exit. You can do this using the GUI or the `peladmin create_ruletable` command.

- When you define a Rule Table **with** exit, Gateway executes the exit that corresponds to the rule table category before executing the normal table scanning process.
- When you define a Rule Table **without** exit, Gateway scans the table without triggering the exit.

## Rule Table exit routines

You can associate exit routines with two types of Rule Table:

- Scheduling Rule Tables
- Directory Scanning Rule Tables

### Scheduling type Rule Tables: `ExitAfterTemporalEvent`

For Scheduling-type Rule Tables, use the `ExitAfterTemporalEvent` exit.

Gateway calls the `ExitAfterTemporalEvent` exit immediately before scanning the Scheduling-type Rule Table. The exit returns a value to the Gateway standard processes that can determine whether or not the Scheduling Rule Table is to be scanned. Gateway transmits to the exit certain scheduling related symbolic variables, as well as the name of the Rule Table. Gateway can also pass user variables that are used in the conditions of the Decision Rules of the table or in the models that are applied when Rule Table conditions are satisfied.

Gateway provides a sample of the `ExitAfterTemporalEvent` exit file, `ExitAfterTemporalEvent.c`, that contains a set of default values. The default values have no effect on Rule Table processing.

To customize the exit function, insert your own C-language code into the `ExitAfterTemporalEvent.c` exit file. Your customized code determines the values that are returned to the standard Rule Table process. Gateway takes the results into account before it scans the table.

When you develop your code, you can create conditions using one or more of the ten variables that are handled by the `ExitAfterTemporalEvent` exit. You can also include many functions in your customized code by using APIs. For example, Gateway includes an API that enables you to obtain information about the calendar that is associated with a Rule Table.

When you develop your code, you can create conditions using one or more of the variables that are handled by the `ExitAfterTemporalEvent` exit. Gateway initializes 12 symbolic variables immediately before scanning that Directory Scanning Rule Table. Of these variables, the `ExitAfterFileStoringEvent` exit can return useful values for the `integer_user_paramX` and `string_user_paramX` variables.

You can also include many functions in your customized code by using APIs.

## Syntax

```
int ExitAfterTemporalEvent(In_ExitTemporal_t *in, Out_ExitRte_t *out)
```

| ExitAfterTemporalEvent |                   |                                                                                                                                                   |
|------------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters       | In_ExitTemporal_t | Structure that comprises Rule Table parameters.                                                                                                   |
| Output parameters      | Out_ExitRte_t     | Structure that specifies Rule Table related attributes modified by user code. Refer to <code>exitrte.h</code> for more details on structure used. |
| Return                 | 0                 | Task performed successfully. Uses returned parameter value and proceeds normally.                                                                 |
|                        | 1                 | Nothing done. Discards output parameters and proceeds normally.                                                                                   |

### Directory Scanning type Rule Tables: ExitAfterFileStoringEvent

For Directory Scanning type Rule Tables, use the `ExitAfterFileStoringEvent` exit.

Gateway calls the `ExitAfterFileStoringEvent` exit immediately before scanning the Storing-type Rule Table. The exit returns a value to the Gateway standard processes that can determine whether or not the Directory Scanning Rule Table is to be scanned. Gateway transmits to the exit certain scheduling related symbolic variables, as well as the name of the Rule Table. Gateway can also pass user variables that are used in the conditions of the Decision Rules of the table or in the models that are applied when Rule Table conditions are satisfied.

Gateway provides a sample of the `ExitAfterFileStoringEvent` exit file, `ExitAfterFileStoringEvent.c`, that contains a set of default values. The default values have no effect on Rule Table processing.



To customize the exit function you insert your own C-language code into the `ExitAfterFileStoringEvent.c` exit file. Your customized code determines the values that are returned to the standard Rule Table process. Gateway takes the results into account before it scans the table.

When you develop your code you can create conditions using one or more of the variables that are handled by the `ExitAfterFileStoringEvent` exit. Gateway initializes 19 symbolic variables immediately before scanning that Directory Scanning Rule Table. Of these variables, the `ExitAfterFileStoringEvent` exit can return useful values for the `integer_user_paramX` and `string_user_paramX` variables.

You can also include many functions in your customized code by using APIs.

## Syntax

```
int ExitAfterFileStoringEvent(In_ExitFileStoring_t *in, Out_ExitRte_t *out)
```

| ExitAfterFileStoringEvent |                           |                                                                                                                                                   |
|---------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters          | In_ExitFileStoring_t      | Structure that comprises Rule Table parameters.                                                                                                   |
| Output parameters         | Out_ExitXferBeforeBegin_t | Structure that specifies Rule Table related attributes modified by user code. Refer to <code>exitrte.h</code> for more details on structure used. |
| Return                    | 0                         | Task performed successfully. Uses returned parameter value and proceeds normally.                                                                 |
|                           | 1                         | Nothing done. Discards output parameters and proceeds normally.                                                                                   |

Axway Gateway: Customizing Gateway Processes

## User exits in Perl

The GIK Perl external user exit **GIKuexpl** comprises a set of five Perl exits:

- `beforebegin.pl` – Transfer receive exit
- `afterbegin.pl` – Start-of-transfer exit
- `afterend.pl` – End-of-transfer exit
- `xfermonitor.pl` – Routing behavior exit
- `exitpcnx.pl` – Protocol connection exit

All the exits are located in the following directory:

### On UNIX

```
<Gateway installation directory>/run_time/scripts/perl/
```

### On Windows

```
<Gateway installation directory>\run_time\scripts\perl\
```

## Parameters

Each Perl exit includes a set of parameters (indication and response parameters).

You can access these parameters via two key = value type tables:

- Indication parameters are referenced by a hash table called **IN**.
- Response parameters are referenced by a hash table called **OUT**.

### Indication parameters: IN

The indications supplied in this table are for information purposes only and any changes to their values have no effect on the transfer. Two types of indication parameter exist:

- Transfer TLS parameters (prefixed with **TlsInfo**)
- Transfer parameters (prefixed with **Params**)

```
TlsInfo_server_auth
```

```
Params_org_alias
```

### Response parameters: OUT

You can modify exit response parameters. Response parameters are prefixed with **Params**:

```
Params_dir_path
```

The **afterbegin** exit does not use response parameters.

### Input/Output user parameters: IN/OUT

The **xfermonitor.pl** exit includes certain user parameters that are used as both input and output parameters. These parameters are always prefixed with **UserParam**:

```
UserParam_string_user_param1
```

### Perl parameter display script

The following is a sample Perl script used to display all transfer indications and response parameters.

```
#transfer indications
@keys = keys %IN;
@values = values %IN;
while (@keys) {
 print pop(@keys), '=', pop(@values);
```

```

}
#response parameters
@keys = keys %OUT;
@values = values %OUT;
while (@keys) {
 print pop(@keys), '=', pop(@values);
}

```

Axway Gateway: Customizing Gateway Processes

## External user exits

**Caution** Please note that care should be taken when **customizing user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide* > *Security best practices* > [Identified threats and possible mitigations](#) section for additional information.

[External user exits](#)

[List of external user exits](#)

[Details of external user exits](#)

## External user exits

This topic describes the external user exits supplied with Axway Gateway.

External exit routines are grouped in the following exit families:

- FT: File Transfer protocol events  
Protocol session establishment/release, file transfer beginning/completion/abortion
- SECS: TLS and SSH Handshake events  
Refer to [User exits: Security functions](#)

On starting, an exit process informs Gateway of the exit families it is expecting.

The following source file samples, located in the <Gateway installation directory>/src/exitextc directory, contain the default implementation of user exits supplied with Gateway:

|            |                                                                   |
|------------|-------------------------------------------------------------------|
| exitcnx.c  | FT family exit routines/connection (session) events               |
| exitxfer.c | FT family exit routines/transfer events                           |
| exitpcnx.c | FT family exit routines/protocol connection events                |
| exitsecs.c | SECS family TLS exit routines                                     |
| exitsech.c | SECS family SSH exit routines                                     |
| exitmisc.c | FT family exit routines/Site state, Transfer state and log events |

Protocol-specific exit routines are supplied with internal routine call mechanisms only. Refer to [Internal user exits](#).

### *Cascadable exits*

All cascadable exits are external exit routines that can be implemented in C or in Perl.

Each of these C and Perl routines must return:

|      |                                                                        |
|------|------------------------------------------------------------------------|
| 0 :  | If nothing has been done, output parameters are discarded.             |
| 1 :  | If task has been performed successfully, returned parameters are used. |
| -1 : | If an error occurs.                                                    |

When a cascadable exit is activated, the corresponding C and Perl exit routines are called in this order, until one of them returns 1 or -1. If both return 0, a default exit routine is called. This default exit routine may do nothing at all (if the default behavior requires no specific processing).

Some external exits are not cascadable.

### *List of external user exits*

The following table lists the available external exits and their associated header and source files.

Follow the links for details about using the exits.

| Access Files                             | User exit                            | Called...                                                                                      |
|------------------------------------------|--------------------------------------|------------------------------------------------------------------------------------------------|
| exitxfer.c<br>and exitext.h<br>( & Perl) | <a href="#">ExitXferBeforeBegin</a>  | Before the beginning of a Transfer Request                                                     |
|                                          | <a href="#">ExitXferAfterBegin</a>   | After the beginning of a Transfer Request                                                      |
|                                          | <a href="#">ExitXferAfterEnd</a>     | When the Transfer is ended                                                                     |
| exitcnx.c and<br>exitext.h               | <a href="#">ExitIcBeforeInitReq</a>  | Before making an outgoing connection request in initiator mode                                 |
|                                          | <a href="#">ExitIcAfterInitConf</a>  | After confirming an outgoing connection request in initiator mode                              |
|                                          | <a href="#">ExitRcBeforeInitResp</a> | When an incoming connection request is received in responder mode                              |
|                                          | <a href="#">ExitIcBeforeTermReq</a>  | Before making an outgoing end of connection request in initiator mode                          |
|                                          | <a href="#">ExitRcAfterTermInd</a>   | When an incoming connection termination request is made in responder mode                      |
|                                          | <a href="#">ExitIcAfterTermConf</a>  | When a confirmation of an outgoing termination of connection request is made in initiator mode |
|                                          | <a href="#">ExitBcAbortReq</a>       | Before aborting an outgoing connection                                                         |
|                                          | <a href="#">ExitBcAbortInd</a>       | When an incoming connection abort is received                                                  |

|                                         |                                       |                                                                                                                                                                                                        |
|-----------------------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| exitpcnx.c<br>and exitext.h<br>(& Perl) | <a href="#">ExitRcProtoConn</a>       | To accept or reject an incoming call                                                                                                                                                                   |
| exitsecs.c<br>and exitext.h             | <a href="#">ExitSecsGetSrvPubKey</a>  | To retrieve the SECS server public key                                                                                                                                                                 |
|                                         | <a href="#">ExitSecsSetSrvPubKey</a>  | To set the SECS server public key                                                                                                                                                                      |
|                                         | <a href="#">ExitSecsSessionInit</a>   | To initialize the SECS session                                                                                                                                                                         |
|                                         | <a href="#">ExitSecsSessionEnd</a>    | To end the SECS session                                                                                                                                                                                |
|                                         | <a href="#">ExitSecsCheckCertList</a> | To check the SECS Certification Path                                                                                                                                                                   |
|                                         | <a href="#">ExitSecsComputation</a>   | To compute the public or private SECS key                                                                                                                                                              |
|                                         | <a href="#">ExitSecsGetCertList</a>   | To access a valid certificate to send                                                                                                                                                                  |
|                                         | <a href="#">ExitSecsBuildCaDnList</a> | To build CA accepted authorities DN list (server mode only)                                                                                                                                            |
|                                         | <a href="#">ExitSecsHandshakeDone</a> | At the end of the handshake phase                                                                                                                                                                      |
| exitmisc.c<br>and exitext.h             | <a href="#">ExitSiteState</a>         | Each time a Site state changes (using the <code>pelctl start_site</code> and <code>pelctl stop_site</code> control commands)                                                                           |
|                                         | <a href="#">ExitXferState</a>         | Each time the Transfer Request state changes                                                                                                                                                           |
|                                         | <a href="#">ExitXferMonitor</a>       | Each time a Transfer must be routed (also available in Perl)                                                                                                                                           |
|                                         | <a href="#">ExitLogMessageExt</a>     | When the internal <code>ExitLogMessage()</code> user exit selects a message. <code>ExitLogMessageExt()</code> is recommended when user processing on log messages does not have a short response time. |

## Details of external user exits

### Pre-process Transfer Request: `ExitXferBeforeBegin_C`

#### Purpose

This exit is called before the beginning of a Transfer Request.

#### Syntax

```
int ExitXferBeforeBegin_C(In_SessionInfo_t *si, In_ExitXferBeforeBegin_t
 *in, Out_ExitXferBeforeBegin_t *out)
```

|                       |
|-----------------------|
| ExitXferBeforeBegin_C |
|-----------------------|

|                   |                                   |                                                                                                                                        |
|-------------------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | In_TlsSessionInfo_t<br>*si        | Structure that comprises TLS or SSH session parameters.                                                                                |
|                   | In_ExitXferBeforeBegin_t<br>*in   | Structure that comprises Transfer record parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.          |
| Output parameters | Out_ExitXferBeforeBegin_t<br>*out | Structure that comprises Transfer parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |
| Return            | 0                                 | Nothing done. Discards output parameters and proceeds normally.                                                                        |
|                   | 1                                 | Task performed successfully. Uses returned parameter value and proceeds normally.                                                      |
|                   | -1                                | Error. Aborts current processing.                                                                                                      |

### *In-process Transfer Request: ExitXferAfterBegin\_C*

#### **Purpose**

This exit is called after the beginning of a Transfer Request.

#### **Syntax**

```
int ExitXferAfterBegin_C (In_TlsSessionInfo_t *si, In_ExitXferAfterBegin_t *in, Out_ExitXferAfterBegin_t *out)
```

| ExitXferAfterBegin_C |                                  |                                                                                                                                        |
|----------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters     | In_TlsSessionInfo_t<br>*si       | Structure that comprises TLS or SSH session parameters                                                                                 |
|                      | In_ExitXferAfterBegin_t<br>*in   | Structure that comprises Transfer record parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.          |
| Output parameters    | Out_ExitXferAfterBegin_t<br>*out | Structure that comprises Transfer parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |

|        |    |                                                                                   |
|--------|----|-----------------------------------------------------------------------------------|
| Return | 0  | Nothing done. Discards output parameters and proceeds normally.                   |
|        | 1  | Task performed successfully. Uses returned parameter value and proceeds normally. |
|        | -1 | Error. Aborts current processing.                                                 |

### *Post-process Transfer Request: ExitXferAfterEnd\_C*

#### **Purpose**

This exit is called when the Transfer is ended.

#### **Syntax**

```
int ExitXferAfterEnd_C (In_TlsSessionInfo_t *si, In_ExitXferAfterEnd_t
 *in, Out_ExitXferAfterEnd_t *out)
```

| ExitXferAfterEnd_C |                             |                                                                                                                                        |
|--------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters   | In_TlsSessionInfo_t *si     | Structure that comprises TLS or SSH session parameters.                                                                                |
|                    | In_ExitXferAfterEnd_t *in   | Structure that comprises Transfer record parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.          |
| Output parameters  | Out_ExitXferAfterEnd_t *out | Structure that comprises Transfer parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |
| Return             | 0                           | Nothing done. Discards output parameters and proceeds normally.                                                                        |
|                    | 1                           | Task performed successfully. Uses returned parameter value and proceeds normally.                                                      |
|                    | -1                          | Error. Aborts current processing.                                                                                                      |

### *Processing routed Transfer Request: ExitXferMonitor\_C*

#### **Purpose**

This exit is called whenever the Transfer is routed.

**Syntax**

```
int ExitXferMonitor_C (In_ExitXferMonitor_t *in, OutIn_ExitXferMonitor_t *out_in, Out_ExitXferMonitor_t *out)
```

| <b>ExitXferMonitor_C</b> |                                    |                                                                                                                               |                                                                                                                                    |
|--------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters         | In_ExitXferMonitor_t<br>*in        | Structure that comprises Transfer record parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used. |                                                                                                                                    |
|                          |                                    |                                                                                                                               |                                                                                                                                    |
| Output/input parameter   | OutIn_ExitXferMonitor_t<br>*out_in | This structure is used as both an input and an output structure.                                                              |                                                                                                                                    |
|                          |                                    | <i>integer_user_param1</i><br><i>integer_user_param5</i>                                                                      | User variables of type Integer                                                                                                     |
|                          |                                    | <i>string_user_param1</i><br><i>string_user_param5</i>                                                                        | User variables of type String                                                                                                      |
|                          |                                    | <i>appli_user_param1appl</i><br><i>i_user_param2</i>                                                                          | User variables relating to the Application object. It is highly recommended that you do not modify these variables in the exit.    |
|                          |                                    | <i>cgate_user_param1cgate</i><br><i>_user_param2</i>                                                                          | User variables relating to the selected CGate object. It is highly recommended that you do not modify these variables in the exit. |
|                          |                                    | <i>site_user_param1site</i><br><i>_user_param2</i>                                                                            | User variables relating to the local Site object. It is highly recommended that you do not modify these variables in the exit.     |



|                  |                                   |                                                                                    |                                                   |
|------------------|-----------------------------------|------------------------------------------------------------------------------------|---------------------------------------------------|
| Output parameter | Out_<br>ExitXferMonitor_t<br>*out | Output structure that determines routing behavior for Transfer.                    |                                                   |
|                  |                                   | <i>exec_type</i>                                                                   | Execution type (Integer)                          |
|                  |                                   | <i>link_to_xfer</i>                                                                | Parameter used to link input and output Transfers |
|                  |                                   | <i>model_name</i>                                                                  | Name of Model to apply                            |
|                  |                                   | <i>perl_script_name</i>                                                            | Absolute path of Perl script to execute           |
|                  |                                   | <i>command_line</i>                                                                | Command line to execute                           |
| Return           | 0                                 | Nothing done. Discards output parameters and proceeds normally.                    |                                                   |
|                  | 1                                 | Task performed successfully. Uses returned parameters value and proceeds normally. |                                                   |
|                  | -1                                | Error occurred. Aborts current processing.                                         |                                                   |

### *Pre-connection request: ExitIcBeforeInitReq\_C*

#### **Purpose**

This exit is called on the initiator Site before sending a protocol connection request to a Remote Site.

#### **Syntax**

```
int ExitIcBeforeInitReq_C (In_TlsSessionInfo_t *tls, In_ExitIcBeforeInitReq_t *in, Out_ExitIcBeforeInitReq_t *out)
```

| ExitIcBeforeInitReq_C |                                   |                                                                                                                                                   |
|-----------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters      | In_TlsSessionInfo_t<br>*tls       | Structure that comprises TLS session parameters                                                                                                   |
|                       | In_ExitIcBeforeInitReq_t<br>*in   | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.                 |
| Output parameters     | Out_ExitIcBeforeInitReq_t<br>*out | Structure that comprises protocol connection parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |

|        |    |                                                                                                                               |
|--------|----|-------------------------------------------------------------------------------------------------------------------------------|
| Return | 0  | Nothing done. Discards output parameters and proceeds normally.                                                               |
|        | 1  | Task performed successfully. Uses returned parameter value and proceeds normally.                                             |
|        | -1 | Error. Aborts current processing. Gateway retries the connection request two seconds later and calls this exit routine again. |

### *Confirm outgoing: ExitIcAfterInitConf\_C*

#### **Purpose**

I: Initiator mode, c: connection, AfterInitConf: After Initialization Confirmation.

This exit is called whenever confirmation of an outgoing connection request is made in Initiator mode.

Use this user routine to:

- Check identification parameters that the Remote Site returns in the connection confirmation
- Decide whether Gateway can start transfers on this connection or should abort the connection

#### **Syntax**

```
int ExitIcAfterInitConf_C (In_TlsSessionInfo_t *tls, In_ExitIcAfterInitConf_t *in, Out_ExitIcAfterInitConf_t *out)
```

| ExitIcAfterInitConf_C |                                |                                                                                                                                                   |
|-----------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters      | In_TlsSessionInfo_t *tls       | Structure that comprises TLS session parameters                                                                                                   |
|                       | In_ExitIcAfterInitConf_t *in   | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.                 |
| Output parameters     | Out_ExitIcAfterInitConf_t *out | Structure that comprises protocol connection parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |
| Return                | 0                              | Nothing done. Discards output parameters and proceeds normally.                                                                                   |
|                       | 1                              | Task performed successfully. Uses returned parameter value and proceeds normally.                                                                 |
|                       | -1                             | Error. Aborts current processing.                                                                                                                 |

## Confirm incoming: *ExitRcBeforeInitResp\_C*

### Purpose

R: Responder mode, c: connection, BeforeInitResp: Before Initialization Response

This exit is called whenever an incoming connection request is received in Responder mode.

This exit is used to check identification parameters sent by a Remote Site and to determine whether or not Gateway should accept incoming connection indications.

### Syntax

```
int ExitRcBeforeInitResp_C (In_TlsSessionInfo_t *tls, In_
ExitRcBeforeInitResp_t *in, Out_ExitRcBeforeInitResp_t *out)
```

| ExitRcBeforeInitResp_C |                                 |                                                                                                                                                   |
|------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters       | In_TlsSessionInfo_t *tls        | Structure that comprises TLS session parameters                                                                                                   |
|                        | In_ExitRcBeforeInitResp_t *in   | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.                 |
| Output parameters      | Out_ExitRcBeforeInitResp_t *out | Structure that comprises protocol connection parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |
| Return                 | 0                               | Nothing done. Discards output parameters and proceeds normally.                                                                                   |
|                        | 1                               | Task performed successfully. Uses returned parameter value and proceeds normally.                                                                 |
|                        | -1                              | Error. Aborts current processing.                                                                                                                 |

## Terminate outgoing: *ExitlcBeforeTermReq\_C*

### Purpose

I: initiator mode, c: connection, BeforeTermReq: Before Termination Request

This exit is called whenever an outgoing termination of connection request is to be made in initiator mode.

**Syntax**

```
int ExitIcBeforeTermReq_C (In_TlsSessionInfo_t *tls, In_
ExitIcBeforeTermReq_t *in)
```

| ExitIcBeforeTermReq_C |                              |                                                                                                                                   |
|-----------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Input parameters      | In_TlsSessionInfo_t *tls     | Structure that comprises TLS session parameters                                                                                   |
|                       | In_ExitIcBeforeTermReq_t *in | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used. |
| Output parameters     | None                         |                                                                                                                                   |
| Return                | 0                            | Nothing done. Discards output parameters and proceeds normally.                                                                   |
|                       | 1                            | Task performed successfully. Uses returned parameter value and proceeds normally.                                                 |
|                       | -1                           | Error. Aborts current processing.                                                                                                 |

*Terminate incoming: ExitRcAfterTermInd\_C***Purpose**

R: Responder mode, c: connection, AfterTermInd: After Termination Indication

This exit is called in responder mode, whenever an incoming connection termination request is made.

**Syntax**

```
int ExitRcAfterTermInd_C (In_TlsSessionInfo_t *tls, In_
ExitRcAfterTermInd_t *in)
```

| ExitRcAfterTermInd_C |                             |                                                                                                                                   |
|----------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Input parameters     | In_TlsSessionInfo_t *tls    | Structure that comprises TLS session parameters                                                                                   |
|                      | In_ExitRcAfterTermInd_t *in | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used. |
| Output parameters    | None                        |                                                                                                                                   |

|        |    |                                                                                   |
|--------|----|-----------------------------------------------------------------------------------|
| Return | 0  | Nothing done. Discards output parameters and proceeds normally.                   |
|        | 1  | Task performed successfully. Uses returned parameter value and proceeds normally. |
|        | -1 | Error. Aborts current processing.                                                 |

### *Terminate outgoing: ExitIcAfterTermConf\_C*

#### **Purpose**

I: initiator mode, c: connection, AfterTermConf: After Termination Confirmation.

This exit is called whenever confirmation of an outgoing termination of a connection request is made in initiator mode.

#### **Syntax**

```
int ExitIcAfterTermConf_C (In_TlsSessionInfo_t *tls, In_ExitIcAfterTermConf_t *in)
```

| ExitIcAfterTermConf_C |                              |                                                                                                                                   |
|-----------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Input parameters      | In_TlsSessionInfo_t *tls     | Structure that comprises TLS session parameters                                                                                   |
|                       | In_ExitIcAfterTermConf_t *in | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used. |
| Output parameters     | None                         |                                                                                                                                   |
| Return                | 0                            | Nothing done. Discards output parameters and proceeds normally.                                                                   |
|                       | 1                            | Task performed successfully. Uses returned parameter value and proceeds normally.                                                 |
|                       | -1                           | Error. Aborts current processing.                                                                                                 |

### *Abort outgoing: ExitBcAbortReq\_C*

#### **Purpose**

B: Both initiator and responder mode, c: connection, AbortReq: Abort Request

This exit is called in both initiator and responder mode whenever an abort request is to be made for an outgoing connection.

**Syntax**

```
int ExitBcAbortReq_C (In_TlsSessionInfo_t *tls, In_ExitBcAbortReq_t *in
)
```

| ExitBcAbortReq_C  |                          |                                                                                                                                   |
|-------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | In_TlsSessionInfo_t *tls | Structure that comprises TLS session parameters                                                                                   |
|                   | In_ExitBcAbortReq_t *in  | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used. |
| Output parameters | None                     |                                                                                                                                   |
| Return            | 0                        | Nothing done. Discards output parameters and proceeds normally.                                                                   |
|                   | 1                        | Task performed successfully. Uses returned parameter value and proceeds normally.                                                 |
|                   | -1                       | Error. Aborts current processing.                                                                                                 |

***Abort incoming: ExitBcAbortInd\_C*****Purpose**

B: Both initiator and responder mode, c: connection, AbortInd: Abort Indication.

This exit is called whenever an abort indication is received for an incoming connection.

**Syntax**

```
int ExitBcAbortInd_C (In_TlsSessionInfo_t *tls, In_ExitBcAbortInd_t *in
)
```

| ExitBcAbortInd_C  |                          |                                                                                                                                   |
|-------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | In_TlsSessionInfo_t *tls | Structure that comprises TLS session parameters                                                                                   |
|                   | In_ExitBcAbortInd_t *in  | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used. |
| Output parameters | None                     |                                                                                                                                   |

|        |    |                                                                                   |
|--------|----|-----------------------------------------------------------------------------------|
| Return | 0  | Nothing done. Discards output parameters and proceeds normally.                   |
|        | 1  | Task performed successfully. Uses returned parameter value and proceeds normally. |
|        | -1 | Error. Aborts current processing.                                                 |

### *Incoming call: ExitRcProtoConn\_C*

#### **Purpose**

R: Responder mode, c: connection, ProtoConn: Protocol Connection

This exit is called to accept or reject an incoming call.

#### **Syntax**

```
int ExitRcProtoConn_C (In_TlsSessionInfo_t *si_info, In_ExitRcProtoConn_t *in, Out_ExitRcProtoConn_t *out)
```

| ExitRcProtoConn_C |                            |                                                                                                                                                   |
|-------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | In_TlsSessionInfo_t *si    | Structure that comprises TLS or SSH session parameters.                                                                                           |
|                   | In_ExitRcProtoConn_t *in   | Structure that comprises protocol connection parameters.<br>Refer to <code>exitxdef.h</code> for more details on structures used.                 |
| Output parameters | Out_ExitRcProtoConn_t *out | Structure that comprises protocol connection parameters modified by user.<br>Refer to <code>exitxdef.h</code> for more details on structure used. |
| Return            | 0                          | Nothing done. Discards output parameters and proceeds normally.                                                                                   |
|                   | 1                          | Task performed successfully. Uses returned parameter value and proceeds normally.                                                                 |
|                   | -1                         | Incoming call rejected or error.                                                                                                                  |

### *Process log message: ExitLogMessageExt\_C*

#### **Purpose**

This exit is called for messages that were selected by the internal `ExitLogMessage()` user exit.

To select a message, `ExitLogMessage()` must return the value `EXIT_TO_NOTIF` each time it is called with this message.

- If user processing applied to log messages has a very short response time, you can use the internal exit `ExitLogMessage()` directly. This exit is embedded with the monitor system process (`p_sys` for UNIX platforms).
- If a very short response time cannot be guaranteed, it is recommended that you implement this external exit `ExitLogMessageExt_C()`.

### Syntax

```
int ExitLogMessageExt_C (In_ExitLogMessageExt_t *in)
```

| ExitLogMessageExt_C |                            |                                                                                                                                              |
|---------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters    | In_ExitLogMessageExt_t *in | Structure that comprises message parameters.<br>Refer to the description of <code>exitxdef.h</code> for more information on structures used. |
| Output parameters   | None                       |                                                                                                                                              |
| Return              | 0                          | Nothing done                                                                                                                                 |
|                     | 1                          | Task performed successfully                                                                                                                  |
|                     | -1                         | Error. Aborts current processing                                                                                                             |

## Site state change: ExitSiteState\_C

### Purpose

This exit is called each time the state of a Site changes (via the `pelctl start_site` and `pelctl stop_site` commands).

### Syntax

```
int ExitSiteState_C (In_TlsSessionInfo_t *tls, In_ExitSiteState_t *in)
```

| ExitSiteState_C   |                          |                                                                                                                                                  |
|-------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | In_TlsSessionInfo_t *tls | Structure that comprises TLS session parameters.<br>Refer to the description of <code>exitxdef.h</code> for more information on structures used. |
|                   | In_ExitSiteState_t *in   | Structure that comprises Site parameters                                                                                                         |
| Output parameters | None                     |                                                                                                                                                  |



|        |    |                                  |
|--------|----|----------------------------------|
| Return | 0  | Nothing done                     |
|        | 1  | Task performed successfully      |
|        | -1 | Error: Aborts current processing |

## *Transfer Request state change: ExitXferState\_C*

### **Purpose**

This exit is called each time the state of a Transfer Request changes.

### **Syntax**

```
int ExitXferState_C (In_ExitXferState_t *in)
```

| ExitXferState_C   |                        |                                                                                                                                            |
|-------------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Input parameters  | In_ExitXferState_t *in | Structure that comprises Transfer parameters. Refer to the description of <code>exitxdef.h</code> for more information on structures used. |
| Output parameters | None                   |                                                                                                                                            |
| Return            | 0                      | Nothing done                                                                                                                               |
|                   | 1                      | Task performed successfully                                                                                                                |
|                   | -1                     | Error: Aborts current processing                                                                                                           |

[About user exits](#)

[User exits: Security functions](#)

[TLS user exits](#)

[SSH user exits](#)

Axway Gateway: Customizing Gateway Processes

## *User exits: Security functions*

[Overview](#)

[Activating Security exits](#)

[SECS exit scheduling mechanisms](#)

[Return values](#)

[Shared parameters](#)

## Overview

The Handshake negotiation at the start of each TLS and SSH layer session enables the client to authenticate the server identity by using public key techniques (and, optionally, for the server to authenticate the client). It also enables the client and server to cooperate in creating symmetric keys for encryption, decryption, and detection of modifications.

User exits provide you with a means of customizing the way these services are processed.

For details about the individual security exits, refer to:

- [TLS user exits](#)
- [SSH user exits](#)

## Activating Security exits

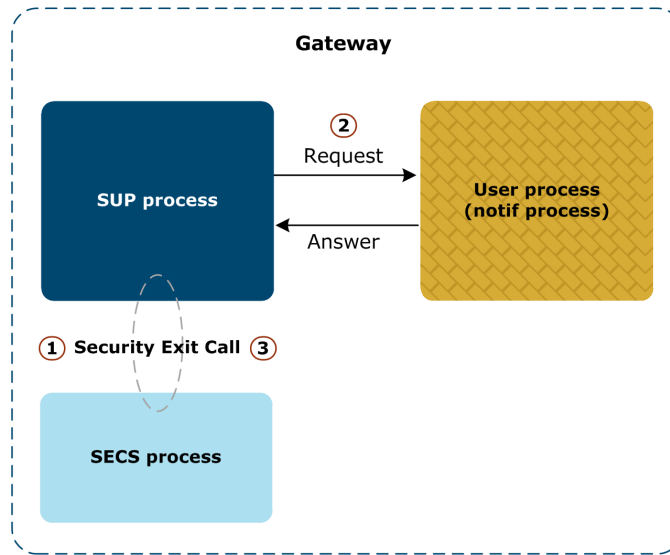
To activate security exit function process calls:

- In the configuration menu, set the **Connection control method** parameter to *Exit*.  
See *Security Guide* > [Managing TLS Security Profiles](#) and *Security Guide* > [Managing TLS Security Profiles \(command line\)](#) for further information.
- For SSH, activate the **exit\_scheduling** parameter.  
See *Security Guide* > [Managing SSH Security Profiles](#) and [Managing SSH Security Profiles \(command line\)](#) for further information.

## SECS exit scheduling mechanisms

Gateway exit scheduling is based on user exit routine mechanisms that occur in the Supervisor Module (SUP) of the Gateway server. These mechanisms are referred to in Gateway documentation as *notif* mechanisms.

The following schema illustrates how the Gateway Supervisor Module (SUP) user exit mechanism (notif) schedules security exits.



1. If activated in the configuration, during the handshake process the SECS process calls an exit. The Supervisor process (SUP) registers the request and places it in a queue.
2. One of the SECS *notif* family user processes (SUP exit mechanisms) retrieves the request from the queue, processes it and returns a response to the Supervisor. To process the request, the *notif* process uses user-specified processing code.
3. The Supervisor then transmits the response to the request sender: SECS.

## Return values

A Security exit routine returns the following two types of value to the Gateway Supervisor (SUP) process:

- [Exit processing return code](#)
- [Application operation result values](#)

### Exit processing return code

Each exit routine returns an integer value. This value is a return code that indicates the general result of the exit execution. The following table lists the possible return values and their meanings.

| Value | Meaning                                                                                                                                                                      |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | Exit function execution is correct.                                                                                                                                          |
| 0     | No operation was carried out during the exit call. Gateway must use internal mechanisms to perform the function. This value is useful when you do not want to use all exits. |
| -1    | A fatal error occurred while executing the exit function. The current session is aborted.                                                                                    |

### Application operation result value: status and reason fields

The status field takes one of the following values:

| Value | Meaning                                                                                                                                                                                                              |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0     | The exit function result is OK (for example, the received certification path is correct, or the RSA computation returns an acceptable result).                                                                       |
| -1    | The exit function returned a value that leads to a disconnection or to a TLS alert. For example, from the accepted authorities DN list, it was impossible to build a valid certification path to send to the server. |

If the status field is set to -1, you can specify a reason for the disconnection (for example, the certification path received was too long). Recognized reason values are available in the `exitxdef.h` file and are exit-dependent. Gateway interprets recognized reason values to send TLS Alerts.

**Warning:** Do not confuse the status and reason fields with the overall return code defined above.

### Examples

`ExitSecsCheckCertList`

#### Example 1

The certificate verification process is performed via an access to an OCSP (Online Certificate Status Protocol ) server. This server is down and a connection timeout indicates that the server was unreachable. Your exit program returns an overall return code of -1.

#### Example 2

If both the OCSP server answers and the certification path are valid, the overall return code is 1 and the output status must be set to 0. There is no need to complete the reason field.

#### Example 3

If the server indicates that one of the path certificates was revoked, the overall return code is still 1 since there was no error while processing the verification. However, the output status must be set to -1. You then specify the reason as `ITP_SECS_CERT_REVOKED`, as defined in `secsdefs.h`, which you should include.

## *Shared parameters*

### **Session identifier: `session_identifier`**

To establish a correlation between various exits (exits called in the same TLS/SSL or SSH session), a session identifier is provided in each exit input structure in the `exitxdef.h` file. For each secured session, the session identifier remains the same throughout the duration of the session and while Gateway is running.

### **SECS user context: `secs_ucontext`**

A user context field is provided in each exit output and input structure (except for the `In_ExitSecsInit_t` structure). It enables you to maintain and modify your own session context.

This user context field can be modified at any step of the exit function call. This new value is taken into account and is provided in the next exit input structure.

Axway Gateway: Customizing Gateway Processes

## *TLS user exits*

This topic describes the following TLS user exits for SECS processes:

- [ExitSecsSessionInit](#)
- [ExitSecsSessionEnd](#)
- [ExitSecsCheckCertList](#)
- [ExitSecsComputation](#)
- [ExitSecsGetCertList](#)
- [ExitSecsBuildCaDnList](#)
- [ExitSecsHandshakeDone](#)
- [ExitSecsGetSrvPubKey](#)
- [ExitSecsSetSrvPubKey](#)

## *TLS user exits*

### **ExitSecsSessionInit**

## *Purpose*

This exit is called when the Security Profile to be used is located within the current TLS session. It indicates that a new TLS session is being processed.

**Caution** If you decide to use this exit to override the list of ciphers in the TLS security profiles, it is only possible to configure *eight* TLS ciphers. (Otherwise, you can configure 32 TLS cyphers; see *Security Guide* > [Managing TLS Security Profiles](#) for further information.

**Note:** This security exit enables you to override the Handshake session parameters taken from the current TLS Profile (sprof). You can select which parameters to override via a set of flags. These flags follow the naming convention:

A flag name is composed of the associated parameter name followed by the sequence `_p`.

**Example:** The flag `cache_enabled_p` corresponds to the output parameter `cache_enabled`.

A value of 1 for a flag indicates that the parameters taken from the exit output structure should override the corresponding parameters taken from the sprof object.

## Syntax

```
int ExitSecsSessionInit(In_ExitSecsSessionInit_t *in, Out_ExitSecsSessionInit_t *out)
```

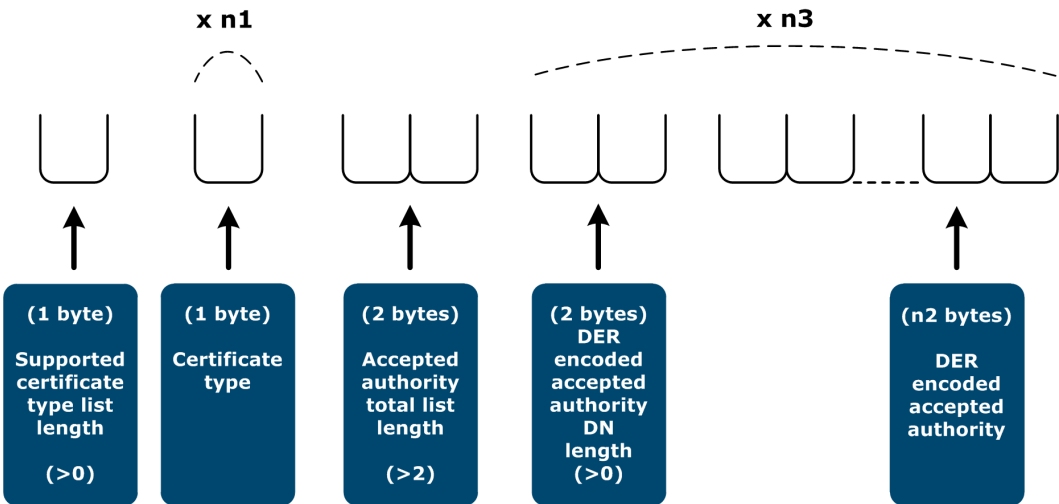
| ExitSecsSessionInit |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | InExitSecsInit_t *in                        |
| Output parameters   | OutExitSecsInit_t *out                      |
| Return              | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

### ExitSecsBuildCaDnList

## Purpose

This exit is only called in server mode and is intended to provide the data that is required to build the Certificate Request TLS PDU. Its output structure contains the DER encoded accepted authorities Distinguished Name list to send to the client. This list must be formatted as follows:



where:

- Byte 1: certificate type supported list length (>0)

Followed by one to n occurrences of:

- Byte 1: supported certificate type:
  - `rsa_sign = 1`
  - `dsa_sign = 2*`
  - `rsa_fixed_dh = 3*`
  - `dss_fixed_dh = 4*`

(\* = not yet supported by the internal manager)

Followed by:

- Byte 1 and 2: authority list total length (>2)

Followed by one to n occurrences of:

- Byte 1 and 2: DER encoded authority Distinguished Name (DN) length (>1)
- Byte 3 to n: DER encoded authority distinguished name

## Syntax

```
int ExitSecsBuildCaDnList

(InExitSecsBuildCaDnList_t *in, OutExitSecsBuildCaDnList_t *out)
```

| ExitSecsBuildCaDnList |                                             |
|-----------------------|---------------------------------------------|
| Input parameters      | InExitSecsBuildCaDnList_t*in                |
| Output parameters     | OutExitSecsBuildCaDnList_t *out             |
| Return                | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

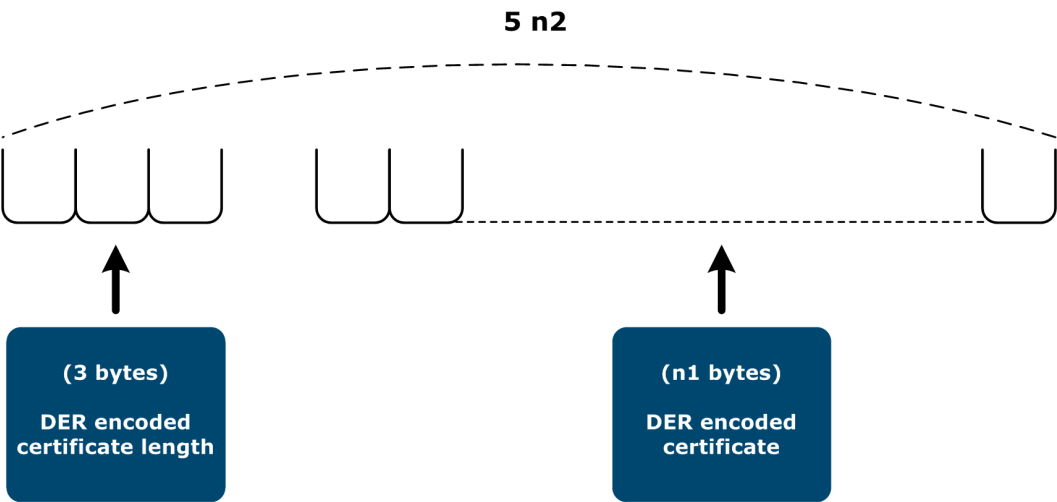
**ExitSecsGetCertList**

*Purpose*

Gateway calls this exit when it operates in either server or client mode. Use this exit to provide the certification chain to send to the remote partner.

In client mode, the accepted authorities distinguished name list sent by the server is supplied in the exit input structure.

The certification chain built (if possible) must be formatted as follows:



Where the bytes represent from 1 to n occurrences of:

- Byte 1 to 3: DER encoded certificate length
- Byte 4 to n: DER encoded certificate

**Note:** To use the internal product RSA computation mechanisms, complete the private key field in the OutExitSecsGetCertList\_t structure and assign the value 1 to the private\_key\_p flag.

*Syntax*

```
int ExitSecsGetCertList
(InExitSecsGetCertList_t *in, OutExitSecsGetCertList_t *out)
```

| ExitSecsGetCertList |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | In_ExitSecsSessionGetCert_t*in              |
| Output parameters   | Out_ExitSecsSessionnGetCert_t *out          |
| Return              | <a href="#">Exit processing return code</a> |



For more details, refer to [TLS security exit data structures](#).

**ExitSecsCheckCertList**

*Purpose*

This exit is called when receiving the remote certificate chain (client and server mode). This command is intended to process the certification path verification.

The certification path given follows the syntax described in the ExitSecsGetCertList definition.

The verification process is used to determine whether the remote partner identity is correct or not. The purpose of such a process is to accept the connection or to refuse it. As detailed in the TLS specification (RFC 2246), various reasons related to the certification path verification may lead to sending an alert (an alert is followed by a disconnection).

For this reason, the output reason field of this exit is mandatory. The available associated reason codes, in case of verification failure, are given in the file `secsdefs.h`.

*Syntax*

```
int ExitSecsCheckCertList(InExitSecsCheckCertList_t *in,
 OutExitSecsCheckCertList_t *out)
```

Input parameters

```
InExitSecsCheckCertList_t*in
```

| ExitSecsCheckCertList |                                             |
|-----------------------|---------------------------------------------|
| Input parameters      | InExitSecsCheckCertList_t*in                |
| Output parameters     | OutExitSecsCheckCertList_t *out             |
| Return                | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

**ExitSecsComputation**

*Purpose*

This exit is called at several steps of the negotiation, and is required when the certificate-related exits (ExitSecsGetCertList and ExitSecsCheckCertList) are called. Its purpose is to create the associated RSA (remote public key and local private key) operations that take part in the authentication process.

**Note:** To perform RSA computations (in both client and server mode), the corresponding cipher key must be available.

Since the security exits are mostly independent, you can schedule this exit whether the building of the certification path to send, and the checking of the certification path received are internally processed or not. It can be useful when hardware-encryption modules are available on the host machine.

## Syntax

```
int ExitSecsComputation
```

```
(InExitSecsComputation_t *in, OutExitSecsComputation_t *out)
```

| ExitSecsComputation |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | InExitSecsComputation_t*in                  |
| Output parameters   | OutExitSecsComputation_t *out               |
| Return              | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

### ExitSecsHandshakeDone

## Purpose

This exit is scheduled just after the TLS (or SSL) handshake negotiation is finished. It informs you of the security parameters used for the current session.

**Note:** Because this is an informative exit, only an overall return code value of 1 can lead to a disconnection.

## Syntax

```
int ExitSecsHandshakeDone
```

```
(InExitSecsHandshakeDone_t *in, OutExitSecsHandshakeDone_t *out)
```

| ExitSecsHandshakeDone |                                             |
|-----------------------|---------------------------------------------|
| Input parameters      | InExitSecsHandshakeDone_t *in               |
| Output parameters     | OutExitSecsHandshakeDone_t *out             |
| Return                | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

### ExitSecsSessionEnd

## Purpose

This exit is scheduled when the data transfer is entirely complete.

**Note:** This exit is used to indicate that the SECS server internal session context is being deleted.

## Syntax

```
int ExitSecsSessionEnd

(InExitSecsSessionEnd_t *in, OutExitSecsSessionEnd_t *out)
```

| ExitSecsSessionEnd |                                             |
|--------------------|---------------------------------------------|
| Input parameters   | InExitSecsSessionEnd_t*in                   |
| Output parameters  | OutExitSecsSessionEnd_t *out                |
| Return             | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

### ExitSecsGetSrvPubKey

## Syntax

```
int ExitSecsGetSrvPubKey

(InExitSecsGetSrvPubKey_t *in, OutExitSecsGetSrvPubKey_t *out)
```

| ExitSecsGetSrvPubKey |                                             |
|----------------------|---------------------------------------------|
| Input parameters     | InExitSecsGetSrvPubKey_t*in                 |
| Output parameters    | OutExitSecsGetSrvPubKey_t *out              |
| Return               | <a href="#">Exit processing return code</a> |

For more details, refer to [TLS security exit data structures](#).

### ExitSecsSetSrvPubKey

## Syntax

```
int ExitSecsSetSrvPubKey

(InExitSecsSetSrvPubKey_t *in, OutExitSecsSetSrvPubKey_t *out)
```

| ExitSecsSetSrvPubKey |                                             |
|----------------------|---------------------------------------------|
| Input parameters     | InExitSecsSetSrvPubKey_t*in                 |
| Output parameters    | OutExitSecsSetSrvPubKey_t *out              |
| Return               | <a href="#">Exit processing return code</a> |

For more details, refer to *TLS security exit data structures*.

## *TLS security exit data structures (exitxdef.h file)*

Here is an example extract of the TLS structures in an `exitxdef.h` header file.

```

/*****

 SECS : secs exits for TLS protocol

*****/
#define SECS_KEY_MAX 4096/8
/* SECS status */
typedef struct
{
 int status; /* Executed Exit status code : 0 or -1 */
 int reason; /* Applicative treatment result */
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX]; /* User Context */
 char message[ITP_SECS_MESSAGE_MAX];
} secs_status_t;
/*
 * SECS session initialization.
 */
typedef struct
{
 int session_identifier;
 int mode; /* ITP_SECS_CLIENT or
ITP_SECS_SERVER */
 char sprof[ITP_SPROF_NAME_MAX+1]; /* Current
security profile name */
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_
MAX]; /* User parameter taken from current sprof */
 unsigned char dest_address[NET_COMM_ADDRESS_MAX+1]; /*
RESPONDER : network called address */
 unsigned char src_address[NET_COMM_ADDRESS_MAX+1]; /*
RESPONDER : network calling address */
 unsigned char user_data[NET_COMM_USER_DATA_MAX]; /*
RESPONDER : X.25 user data */
 char xpp_entity_name[ITP_XPP_ENTITY_NAME_MAX+1]; /*
PassPort PS entity for certificates and keys */
 char xpp_entity_password[ITP_XPP_ENTITY_PASSWORD_MAX+1]; /*
PassPort PS entity password */
 char xpp_partner_entity_name[ITP_XPP_ENTITY_NAME_MAX+1];
/* PassPort PS partner entity */
 char xpp_address[ITP_HOST_NAME_MAX+1]; /*
PassPort PS server location */
 int xpp_port;
 int xpp_trace_level;
 int xps_trace_level;
} InExitSecsSessionInit_t;
typedef struct

```

```

{
 /* flags indicating whether overriding security profile
 parameters or not */
 short cache_enabled_p;
 short renegotiate_enabled_p;
 short close_notify_disabled_p;
 short client_auth_policy_p;
 short server_auth_policy_p;
 short cipher_suites_p;
 short __filler_p[4];
 secs_status_t status;
 short cache_enabled; /* 0=NO, 1=YES
*/
 short renegotiate_enabled; /* 0=NO, 1=YES
*/
 short close_notify_disabled; /* 0=NO, 1=YES
*/
 int client_auth_policy; /* MANDATORY,
OPTIONAL or ANONYMOUS */
 int server_auth_policy; /* MANDATORY,
OPTIONAL or ANONYMOUS */
 int nb_cipher_suites; /* cipher
suites element number */
 int cipher_suites[ITP_CIPHER_SUITES_MAX];
 char __filler[128];
} OutExitSecsSessionInit_t;
/*
 * SECS session termination.
 */
typedef struct
{
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 char __filler[128];
} InExitSecsSessionEnd_t;
typedef struct
{
 secs_status_t status;
 char __filler[128];
} OutExitSecsSessionEnd_t;
/*
 * SECS key
 */
typedef struct
{
 unsigned char modulo[SECS_KEY_MAX]; /* RSA key modulo */
 unsigned char private_exp[SECS_KEY_MAX]; /* RSA key private
exponent */

```

```

 unsigned char public_exp[SECS_KEY_MAX]; /* RSA key public
 exponent */
 } secs_rsa_key_t;
 typedef struct
 {
 int key_type; /* ITP_SECS_RSA_512, ITP_SECS_RSA_1024, ITP_
 SECS_RSA_2048 */
 union key_s
 {
 secs_rsa_key_t rsa_key;
 } key;
 } secs_key_t;
 /*
 * SECS certificates to send to partner
 */
 typedef struct
 {
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 int ca_list_len;
 unsigned char ca_list[ITP_CA_DN_LIST_MAX]; /* received DER
 encoded CA DN list buffer (client only) */
 char __filler[128];
 } InExitSecsGetCertList_t;
 typedef struct
 {
 short private_key_p; /* flag indicating that a private key
 is provided or not */
 short cert_list_p; /* flag indicating whether a cert
 list is provided or not */
 short external_cert_ref_p;
 short __filler_p[2];
 secs_status_t status;
 int cert_list_len; /* total length of the received DER
 encoded certificate chain */
 unsigned char cert_list[ITP_CERT_LIST_MAX]; /* received DER
 encoded certificate chain */
 char external_cert_ref[ITP_CERT_NAME_MAX+1];
 secs_key_t private_key; /* private Key to Use for internal
 RSA computation purpose */
 char __filler[128];
 } OutExitSecsGetCertList_t;
 /*
 * SECS partner certificates checking
 */
 typedef struct
 {
 int session_identifier;

```

```

 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 int cert_list_len; /* Total length of
the received DER encoded certificate chain
*/
 unsigned char cert_list[ITP_CERT_LIST_MAX]; /* Received
DER encoded certificate chain */
 char __filler[128];
} InExitSecsCheckCertList_t;

typedef struct
{
 short public_key_p; /* flag
indicating that a private key is provided or
not */
 short __filler_p[4];
 secs_status_t status;
 secs_key_t public_key; * Public key to use for
RSA computation */
 char xpp_partner_entity_name[ITP_XPP_ENTITY_NAME_
MAX+1];
 char __filler[128];
} OutExitSecsCheckCertList_t;
/*
 * SECS build accepted CA authorities DN list (server only)
 */
typedef struct
{
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
} InExitSecsBuildCaDnList_t;
typedef struct
{
 short ca_list_p;
 short __filler_p[2];
 secs_status_t status;
 int ca_list_len; /* Total length of the DER encoded
CA DN list to send */
 unsigned char ca_list[ITP_CA_DN_LIST_MAX]; /* DER encoded CA
DN list to send buffer */
 char __filler[128];
} OutExitSecsBuildCaDnList_t;
/*
 * SECS get server public key (server only)
 */

```

```

typedef struct
{
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 char __filler[128];
} InExitSecsGetSrvPubKey_t;
typedef struct
{
 short srv_pub_key_p; /* indicates that parameter is
provided */
 short __filler_p[2];
 secs_status_t status;
 secs_key_t srv_pub_key;
 char __filler[128];
} OutExitSecsGetSrvPubKey_t;
/*
 * SECS set server public key (client only)
 */
typedef struct
{
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 secs_key_t srv_pub_key;
 char __filler[128];
} InExitSecsSetSrvPubKey_t;
typedef struct
{
 secs_status_t status;
 char __filler[128];
} OutExitSecsSetSrvPubKey_t;
/*
 * SECS public or private key computation
 */
typedef struct
{
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 char external_cert_ref[ITP_CERT_NAME_MAX+1]; /* external
reference to the local certificate sent
 if the RSA
computation exit is to be scheduled */
 int key_mode; /* ITP_SECS_KEY_TYPE_PUB or ITP_
SECS_KEY_TYPE_PRV */

```



```

 int cipher_mode; /* ITP_SECS_MODE_ENCIPHER or ITP_
SECS_MODE_DECIPHER */
 int padding_type;
 int data_len;
 unsigned char data[ITP_COMPUTATION_DATA_MAX]; /* input data
to be computed */
 int hash_len;
 unsigned char hash[ITP_COMPUTATION_DATA_MAX]; /* hash may be
used when verifying signature (read only) */
 secs_key_t key;
 char __filler[128];
} InExitSecsComputation_t;
typedef struct
{
 short sign_verified_p;
 short data_p;
 short __filler_p[4];
 secs_status_t status;
 int sign_verified;
 int data_len; /* Length of computed
data */
 unsigned char data[ITP_COMPUTATION_DATA_MAX]; /* Result data
buffer */
 char __filler[128];
} OutExitSecsComputation_t;
/*
 * SECS handshake Done
 */
typedef struct
{
 int session_identifier;
 char sprof[ITP_SPROF_NAME_MAX+1];
 char secs_ucontext[ITP_SECS_UCONTEXT_MAX];
 char sprof_user_param[ITP_SECS_USER_PARAM_SIZE_MAX];
 int version; /* Protocol version used during the
Handshake */
 int cipher_suite; /* cipher suite used */
 int client_auth; /* client authentication level */
 int server_auth; /* server authentication level */
 int cache_used; /* Usage of session caching or not */
 char __filler[128];
} InExitSecsHandshakeDone_t;
typedef struct
{
 short subject_dn_p;
 short issuer_dn_p;
 short __filler_p[2];
 secs_status_t status;
 char subject_dn[DN_NAME_MAX+1]; /* remote subject dn */
 char issuer_dn[DN_NAME_MAX+1]; /* remote issuer dn */

```

```
char __filler[128];
} OutExitSecsHandshakeDone_t;
```

[About user exits](#)

[External user exits](#)

[User exits: Security functions](#)

[SSH user exits](#)

Axway Gateway: Customizing Gateway Processes

## SSH user exits

This topic describes SSH user exits for SECS processes. SSH exits can be divided into three groups:

| Group                                       | Exits                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Session management</a>          | <ul style="list-style-type: none"> <li>• <a href="#">ExitSechSessionInit</a></li> <li>• <a href="#">ExitSechSessionEnd</a></li> <li>• <a href="#">ExitSechSessionSuccess</a></li> </ul>                                                                                                                                                                     |
| <a href="#">Diffie Hellman key exchange</a> | <ul style="list-style-type: none"> <li>• <a href="#">ExitSechDhParamsGet</a></li> <li>• <a href="#">ExitSechDhParamsSet</a></li> <li>• <a href="#">ExitSechDhPubKeyGet</a></li> <li>• <a href="#">ExitSechDhComputation</a></li> </ul>                                                                                                                      |
| <a href="#">Authentication</a>              | <ul style="list-style-type: none"> <li>• <a href="#">ExitSechSignaturePerform</a></li> <li>• <a href="#">ExitSechSignatureVerify</a></li> <li>• <a href="#">ExitSechPublicParamsGet</a></li> <li>• <a href="#">ExitSechPublicParamsSet</a></li> <li>• <a href="#">ExitSechUserPasswordGet</a></li> <li>• <a href="#">ExitSechUserPasswordSet</a></li> </ul> |

### About SSH user exits

Gateway invokes SSH exits during the SSH negotiation phase of the session-layer handshake process.

An invoked SSH exit returns one of three possible values:

| Return value | Meaning                                               |
|--------------|-------------------------------------------------------|
| 0            | No action. The exit asks Gateway to proceed normally. |

|    |                                                                                                                                                                                                                                                   |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | The exit processes performed successfully. The exit asks Gateway to use the returned exit parameters.                                                                                                                                             |
| -1 | Fatal error. Aborts the current session handshake negotiation without invoking the ExitSecSessionEnd exit.<br><b>Note:</b> In general it is recommended that you do not return a -1 value. It is preferable to return a 1 value with a -1 status. |

## SSH Session management exits

### ExitSecSessionInit

#### Purpose

This exit is called when a new SSH session begins. The exit classes the session as *Client* or *Server*.

#### Syntax

```
int ExitSecshSessionInit(In_ExitSecshSessionInit_t *in, Out_ExitSecshSessionInit_t *out)
```

| ExitSecsSessionInit |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | InExitSecshSessionInit_t *in                |
| Output parameters   | OutExitSecshSessionInit_t *out              |
| Return              | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

### ExitSecSessionEnd

#### Purpose

This exit is called when an SSH session terminates. You can use this exit to release all allotted resources.

#### Syntax

```
int ExitSecshSessionEnd

(InExitSecshSessionEnd_t *in, OutExitSecshSessionEnd_t *out)
```

| ExitSecsSessionEnd |
|--------------------|
|--------------------|

|                   |                                             |
|-------------------|---------------------------------------------|
| Input parameters  | InExitSecshSessionEnd_t*in                  |
| Output parameters | OutExitSecshSessionEnd_t *out               |
| Return            | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## ExitSecSessionSuccess

### Purpose

This exit is called to indicate that the current SSH session has been successfully established. The exit receives the SSH algorithms used for negotiation as input parameters.

### Syntax

```
int ExitSecSessionSuccess
```

```
(InExitSecSessionSuccess_t *in, OutExitSecSessionSuccess_t *out)
```

| ExitSecSessionSuccess |                                             |
|-----------------------|---------------------------------------------|
| Input parameters      | InExitSecSessionSuccess_t*in                |
| Output parameters     | OutExitSecSessionSuccess_t *out             |
| Return                | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## SSH Diffie Hellman key exchange exits

Gateway calls key exchange exits at the beginning of key exchange negotiation. The practical value of these exits for users is limited. We recommend that you let Gateway handle negotiations with its standard set of processes.

For detailed information about the Diffie Hellman algorithm refer to the SSH transport layer protocol documentation at [www.ietf.com](http://www.ietf.com).

## ExitSecDhParamsGet

### Purpose

This exit is called to request the *p* and *g* parameters for the Diffie Hellman algorithm, to be sent to the remote partner. This exit is called only if the key exchange algorithm has the value **DH\_GROUP\_EXCHANGE\_SHA1** or **DH\_GROUP\_EXCHANGE\_SHA256**.

## Syntax

```
int ExitSechDhParamsGet
```

```
(InExitSechDhParamsGet_t *in, OutExitSechDhParamsGet_t *out)
```

| ExitSechDhParamsGet |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | InExitSechDhParamsGet_t*in                  |
| Output parameters   | OutExitSechDhParamsGet_t *out               |
| Return              | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

### ExitSechDhParamsSet

## Purpose

This exit is called to provide the *p* and *g* parameters for the Diffie Hellman algorithm, acquired from the remote partner. This exit is called only if the key exchange algorithm has the value `DH_GROUP_EXCHANGE_SHA1` or `DH_GROUP_EXCHANGE_SHA256`.

## Syntax

```
int ExitSechDhParamsSet
```

```
(InExitSechDhParamsSet_t *in, ExitSechDhParamsSet_t *out)
```

| ExitSechDhParamsSet |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | InExitSechDhParamsSet_t*in                  |
| Output parameters   | OutExitSechDhParamsSet_t *out               |
| Return              | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

### ExitSechDhPubKeyGet

## Purpose

This exit requests the Diffie Hellman public key that is sent to the remote partner.

## Syntax

```
int ExitSechDhPubKeyGet
```

```
(InExitSechDhPubKeyGet_t *in, OutExitSechDhPubKeyGet_t *out)
```

| ExitSechDhPubKeyGet |                                             |
|---------------------|---------------------------------------------|
| Input parameters    | InExitSechDhPubKeyGet_t*in                  |
| Output parameters   | OutExitSechDhPubKeyGet_t *out               |
| Return              | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

**ExitSechDhComputation**

*Purpose*

This exit requests the calculation of a shared secret via the public key provided as input.

*Syntax*

```
int ExitSechDhComputation
(InExitSechDhComputation_t *in, OutExitSechDhComputation_t *out)
```

| ExitSechDhComputation |                                             |
|-----------------------|---------------------------------------------|
| Input parameters      | InExitSecsComputation_t*in                  |
| Output parameters     | OutExitSecsComputation_t *out               |
| Return                | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

*Authentication exits*

**ExitSechSignaturePerform**

*Purpose*

This exit requests the return of a signature created via your private key using the hash furnished as input. According to the algorithm indicated, this may require a private RSA key, a private DSS key, a private RSA key or a private ECDSA key linked to your x509 certificate.

*Syntax*

```
int ExitSechSignaturePerform
(InExitSechSignaturePerform_t *in, OutExitSechSignaturePerform_t *out)
```

| ExitSechSignaturePerform |                                             |
|--------------------------|---------------------------------------------|
| Input parameters         | InExitSechSignaturePerform_t*in             |
| Output parameters        | OutExitSechSignaturePerform_t *out          |
| Return                   | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## ExitSechSignatureVerify

### Purpose

This exit requests a verification of the transfer partner certificate using the partner's public key. The verification result must correspond to the hash provided as input. According to the indicated algorithm, you provide the RSA public key, the DSS public key, the ECDSA public key or the partner's x509 certificate.

### Syntax

```
int ExitSechSignatureVerify
```

```
(InExitSechSignatureVerify_t *in, OutExitSechSignatureVerify_t *out)
```

| ExitSechSignatureVerify |                                             |
|-------------------------|---------------------------------------------|
| Input parameters        | InExitSechSignatureVerify_t*in              |
| Output parameters       | OutExitSechSignatureVerify_t *out           |
| Return                  | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## ExitSechPublicParamsGet

### Purpose

This exit requests your public parameters for forwarding to the exchange partner. According to the indicated algorithm, you provide your public RSA key, your public DSS key or your x509 certificate. In client mode, Gateway only calls this exit if the client authentication mode is *public key*. In mode server this exit is always called.

### Syntax

```
int ExitSechPublicParamsGet
```

```
(InExitSechPublicParamsGet_t *in, OutExitSechPublicParamsGet_t *out)
```

| ExitSechPublicParamsGet |                                             |
|-------------------------|---------------------------------------------|
| Input parameters        | InExitSechPublicParamsGet_t*in              |
| Output parameters       | OutExitSechPublicParamsGet_t *out           |
| Return                  | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## ExitSechPublicParamsSet

### Purpose

This exit provides the public parameters sent by the partner. According to the indicated algorithm, you obtain the partner public RSA key, the partner public DSS key or the partner x509 certificate. In client mode, Gateway always calls this exit. In server mode, Gateway calls the exit only when the authentication method is set to *public key*.

### Syntax

```
int ExitSechPublicParamsSet
```

```
(InExitSechPublicParamsSet_t *in, OutExitSechPublicParamsSet_t *out)
```

| ExitSechPublicParamsSet |                                             |
|-------------------------|---------------------------------------------|
| Input parameters        | InExitSechPublicParamsSet_t*in              |
| Output parameters       | OutExitSechPublicParamsSet_t *out           |
| Return                  | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## ExitSechUserPasswordGet

### Purpose

This exit requests the user name and user password to send to the partner server.

### Syntax

```
int ExitSechUserPasswordGet
```

```
(InExitSechUserPasswordGet_t *in, OutExitSechUserPasswordGet_t *out)
```

| ExitSechUserPasswordGet |                                |
|-------------------------|--------------------------------|
| Input parameters        | InExitSechUserPasswordGet_t*in |



|                   |                                             |
|-------------------|---------------------------------------------|
| Output parameters | OutExitSechUserPasswordGet_t *out           |
| Return            | <a href="#">Exit processing return code</a> |

For more information, refer to [SSH security exit data structures](#).

## ExitSechUserPasswordSet

### Purpose

This exit provides the user name and password sent by the partner client for verification.

Gateway only calls this exit when operating in server mode.

### Syntax

```
int ExitSechUserPasswordSet
```

```
(InExitSechUserPasswordSet_t *in, OutExitSechUserPasswordSet_t *out)
```

| ExitSechUserPasswordSet |                                             |
|-------------------------|---------------------------------------------|
| Input parameters        | InExitSechUserPasswordSet_t*in              |
| Output parameters       | OutExitSechUserPasswordSet_t *out           |
| Return                  | <a href="#">Exit processing return code</a> |

For more information, refer to *SSH security exit data structures*.

### Subsequent ssh exit calls

To pass custom user data in subsequent ssh exit calls, use the `SechStatus_t->userContext` field of the `OutExitSechSession*_t` structures. `ExitSechSession*` functions that fills this member, must return 1 for the value saved in `userContext`, to be taken into account and passed to subsequent exit calls.

All subsequent exits following the one that filled `userContext`, will receive that value in the `userContext` field of the **in** parameter (`InExitSech*_t` structures). If this value is changed, it has to be saved in the corresponding **out** parameter (see above) and the exit changing this value must return 1. Note that this value can be *changed* only in session-related exits, but it is *available* as input for all exits.

To pass this value to the transfer exits, store it in the output parameter of the `ExitSechSessionSuccess` function. The reason for this is that the file transfer context is aware of the connection only if the session was successfully established. Don't forget to return 1 from this function to be able to pass the value.

### SSH security exit data structures (exitxdef.h file)

Here is an example extract of the SSH structures in an `exitxdef.h` header file.

```

 SECH : secs exits for SSH protocol
 *****/
 /*
 * SECH status
 */
typedef struct
{
 int status; /* exit code : 0 = ok, -1 = error */
 int reason; /* applicative processing result */
 char userContext[ITP_SECS_UCONTEXT_MAX];
} SechStatus_t;
/*
 * SECH session initialization
 */
typedef struct
{
 int sessionIdentifier;
 int connectionEnd; /* SSH_CLIENT or SSH_SERVER */
 char sshProf[ITP_SSH_PROF_NAME_MAX+1]; /* current ssh profile
name */
 /* client only : user name and password from the site */
 char userName[ITP_USER_NAME_MAX+1];
 char password[ITP_PASSWORD_MAX+1];
 /* client only : the called remote site */
 char siteName[ITP_SITE_NAME_MAX+1];
 /* server only : network called address */
 unsigned char destAddress[NET_COMM_ADDRESS_MAX+1];
 /* server only : network calling address */
 unsigned char srcAddress[NET_COMM_ADDRESS_MAX+1];
 /* server only : X.25 user data */
 unsigned char userData[NET_COMM_USER_DATA_MAX];
 /* PassPort PS server location */
 unsigned char xppAddress[NET_ADDRESS_MAX+1];
 int xppPort;
} InExitSechSessionInit_t;
typedef struct
{
 /* flags indicating whether overriding security profile
parameters or not */
 short securityOption_p;
 short version_p;
 short keyExchangeAlgos_p;
 short cipherAlgos_p;
 short macAlgos_p;
 short compressionAlgos_p;
 short publicKeyAlgos_p;
 short clientAuthMethods_p;
 short performAllAuthMethods_p;

```

```

 short userName_p;
 short MBytesBeforeNewKeys_p;
 short minutesBeforeNewKeys_p;
 short __filler_p[2];
 SechStatus_t status;
 int securityOption; /* none or ssh
*/
 int version; /* protocol
version */
 int nbKeyExchangeAlgos; /* nb value in
list below */
 int keyExchangeAlgos[ITP_SSH_MAX_KEY_EXCHANGE]; /* key exchange
algorithms */
 int nbCipherAlgos; /* nb value in
list below */
 int cipherAlgos[ITP_SSH_MAX_CIPHER]; /* encryptions
algorithms */
 int nbMacAlgos; /* nb value in
list below */
 int macAlgos[ITP_SSH_MAX_MAC]; /* mac
algorithms */
 int nbCompressionAlgos; /* nb value in
list below */
 int compressionAlgos[ITP_SSH_MAX_COMPRESSION]; /* compression
algorithms */
 int nbPublicKeyAlgos; /* nb value in
list below */
 int publicKeyAlgos[ITP_SSH_MAX_PUBLIC_KEY]; /* public key
algorithms */
 int nbClientAuthMethods; /* nb value in
list below */
 int clientAuthMethods[ITP_SSH_MAX_AUTH_METHOD]; /* client auth
methods */
 int performAllAuthMethods; /* all methods
required */
 char userName[ITP_USER_NAME_MAX+1]; /* user name,
client only */
 int MBytesBeforeNewKeys; /* MBytes to
receive or send before negotiating new keys */
 int minutesBeforeNewKeys; /* Minutes to
wait before negotiating new keys */
 char __filler[248];
} OutExitSechSessionInit_t;
/*
* SECH session ending
*/
typedef struct
{
 int sessionIdentifier;
 char __filler[128];
} InExitSechSessionEnd_t;
typedef struct

```

```
{
 SechStatus_t status;
 char __filler[128];
} OutExitSechSessionEnd_t;
/*
 * SECH session success
 */
typedef struct
{
 int sessionIdentifier;
 int keyExchangeAlgo;
 int publicKeyAlgo;
 int cipherAlgoCS;
 int cipherAlgoSC;
 int macAlgoCS;
 int macAlgoSC;
 int compressionAlgoCS;
 int compressionAlgoSC;
 char __filler[128];
} InExitSechSessionSuccess_t;
typedef struct
{
 SechStatus_t status;
 char __filler[128];
} OutExitSechSessionSuccess_t;
/*
 * SECH Diffie Hellmann parameters get
 */
typedef struct
{
 int sessionIdentifier;
 char __filler[128];
} InExitSechDhParamsGet_t;
typedef struct
{
 SechStatus_t status;
 int pLen;
 unsigned char p[ITP_DH_KEY_MAX];
 unsigned char g;
 char __filler[128];
} OutExitSechDhParamsGet_t;
/*
 * SECH Diffie Hellmann parameters set
 */
typedef struct
{
 int sessionIdentifier;
 int pLen;
 unsigned char p[ITP_DH_KEY_MAX];
```

```
 unsigned char g;
 char __filler[128];
} InExitSechDhParamsSet_t;
typedef struct
{
 SechStatus_t status;
 char __filler[128];
} OutExitSechDhParamsSet_t;
/*
 * SECH Diffie Hellmann public key get
 */
typedef struct
{
 int sessionIdentifier;
 char versionString[ITP_SSH_VERSION_STRING_MAX+1];
 char __filler[128];
} InExitSechDhPubKeyGet_t;
typedef struct
{
 SechStatus_t status;
 int pubKeyLen;
 unsigned char pubKey[ITP_DH_KEY_MAX];
 char __filler[128];
} OutExitSechDhPubKeyGet_t;
/*
 * SECH Diffie Hellmann computation
 */
typedef struct
{
 int sessionIdentifier;
 int pubKeyLen;
 unsigned char pubKey[ITP_DH_KEY_MAX];
 char __filler[128];
} InExitSechDhComputation_t;
typedef struct
{
 SechStatus_t status;
 int sharedSecretLen;
 unsigned char sharedSecret[ITP_DH_KEY_MAX];
 char __filler[128];
} OutExitSechDhComputation_t;
/*
 * SECH signature perform
 */
typedef struct
{
 int sessionIdentifier;
 int algorithmIdentifier;
 int hashLen;
```

```
 unsigned char hash[ITP_HASH_MAX];
 char __filler[128];
} InExitSechSignaturePerform_t;
typedef struct
{
 SechStatus_t status;
 int algorithmIdentifier;
 union
 {
 struct
 {
 int signatureLen;
 unsigned char signature[ITP_RSA_SIGN_MAX];
 } rsa;
 struct
 {
 int rLen;
 unsigned char r[ITP_DSS_SIGN_MAX];
 int sLen;
 unsigned char s[ITP_DSS_SIGN_MAX];
 } dss;
 struct
 {
 int signatureLen;
 unsigned char signature[ITP_ECDSA_SIGN_MAX];
 } ecdsa;
 } value;
 char __filler[128];
} OutExitSechSignaturePerform_t;
/*
 * SECH signature verify
 */
typedef struct
{
 int sessionIdentifier;
 int algorithmIdentifier;
 union
 {
 struct
 {
 int signatureLen;
 unsigned char signature[ITP_RSA_SIGN_MAX];
 } rsa;
 struct
 {
 int rLen;
 unsigned char r[ITP_DSS_SIGN_MAX];
 int sLen;
```

```
unsigned char s[ITP_DSS_SIGN_MAX];
} dss;

struct
{
int signatureLen;
unsigned char signature[ITP_ECDSA_SIGN_MAX];
} ecdsa;
} value;
int hashLen;
unsigned char hash[ITP_HASH_MAX];
char __filler[256];
} InExitSechSignatureVerify_t;
typedef struct
{
 SechStatus_t status;
 int algorithmIdentifier;
 int isSignatureVerified;
 char __filler[128];
} OutExitSechSignatureVerify_t;
/*
 * SECH public parameters get
 */
typedef struct
{
 int sessionIdentifier;
 int algorithmIdentifier;
 char __filler[128];
} InExitSechPublicParamsGet_t;
typedef struct
{
 SechStatus_t status;
 char userName[ITP_USER_NAME_MAX+1];
 int algorithmIdentifier;
 union
 {
 struct
 {
 int nLen;
 unsigned char n[ITP_RSA_KEY_MAX]; /* modulus */
 int eLen;
 unsigned char e[ITP_RSA_KEY_MAX]; /* public exponent */
 } rsa;
 struct
 {
 int pLen;
 unsigned char p[ITP_DSS_KEY_MAX]; /* prime modulus */
 int qLen;
```

```

unsigned char q[ITP_DSS_KEY_MOD_MAX]; /* prime divisor */
int gLen;
unsigned char g[ITP_DSS_KEY_MAX]; /* order of q mod p */
int yLen;
unsigned char y[ITP_DSS_KEY_MAX]; /* public key */
} dss;

struct
{
int pointLen;
unsigned char point[2 * ITP_ECDSA_KEY_MAX + 1]; /* public key -
Point coordinates (x and y) */
} ecdsa;

struct
{
int certificateLen;
unsigned char certificate[ITP_CERT_LIST_MAX];
} x509v3Sign;
} value;

char __filler[256];
} OutExitSechPublicParamsGet_t;
/*
* SECH public parameters set
*/

typedef struct
{
int sessionIdentifier;
char userName[ITP_USER_NAME_MAX+1];
int algorithmIdentifier;
union
{
struct
{
int nLen;
unsigned char n[ITP_RSA_KEY_MAX]; /* modulus */
int eLen;
unsigned char e[ITP_RSA_KEY_MAX]; /* public exponent */
} rsa;
struct
{
int pLen;
unsigned char p[ITP_DSS_KEY_MAX]; /* prime modulus */
int qLen;
unsigned char q[ITP_DSS_KEY_MOD_MAX]; /* prime divisor */
int gLen;
unsigned char g[ITP_DSS_KEY_MAX]; /* order of q mod p */
int yLen;
unsigned char y[ITP_DSS_KEY_MAX]; /* public key */

```



```
} dss;

struct
{
int certificateLen;
unsigned char certificate[ITP_CERT_LIST_MAX];
} x509v3Sign;

struct
{
int pointLen;
unsigned char point[2 * ITP_ECDSA_KEY_MAX + 1]; /* point (x and
y)*/
} ecdsa;
} value;

char __filler[256];
} InExitSechPublicParamsSet_t;
typedef struct
{
 SechStatus_t status;
 char __filler[128];
} OutExitSechPublicParamsSet_t;
/*
 * SECH user password get
 */
typedef struct
{
 int sessionIdentifier;
 char __filler[128];
} InExitSechUserPasswordGet_t;
typedef struct
{
 SechStatus_t status;
 char userName[ITP_USER_NAME_MAX+1];
 char password[ITP_PASSWORD_MAX+1];
 char __filler[128];
} OutExitSechUserPasswordGet_t;
/*
 * SECH user password set
 */
typedef struct
{
 int sessionIdentifier;
 char userName[ITP_USER_NAME_MAX+1];
 char password[ITP_PASSWORD_MAX+1];
 char __filler[128];
} InExitSechUserPasswordSet_t;
typedef struct
{
 SechStatus_t status;
```

```

char __filler[128];
} OutExitSechUserPasswordSet_t;
#endif /* _EXITXDEF_H */

```

Axway Gateway: Customizing Gateway Processes

## Compiling C user exits

**Caution** Please note that care should be taken when **customizing user exits**, as there are some **security risks** involved if the custom code is not properly reviewed and security best practices are not followed. Please see the *Security Guide > Security best practices > [Identified threats and possible mitigations](#)* section for additional information.

This topic describes how to compile the C language external user exits that you modify for use with Gateway.

This information is intended for application programmers familiar with the C programming language. It does not discuss the C language, the C run-time library or the C execution environment (compiler and system dependencies). Refer to public reference manuals on C language, and to your platform documentation, for detailed information on this programming language.

### Exit file locations

| File<br>(extension)    | Location                                                   |
|------------------------|------------------------------------------------------------|
| Exit header files (.h) | <Gateway installation directory>/inc                       |
| Exit source files (.c) | sub-directories in<br><Gateway installation directory>/src |

### How to compile and link C user exits: Windows

Go to the directory <Gateway installation directory>\run\_time\etc\ and use the script `profile.bat` to load the appropriate environment variables for the operating system.

In the directory where the exit file was modified, execute:

```
nmake /f Makefile
```

### *How to compile and link C user exits: UNIX*

Go to the directory `<Gateway installation directory>/src` and use the script `userprof.sh` to load the appropriate environment variables for the operating system.

If you want to compile and link all the exits to Gateway programs, enter:

```
userbuild.sh
```

If you want to compile only part of them, enter:

```
make xxxxxx
```

where:

xxxxxx = one of the following:

- `exitclnt`
- `exituser`
- `exitsecs`
- `exitsech`
- `exitc`
- `exitrte`
- `perlstub`

Axway Gateway: Customizing Gateway Processes

## C API primitives

This topic introduces the C API primitives supplied with Gateway.

### Interface header

The `gikapic.h` header file contains the API primitive prototypes. This file is located in `<Gateway installation directory>/inc`.

### List of API primitives

The following table lists the available API primitives.

Follow the links for details about using these primitives.

| API primitive                                                               | Usage                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">GikSetTraceLevel</a>                                            | Use this function to set the desired trace level.                                                                                                                                                                                                                                                                                  |
| <a href="#">GikSessionOpen</a><br><a href="#">GikSessionOpenFromNotif</a>   | <p>Use the GikSessionOpen function to open the communication session. This function provides access security features, to use the interface in client/server mode. You must call the function before you use any other API primitive.</p> <p>Use the GikSessionOpenFromNotif function to open the session from external exits.</p> |
| <a href="#">GikSessionClose</a><br><a href="#">GikSessionCloseFromNotif</a> | <p>Use these functions to close the communication session and releases all resources. The last Transfer Request locked is automatically unlocked.</p> <p>Use the GikSessionCloseFromNotif function to close a session opened with GikSessionOpenFromNotif.</p>                                                                     |
| <a href="#">GikErrMsg</a>                                                   | When an error occurs, API primitives return <b>1</b> and set an error code in the global variable ErrCod. This function returns the error message text that corresponds to the value of ErrCod.                                                                                                                                    |
| <a href="#">GikGetErrCod</a>                                                | When an error occurs, API primitives return <b>1</b> and set an error code in the global variable ErrCod. This function returns the value of ErrCod.                                                                                                                                                                               |
| <a href="#">GikInitParams</a>                                               | Use this function to retrieve a handle on an empty parameter list.                                                                                                                                                                                                                                                                 |
| <a href="#">GikFreeParams</a>                                               | Use this function to free a handle on a parameter list.                                                                                                                                                                                                                                                                            |
| <a href="#">GikSetxxx</a>                                                   | Use the GikSetxxx series of functions to set the value of a field in the parameters list.                                                                                                                                                                                                                                          |
| <a href="#">GikGetxxx</a>                                                   | Use the GikGetxxx series of functions to retrieve the value of a field from the parameter list.                                                                                                                                                                                                                                    |
| <a href="#">GikXferPut</a>                                                  | Use this function to submit a Transfer Request.                                                                                                                                                                                                                                                                                    |
| <a href="#">GikXferGet</a>                                                  | Use this function to read the Transfer Request that you specify by its identifier <i>ident</i> , and copy the Transfer parameters in the parameter list <i>xferParams</i> .                                                                                                                                                        |
| <a href="#">GikXferUpdate</a>                                               | Use this function to modify a Transfer Request.                                                                                                                                                                                                                                                                                    |

| API primitive                     | Usage                                                                                                                                                                                                |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">GikXferDelete</a>     | Use this function to delete a Transfer Request.                                                                                                                                                      |
| <a href="#">GikXferUnlock</a>     | Use this function to unlock a Transfer Request that was previously locked during a read operation.                                                                                                   |
| <a href="#">GikXferOpenSet</a>    | Use this function to select a set of Transfer Requests, depending on the criteria specified in the parameter list <code>selectParams</code> .                                                        |
| <a href="#">GikXferGetNext</a>    | Use this function to read the Transfer Requests previously selected by the <code>GikXferOpenSet</code> primitive. It copies the Transfer parameters to the parameters list <code>xferParams</code> . |
| <a href="#">GikXferCloseSet</a>   | Use this function to close the set of Transfer Requests previously selected using the <code>GikXferOpenSet</code> primitive.                                                                         |
| <a href="#">GikXferModifState</a> | Use this function to modify the state of the Transfer identified by the <code>ident</code> parameter.                                                                                                |

Axway Gateway: Customizing Gateway Processes

## C API primitives

[Error reporting on API primitives](#)

[API primitive descriptions](#)

### *Error reporting on API primitives*

All primitives (except `GikErrMsg`) return the following:

- **0** if the call is successful
- **1** if an error is detected:
  - the error value is indicated in a global variable, `ErrCod`
  - the `GikErrMsg` function returns the text of the error
  - the `ErrCod` variable is declared and automatically set by the `GikAPI` functions

## *API primitive descriptions*

### *GikSetTraceLevel*

#### **Syntax**

```
int GikSetTraceLevel (short traceLevel)
```

#### **Description**

Use the GikSetTraceLevel function to set the desired trace level. The `traceLevel` parameter can be:

- `GIK_NO_TRACE`: No trace activated. Only fatal errors are displayed.
- `GIK_WARNINGS`: Warnings are displayed.
- `GIK_INFOS`: Information and warnings are displayed.

Errors and warnings are displayed in the standard error output "stderr".

Information is displayed in the standard output "stdout".

**Note:** If you do not call GikSetTraceLevel, the trace level is set to `GIK_NO_TRACE`.

### *GikSessionOpen / GikSessionOpenFromNotif*

#### **Syntax**

```
int GikSessionOpen (char *cs_profile_name, char *user, char *password,
char *new_password, void **session)
```

#### **Description**

The GikSessionOpen function opens the communication session. This function provides access security features, to use the interface in client/server mode. You must call this function before you use any other API primitive.

Use the GikSessionOpenFromNotif function to open the session from external exits.

GikSessionClose() must be called to close the session at the end of the program.

Use the GikSessionCloseFromNotif function to close a session opened with GikSessionOpenFromNotif.

The parameter `cs_profile_name` identifies the section in the client `CSCONFIG.INI` file that describes the target server. If you open the session locally, this parameter is ignored.

#### **Example**

The following section in the `CONFIG.INI` file on the Gateway client station describes the Gateway "Gateway\_1":

```
[Gateway_1]
HostName=193.56.234.29
Port=23000
```

If the client station wants to open a session with this Gateway, the `cs_profile_name` must be `Gateway_1`.

If not NULL, the parameter `new_password` is used to modify the user password.

**Note:** The returned session handle must be supplied in calls to other primitives.

## *GikSessionClose / GikSessionCloseFromNotif*

### **Syntax**

```
int GikSessionClose (void *session)
```

### **Description**

This function closes the communication session and releases all resources.

The last Transfer Request locked is automatically unlocked.

**Note:** No API primitive is available after a call to this function.

Use the function `GikSessionCloseFromNotif` to close a session opened using `GikSessionOpenFromNotif`.

## *GikErrMsg*

### **Syntax**

```
char *GikErrMsg()
```

### **Description**

When an error occurs, API primitives return **1** and set an error code in the global variable `ErrCod`. This function returns the error message text that corresponds to the value of `ErrCod`.

**Note:** It is recommended that you copy the contents of the buffer into your program space. A subsequent call to `GikErrMsg()` overrides the contents of the buffer.

### **Return value**

The return value is a pointer to a Gateway internal static buffer containing the text that explains the error.

## *GikGetErrCod*

### **Syntax**

```
int GikGetErrCod ()
```

### **Description**

When an error occurs, API primitives return **1** and set an error code in the global variable `ErrCod`. This function returns the value of `ErrCod`.

### **Return value**

The return value is the error code.

## *GikInitParams*

### **Syntax**

```
int GikInitParams (short type, void **params)
```

### **Description**

This function retrieves a handle on an empty parameter list.

Parameter types can be:

- GIK\_TRANSFER Creates parameters list for use in `GikXferPut`, `GikXferGet`, `GikXferGetNext` or `GikXferUpdate` requests
- GIK\_SELECTION Creates parameters list for use in `GikXferOpenSet` request

Calls to `GikSetxxx` or `GikGetxxx` primitives fill or read the parameter list.

**Note:** You must call `GikFreeParams` to free the handle on the list.

## *GikFreeParams*

### **Syntax**

```
int GikFreeParams (void *params)
```

### **Description**

This function frees a handle on a parameter list.



## GikSetxxx

### Syntax

```
int GikSetInt (void *params, int fieldIdent, int value)
int GikSetUInt (void *params, int fieldIdent, unsigned int value)
int GikSetShort (void *params, int fieldIdent, short value)
int GikSetUShort (void *params, int fieldIdent, unsigned short
value)
int GikSetLong (void *params, int fieldIdent, long value)
int GikSetULong (void *params, int fieldIdent, unsigned long
value)
int GikSetChar (void *params, int fieldIdent, char value)
int GikSetUChar (void *params, int fieldIdent, unsigned char
value)
int GikSetString (void *params, int fieldIdent, char *value)
int GikSetUString (void *params, int fieldIdent, unsigned char
*value)
int GikSetBuff (void *params, int fieldIdent, void *ptr, int
length)
int GikSetTime (void *params, int fieldIdent, time_t value)
int GikSetOffset (void *params, int fieldIdent, Offset_t value)
```

### Description

The GikSetxxx series of functions set the value of a field in the parameters list. The `fieldIdent` parameter specifies the field to set. The function used must be compatible with the type of the field that you specify. Refer to [C API Transfer and Selection parameters](#).

GikSetBuff needs a pointer to the buffer and the buffer length (in bytes). You must:

- Allocate the buffer
- Free the buffer by calling GikFreeParams

Before you call GikSetxxx functions, you must call GikInitParams to retrieve the handle on the parameter list.

## GikGetxxx

### Syntax

```
int GikGetInt (void *params, int fieldIdent, int *value)
int GikGetUInt (void *params, int fieldIdent, unsigned int *value)
int GikGetShort (void *params, int fieldIdent, short *value)
int GikGetUShort (void *params, int fieldIdent, unsigned short
*value)
int GikGetLong (void *params, int fieldIdent, long *value)
```

```
int GikGetULong (void *params, int fieldIdent, unsigned long
*value)
int GikGetChar (void *params, int fieldIdent, char *value)
int GikGetUChar (void *params, int fieldIdent, unsigned char
*value)
int GikGetString (void *params, int fieldIdent, char *value, int
length)
int GikGetUString (void *params, int fieldIdent, unsigned char
*value, int length)
int GikGetBuff (void *params, int fieldIdent, void *ptr, int
*length)
int GikGetBuffSize (void *params, int fieldIdent, int *size)
int GikGetTime (void *params, int fieldIdent, time_t *value)
int GikGetOffset (void *params, int fieldIdent, Offset_t *value)
```

### Description

The GikGetxxx series of functions retrieve the value of a field from the parameter list. The `fieldIdent` parameter specifies the field to retrieve. The data type of the field to retrieve determines which GikGetxxx function to use. Refer to [C API Transfer and Selection parameters](#).

- **GikGetBuff**

- *ptr*: buffer that the user allocates
- *length*: allocated size in bytes

GikGetBuff completes the *ptr* buffer on condition that the specified *length* is sufficient. On return, it updates *length* with the actual size used. If the given size *length* is not sufficient, it returns an error.

Before calling GikGetBuff with `fieldIdent` GIK\_T\_MSG\_BUF, you must set the GIK\_T\_MSG\_MAX parameter. To set GIK\_T\_MSG\_MAX, specify the maximum buffer size you can retrieve for the message.

- **GikGetBuffSize** retrieves the buffer size in bytes.
- **GikGetString** and **GikGetUString**:
  - *value*: buffer that the user allocates
  - *length*: allocated size. If this size is not sufficient, an error is returned.

## GikXferPut

### Syntax

```
int GikXferPut (void *session, void *xferParams, GikXferIdent_t *ident)
```

### Description

This function submits a Transfer Request.

The Transfer identifier is returned in the parameter `ident`. This identifier is used for subsequent operations on the Request.

Before calling the function, the user program must create the parameter list `xferParams` (using the `GikInitParams` primitive), and complete it (using `GikSetxxx` primitives). Refer to [C API Transfer and Selection parameters](#).

## *GikXferGet*

### Syntax

```
int GikXferGet (void *session, int options, GikXferIdent_t *ident, void
*xferParams)
```

### Description

This function reads the Transfer Request that you specify by its identifier `ident`, and copies the Transfer parameters in the parameter list `xferParams`.

Before calling this function:

- The user program must create the parameter list *xferParams* using the `GikInitParams` primitive
- You must initialize the arguments `ident` (Transfer Request identifier) and *options* (read option)

The *options* parameter (`XFER_LOCK` or 0) indicates whether the **read** operation is locked or not. If the "read" operation is followed by an update or delete operation, the lock option is mandatory.

You can only lock one Transfer record at a time. A record is automatically unlocked when:

- You lock another Transfer record
- You perform an update or delete operation successfully

## *GikXferUpdate*

### Syntax

```
int GikXferUpdate (void *session, void *xferParams)
```

### Description

This function modifies a Transfer Request.

You can only call this function after a successful call to a read function (using `GikXferGet` or `GikXferGetNext`) with the lock option enabled. To enable the lock option for a read operation, set the *option* parameter to `XFER_LOCK`.

The `XFER_LOCK` option prevents concurrent user access to the Transfer record. If another user initiates a read operation on a locked Transfer record, the error code `E_FLOCKED` is returned.

You can only lock one Transfer record at a time. A record is automatically unlocked when:

- You lock another Transfer record
- You successfully update the record

After reading the Transfer Request, the user program uses the GikSetxxx primitives to modify the parameter list obtained (xferParams), and calls GikXferUpdate. Refer to [C API Transfer and Selection parameters](#).

**Note:** If the Transfer state is ITP\_XFER\_ACKED, ITP\_XFER\_ENDED or ITP\_XFER\_CANCELED, you can only modify the GIK\_S\_USER\_STATE parameter.

## *GikXferDelete*

### **Syntax**

```
int GikXferDelete (void *session)
```

### **Description**

This function deletes a Transfer Request.

You can only call this function after a successful call to a read function (using GikXferGet or GikXferGetNext) with the lock option enabled. To enable the lock option for a read operation, set the *option* parameter to XFER\_LOCK.

The XFER\_LOCK option prevents concurrent user access to the Transfer record. If another user initiates a read operation on a locked Transfer record, the error code E\_FLOCKED is returned.

If you delete a Transfer Request during the file transfer, the transfer aborts and the Request is deleted.

## *GikXferUnlock*

### **Syntax**

```
int GikXferUnlock (void *session)
```

### **Description**

This function unlocks a Transfer Request that was previously locked during a read operation.

## *GikXferOpenSet*

### **Syntax**

```
int GikXferOpenSet (void *session, void *selectParams)
```

### Description

This function is used to select a set of Transfer Requests, depending on the criteria specified in the parameter list `selectParams`. Refer to [C API Transfer and Selection parameters](#).

To select all Transfers, call `GikInitParams` (to initialize the list `selectParams` ), but do not set any criteria with `GikSetxxx` primitives.

**Note:** After a successful selection, calls to `GikXferGetNext` sequentially read the selected Transfer Requests.

After a successful call to `GikXferGetNext`, you can call the following primitives:

- `GikXferGetNext` - to get the next record of the selection
- `GikXferDelete` - to delete the Transfer Request that was read
- `GikXferUpdate` - to update the Transfer Request that was read
- `GikXferCloseSet` - to close the selection

## *GikXferGetNext*

### Syntax

```
int GikXferGetNext (void *session, int options, void *xferParams)
```

### Description

This function reads the Transfer Requests previously selected by the `GikXferOpenSet` primitive. It copies the Transfer parameters to the parameters list `xferParams`. Refer to [C API Transfer and Selection parameters](#).

Before calling this function:

- The user program must create the parameter list `xferParams` with the `GikInitParams` primitive
- You must declare and specify the *options* parameter (read option)

The *options* parameter (`XFER_LOCK` or 0) indicates whether the read operation is locked or not. If the read operation is followed by an update or delete operation, the lock option is mandatory.

You can only lock one Transfer record at a time. A record is automatically unlocked when:

- You lock another Transfer record
- You successfully update or delete the record

**Note:** The error code `E_EOF` means that no more Transfer records are available.

## *GikXferCloseSet*

### **Syntax**

```
int GikXferCloseSet (void *session)
```

### **Description**

This function closes the set of Transfer Requests previously selected using the `GikXferOpenSet` primitive.

## *GikXferModifState*

### **Syntax**

```
int GikXferModifState (void *session, GikXferIdent_t *ident, short
action);
```

### **Description**

This function modifies the state of the Transfer identified by the `ident` parameter.

The Transfer Request does not have to be locked for you to perform this operation.

The *action* parameter can be:

- CANCEL
- SUSPEND
- RESUME (to reactivate a suspended Transfer or to repeat a previous Transfer)

If the *action* parameter does not correspond to one of these states, the function returns the error code `E_WRONGSTATE` (Invalid Transfer state).

Axway Gateway: Customizing Gateway Processes

## **C API usage examples**

[Overview](#)

[Transfers](#)

[Messages](#)

### *Overview*

Use C API primitive calls to implement actions on Transfers and messages.

User programs always call:

- **GikSessionOpen()** primitive to start communication with Gateway
- **GikSessionOpenFromNotif()** primitive to use call the API from external exits
- **GikSessionClose()** primitive to end communication
- **GikSessionCloseFromNotif()** primitive to close a session that was opened with GikSessionOpenFromNotif()

## Transfers

### Submitting Transfers

Use the following primitives to submit Transfers:

| Primitive                                                                                                                        | Purpose                                                         |
|----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| GikSessionOpen(cs_profile_name, user, password, &session)                                                                        | To open the communication session                               |
| GikInitParams(GIK_TRANSFER, &params)                                                                                             | To initialize a parameters list                                 |
| GikSetxxx(params, fieldIdent_1, value_1)<br>GikSetxxx(params, fieldIdent_i, value_i)<br>GikSetxxx(params, fieldIdent_n, value_n) | To set the value of the fieldIdent field in the parameters list |
| GikXferPut(session, params, &ident)                                                                                              | To retrieve the Transfer identifier                             |
| GikFreeParams(params)                                                                                                            | To free the handle on the parameters list                       |
| GikSessionClose(session)                                                                                                         | To close the session                                            |

### Retrieving Transfer parameters

Use the following primitives to retrieve Transfer parameters:

| Primitive                                                                                                                           | Purpose                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GikSessionOpen(cs_profile_name, user, password, &session)                                                                           | To open the communication session                                                                                                                                                                                                                                                                                                                                                     |
| GikInitParams(GIK_TRANSFER, &params)                                                                                                | To retrieve a parameters list                                                                                                                                                                                                                                                                                                                                                         |
| GikXferGet(session, option, &ident, params)                                                                                         | <p>To retrieve a Transfer Request and copy the Transfer parameters to the parameters list <i>xferParams</i>.</p> <p>The value of:</p> <ul style="list-style-type: none"> <li><i>option</i> is XFER_LOCK or 0</li> <li><i>ident</i> is the Transfer Request identifier supplied by the user</li> <li><i>params</i> is the parameters list built by the GikXferGet primitive</li> </ul> |
| GikGetxxx(params, fieldIdent_1, &value_1)<br>GikGetxxx(params, fieldIdent_i, &value_i)<br>GikGetxxx(params, fieldIdent_n, &value_n) | To retrieve the value of the fieldIdent field from the parameters list                                                                                                                                                                                                                                                                                                                |
| GikFreeParams(params)                                                                                                               | To free the handle on the parameters list                                                                                                                                                                                                                                                                                                                                             |
| GikSessionClose(session)                                                                                                            | To close the session                                                                                                                                                                                                                                                                                                                                                                  |

## Updating Transfers

Use the following primitives to update Transfers:

| Primitive                                                 | Purpose                                     |
|-----------------------------------------------------------|---------------------------------------------|
| GikSessionOpen(cs_profile_name, user, password, &session) | To open the session                         |
| GikInitParams(GIK_TRANSFER, &params)                      | To retrieve the handle on a parameters list |



| Primitive                                                                                                                        | Purpose                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GikXferGet(session, option, &ident, params)                                                                                      | <p>To retrieve a Transfer Request and copy the Transfer parameters to the parameters list <i>xferParams</i>.</p> <p>The value of:</p> <ul style="list-style-type: none"> <li><i>option</i> is XFER_LOCK</li> <li><i>ident</i> is the Transfer Request identifier supplied by the user</li> <li><i>params</i> is the parameters list built by the GikXferGet primitive</li> </ul> |
| GikSetxxx(params, fieldIdent_1, value_1)<br>GikSetxxx(params, fieldIdent_i, value_i)<br>GikSetxxx(params, fieldIdent_n, value_n) | To update the value of the fieldIdent field in the parameters list                                                                                                                                                                                                                                                                                                               |
| GikXferUpdate(session, params)                                                                                                   | To update the last Transfer Request retrieved by GikXferGet (or GikXferGetNext)                                                                                                                                                                                                                                                                                                  |
| GikFreeParams(params)                                                                                                            | To free the handle on the parameters list                                                                                                                                                                                                                                                                                                                                        |
| GikSessionClose(session)                                                                                                         | To close the session                                                                                                                                                                                                                                                                                                                                                             |

### Deleting Transfers from the Mailbox

Use the following primitives to delete Transfers from the Mailbox:

| Primitive                                                 | Purpose                                                                                                                                                                          |
|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GikSessionOpen(cs_profile_name, user, password, &session) | To open a session                                                                                                                                                                |
| GikInitParams(GIK_TRANSFER, &xferParams)                  | <p>To initialize the parameters list for use in:</p> <ul style="list-style-type: none"> <li>GikXferPut</li> <li>GikXferGet</li> <li>GikGetNext</li> <li>GikXferUpdate</li> </ul> |
| GikInitParams(GIK_SELECTION, &selectParams)               | To initialize the parameters list for use in GikXferOpenSet                                                                                                                      |

| Primitive                                    | Purpose                                                                                                           |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| GikXferOpenSet(session, selectParams)        | To select a set of Transfer Requests                                                                              |
| GikXferGetNext(session, options, xferParams) | To read the Transfer Requests previously selected by GikXferOpenSet.<br>The value of <i>options</i> is XFER_LOCK. |
| GikXferDelete(session)                       | To delete the last Transfer Request retrieved by GikXferGet (or GikXferGetNext).                                  |
| GikXferCloseSet(session)                     | To close the set of Transfer Requests previously opened by GikXferOpenSet.                                        |
| GikFreeParams(xferParams)                    | To free parameters list                                                                                           |
| GikFreeParams(selectParams)                  | To free parameters list                                                                                           |
| GikSessionClose(session)                     | To close the session                                                                                              |

## Suspending Transfers

Use the following primitives to suspend Transfers:

| Primitive                                                  | Purpose                                                                                                                                                                                                                         |
|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GikSessionOpen (cs_profile_name, user, password, &session) | To open a session. The Transfer Request must be in one of the following states: <ul style="list-style-type: none"> <li>• XFER_TO_BEGIN</li> <li>• XFER_PROGRESSING</li> <li>• XFER_SERVICING</li> <li>• XFER_DELAYED</li> </ul> |
| GikXferModifState(session, &ident, SUSPEND)                | To modify the state of the Transfer Request that you identify by the parameter <i>ident</i> .                                                                                                                                   |
| GikSessionClose(session)                                   | To close the session                                                                                                                                                                                                            |

## Messages

### Submitting message requests via a buffer

Use the following primitives to submit message requests via a buffer:

| Primitive                                                                                                                        | Purpose                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| GikSessionOpen(cs_profile_name, user, password, &session)                                                                        | To open a session                                                                                   |
| GikInitParams(GIK_TRANSFER, &params)                                                                                             | To initialize a parameters list                                                                     |
| GikSetxxx(params, fieldIdent_1, value_1)<br>GikSetxxx(params, fieldIdent_i, value_i)<br>GikSetxxx(params, fieldIdent_n, value_n) | To set the value of the fieldIdent field in the parameters list                                     |
| GikSetBuff(params, GIK_T_MSG_BUF, ptr, length)                                                                                   | GikSetBuff is used to allocate buffer space. The parameter must be previously allocated and filled. |
| GikXferPut(session, params, &ident)                                                                                              | To retrieve the identifier of the Transfer to submit                                                |
| GikFreeParams(params)                                                                                                            | To free the parameters                                                                              |
| free(ptr)                                                                                                                        | The parameter can be freed after a call to GikFreeParams                                            |
| GikSessionClose(session)                                                                                                         | To close the session                                                                                |

### *Retrieving message requests via a buffer*

Use the following primitives to retrieve message requests via a buffer:

| Primitive                                                  | Purpose                                                                              |
|------------------------------------------------------------|--------------------------------------------------------------------------------------|
| GikSessionOpen (cs_profile_name, user, password, &session) | To open a session                                                                    |
| GikInitParams (GIK_TRANSFER, &params)                      | To initialize a parameters list                                                      |
| GikSetInt (params, GIK_T_MSG_MAX, max)                     | To set the maximum message size that you can retrieve, via the <i>max</i> parameter. |

| Primitive                                                         | Purpose                                                                                                                                 |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| GikXferGet (session, option, &ident, params)                      | To retrieve the identifier of the Transfer Request to read                                                                              |
| GikGetxxx (params, fieldIdent_1, &value_1)                        | To retrieve Transfer Request parameters and buffer size.                                                                                |
| GikGetxxx (params, fieldIdent_i, &value_i)                        | The length (size allocated in ptr) must be less than <i>max</i> and greater than (or equal to) <i>size</i> . It is often equal to size. |
| GikGetxxx (params, fieldIdent_n, &value_n)                        |                                                                                                                                         |
| GikGetBuffSize (params, GIK_T_MSG_BUF, size); (allocation of ptr) |                                                                                                                                         |
| GikGetBuff (params, GIK_T_MSG_BUF, ptr, length)                   |                                                                                                                                         |
| GikFreeParams (params)                                            | To free the parameters list                                                                                                             |
| GikSessionClose(session)                                          | To close the session                                                                                                                    |

Axway Gateway: Customizing Gateway Processes

## C API Transfer and Selection parameters

Transfer parameters:

- [Gateway transfer parameters](#)
- [Scheduling transfer parameters](#)
- [File/Transfer organization transfer parameters](#)
- [Options transfer parameters](#)
- [Message transfer parameters](#)
- [PeSIT transfer parameters](#)
- [FTP transfer parameters](#)
- [SMTP/POP3 transfer parameters](#)
- [X.400 transfer parameters](#)
- [SFTP transfer parameters](#)
- [Trading Partner transfer parameters](#)
- [Sentinel transfer parameters](#)
- [Transfer parameters used \*only\* with PUT](#)
- [Transfer parameters used \*only\* with GET](#)

Selection parameters:

- [Gateway selection parameters](#)

## Transfer parameters

### Gateway transfer parameters

| FieldIdent                                           | Type   | Max String              |
|------------------------------------------------------|--------|-------------------------|
| GIK_T_ORG_ALIAS                                      | String | ITP_SITE_ALIAS_MAX      |
| GIK_T_DEST_ALIAS                                     | String | ITP_SITE_ALIAS_MAX      |
| GIK_T_ORIGINATOR                                     | String | ITP_SITE_NAME_MAX       |
| GIK_T_DESTINATION                                    | String | ITP_SITE_NAME_MAX       |
| GIK_T_LOCAL_AGENT                                    | String | ITP_SITE_NAME_MAX       |
| GIK_T_REMOTE_AGENT                                   | String | ITP_SITE_NAME_MAX       |
| GIK_T_APPLI                                          | String | ITP_PROTO_FILE_NAME_MAX |
| GIK_T_DIR_PATH                                       | String | ITP_DIR_PATH_MAX        |
| GIK_T_FILE_COMPONENT                                 | String | ITP_FILE_NAME_MAX       |
| GIK_T_PATHNAME (match dir_<br>path + file_component) | String | ITP_FILE_PATH_MAX       |
| GIK_T_REPLY_TO_XFER                                  | Long   |                         |
| GIK_T_TYPE                                           | Short  |                         |
| GIK_T_USER_STATE                                     | Uchar  |                         |
| GIK_T_MODE                                           | Short  |                         |
| GIK_T_DIRECTION                                      | Short  |                         |
| GIK_T_COMPRESSION                                    | Short  |                         |
| GIK_T_FILE_TYPE                                      | String | ITP_PROTO_FILE_TYPE_MAX |
| GIK_T_FILE_NAME                                      | String | ITP_PROTO_FILE_NAME_MAX |
| GIK_T_PARAM1                                         | String | ITP_USER_PARAM_MAX      |

| FieldIdent            | Type   | Max String           |
|-----------------------|--------|----------------------|
| GIK_T_PARAM2          | String | ITP_USER_PARAM_MAX   |
| GIK_T_NACK_ACTION     | Uchar  |                      |
| GIK_T_REQUEST_METHOD  | Char   |                      |
| GIK_T_DIR_NAME        | String | ITP_FILE_PATH_MAX    |
| GIK_T_ROUTE_FROM_XFER | Long   |                      |
| GIK_T_CONNECTOR_ID    | String | ITP_CONNECTOR_ID_MAX |
| GIK_T_MODEL           | String | ITP_MODEL_NAME_MAX   |

### *Scheduling transfer parameters*

| FieldIdent             | Type   |
|------------------------|--------|
| GIK_T_YDAY             | Short  |
| GIK_T_DATE_TO_BEGIN    | time_t |
| GIK_T_DATE_TO_END      | time_t |
| GIK_T_HIST_CREATE_DATE | time_t |
| GIK_T_PRIORITY         | Short  |
| GIK_T_RETRY_COUNT_MAX  | Short  |
| GIK_T_INTERVAL         | Long   |
| GIK_T_PERMANENT        | Uchar  |

### *File/Transfer organization transfer parameters*

| FieldIdent     | Type  | Max String |
|----------------|-------|------------|
| GIK_T_FILE_ORG | Short |            |
| GIK_T_REC_FMT  | Short |            |
| GIK_T_REC_LEN  | Long  |            |

| FieldIdent               | Type   | Max String                 |
|--------------------------|--------|----------------------------|
| GIK_T_SYS_DEP            | String | ITP_FILE_SYS_DEP_PARAM_MAX |
| GIK_T_BLOCK_SIZE         | Long   |                            |
| GIK_T_DATA_CODE          | Short  |                            |
| GIK_T_PADDING            | Short  |                            |
| GIK_T_X_REC_FMT          | Short  |                            |
| GIK_T_X_REC_LEN          | Long   |                            |
| GIK_T_X_DATA_CODE        | Short  |                            |
| GIK_T_KEY_LEN            | Long   |                            |
| GIK_T_KEY_OFFSET         | Long   |                            |
| GIK_T_TEXT_FILE          | Uchar  |                            |
| GIK_T_TRUNCATING_ALLOWED | Uchar  |                            |
| GIK_T_FILE_LABEL         | String | ITP_FILE_LABEL_MAX         |

### *Options transfer parameters*

| FieldIdent                  | Type  |
|-----------------------------|-------|
| GIK_T_ACKNOWLEDGMENT_OPTION | Uchar |
| GIK_T_PURGE_OPTION          | Uchar |

### *Message transfer parameters*

| FieldIdent    | Type            | Max String        |
|---------------|-----------------|-------------------|
| GIK_T_MSG_BUF | C=unsigned char |                   |
| GIK_T_MSG_MAX | Int             |                   |
| GIK_T_SND_MSG | String          | ITP_USER_MSGX_MAX |

*PeSIT transfer parameters*

| FieldIdent        | Type   | Max String          |
|-------------------|--------|---------------------|
| GIK_T_SND_USER    | String | ITP_PESIT_USER_MAX  |
| GIK_T_SND_APPLI   | String | ITP_PESIT_APPLI_MAX |
| GIK_T_SND_TEXT    | String | ITP_PESIT_TEXT_MAX  |
| GIK_T_RCV_USER    | String | ITP_PESIT_USER_MAX  |
| GIK_T_RCV_APPLI   | String | ITP_PESIT_APPLI_MAX |
| GIK_T_RCV_TEXT    | String | ITP_PESIT_TEXT_MAX  |
| GIK_T_MAX_REQUEST | Short  |                     |
| GIK_T_MAX_FILE    | Short  |                     |

*FTP transfer parameters*

| FieldIdent          | Type   | Max String           |
|---------------------|--------|----------------------|
| GIK_T_FTP_COMMAND   | String | ITP_USER_COMMAND_MAX |
| GIK_T_APPEND        | Uchar  |                      |
| GIK_T_UNIQ_FILENAME | Uchar  |                      |

*SMTP/POP3 transfer parameters*

| FieldIdent                 | Type   | Max String           |
|----------------------------|--------|----------------------|
| GIK_T_MAIL_SUBJECT         | String | ITP_MAIL_SUBJECT_MAX |
| GIK_T_ATTACH_EXTRACT_FILE  | Int    |                      |
| GIK_T_GENERATE_XFB_SUBJECT | Int    |                      |
| GIK_T_FROM                 | String | ITP_MAIL_ADDRESS_MAX |
| GIK_T_TO                   | String | ITP_MAIL_ADDRESS_MAX |



| FieldIdent | Type   | Max String           |
|------------|--------|----------------------|
| GIK_T_CC   | String | ITP_MAIL_ADDRESS_MAX |
| GIK_T_BCC  | String | ITP_MAIL_ADDRESS_MAX |

### *X.400 transfer parameters*

| FieldIdent                           | Type   | Max String            |
|--------------------------------------|--------|-----------------------|
| GIK_T_X420_ORG_ALIAS                 | String | ITP_SITE_NAME_MAX     |
| GIK_T_X420_DEST_ALIAS                | String | ITP_SITE_NAME_MAX     |
| GIK_T_X400_ORIGINATOR                | String |                       |
| GIK_T_X400_RECIPIENT                 | String |                       |
| GIK_T_X400_DN_REQUEST                | Uchar  |                       |
| GIK_T_X400_PRIORITY                  | short  |                       |
| GIK_T_X420_IPMID_USER                | String |                       |
| GIK_T_X420_IPMID_USER_ID             | String | X420_USER_RELATIVE_ID |
| GIK_T_X420_BODY_PART_TYPE            | Uchar  |                       |
| GIK_T_X420_SUBJECT                   | String | ITP_MAIL_SUBJECT_MAX  |
| GIK_T_X420_ORIGINATOR_FORMAL_NAME    | String |                       |
| GIK_T_X420_ORIGINATOR_FREE_FORM_NAME | String |                       |
| GIK_T_X420_ORIGINATOR_TEL_NUMBER     | String |                       |
| GIK_T_X420_RECIPIENT_FORMAL_NAME     | String |                       |
| GIK_T_X420_RECIPIENT_FREE_FORM_NAME  | String |                       |
| GIK_T_X420_RECIPIENT_TEL_NUMBER      | String |                       |

| FieldIdent                                                          | Type   | Max String          |
|---------------------------------------------------------------------|--------|---------------------|
| GIK_T_X420_RECIPIENT_RECEIPT_NOTIF_REQ                              | Uchar  |                     |
| GIK_T_X420_RECIPIENT_REPLY_REQUEST                                  | Uchar  |                     |
| GIK_T_X420_REPLY_TO_IPM_ID_USER                                     | String |                     |
| GIK_T_X420_REPLY_TO_IPM_ID_USER_ID                                  | String |                     |
| GIK_T_X420_EXPIRE_DATE_AND_TIME                                     | time_t |                     |
| GIK_T_X420_IMPORTANCE                                               | Uchar  |                     |
| GIK_T_X420_SENSITIVITY                                              | Uchar  |                     |
| <b>The following parameters are read-only (used only with GET):</b> |        |                     |
| GIK_T_X400_CONTENT_ID                                               | String | X400_CONTENT_ID_MAX |
| GIK_T_X420_UA_RN_ID                                                 | String | X420_UA_RN_ID_MAX   |
| GIK_T_X400_MTS_ID                                                   | String | X400_MTS_ID_MAX     |
| GIK_T_X420_REPLY_DATE_AND_TIME                                      | time_t |                     |
| GIK_T_X400_TRANSFER_SIZE                                            | Ulong  |                     |
| GIK_T_X400_TRANSFER_START_TIME                                      | Ulong  |                     |
| GIK_T_X400_TRANSFER_TIME                                            | String |                     |
| GIK_T_X400_SUBMISSION_TIME                                          | time_t |                     |
| GIK_T_X420_AUTO_FORWARD                                             | String |                     |
| GIK_T_X400_SUBJECT_MTS_ID                                           | String |                     |

*SFTP transfer parameters*

| FieldIdent               | Type |
|--------------------------|------|
| GIK_T_NEWLINE_CONVENTION | Char |

*Trading Partner transfer parameters*

| FieldIdent                     | Type   |
|--------------------------------|--------|
| GIK_T_RECEIPT_REPLY_TO_XFER    | long   |
| GIK_T_RECEIPT_TYPE             | int    |
| GIK_T_RECEIPT_NACK_DESC        | String |
| GIK_T_RECEIPT_REQ              | int    |
| GIK_T_RECEIPT_REQ_SYNC         | int    |
| GIK_T_RECEIPT_REQ_TO           | String |
| GIK_T_RECEIPT_SUBTYPE          | int    |
| GIK_T_RECEIPT_SEND_MODE        | int    |
| GIK_T_RECEIPT_REPLY_BY_XFER    | long   |
| GIK_T_TRADE_FORMAT             | int    |
| GIK_T_TRADE_SIGN               | int    |
| GIK_T_TRADE_CIPHER             | int    |
| GIK_T_TRADE_COMPRESS           | int    |
| GIK_T_TRADE_ORG                | String |
| GIK_T_TRADE_DEST               | String |
| GIK_T_CONTENT_TYPE             | String |
| GIK_T_TRADE_STATE              | int    |
| GIK_T_TRADE_CIPHERING_ALGO     | int    |
| GIK_T_TRADE_KEY_CIPHERING_ALGO | int    |

| FieldIdent                       | Type   |
|----------------------------------|--------|
| GIK_T_TRADE_COMPRESSION_ALGO     | int    |
| GIK_T_TRADE_SIGNING_ALGO         | int    |
| GIK_T_TRADE_RECEIPT_SIGNING_ALGO | int    |
| GIK_T_TRADE_REPLY_BY_XFER        | long   |
| GIK_T_TRADE_MIC_ALGO             | int    |
| GIK_T_TRADE_MIC                  | String |
| GIK_T_TRADE_MSG_ID               | String |
| GIK_T_TRADE_RECEIPT_RECIPIENT    | String |

### *Sentinel transfer parameters*

| FieldIdent                     | Type |
|--------------------------------|------|
| GIK_T_SENTINEL_TRANSFER_FILTER | Char |

### *Transfer parameters used only with PUT*

| FieldIdent                   | Type   | Max String              |
|------------------------------|--------|-------------------------|
| GIK_T_FROZEN_STATE           | Uchar  |                         |
| GIK_T_DIR_NAME_ALIAS         | String | X_VFD_DIR_ALIAS_LEN     |
| GIK_T_EX_SENTINEL_IDENT      | String | ITP_E_TRKIDENT_MAX      |
| GIK_T_EX_SENTINEL_OBJECTNAME | String | ITP_E_TRKOBJECTNAME_MAX |

### *Transfer parameters used only with GET*

| FieldIdent                     | Type | Max String |
|--------------------------------|------|------------|
| <b>Supervision information</b> |      |            |
| GIK_T_LOCAL_IDENT              | Long |            |

| FieldIdent                    | Type     | Max String         |
|-------------------------------|----------|--------------------|
| GIK_T_DATE_CREATE             | time_t   |                    |
| GIK_T_SEQUENCE                | Short    |                    |
| GIK_T_USER_PROCESSED          | Uchar    |                    |
| GIK_T_SELECT_REQ              | Long     |                    |
| GIK_T_REC_COUNT (file_att)    | Long     |                    |
| GIK_T_FILE_SIZE (file_att)    | offset_t |                    |
| GIK_T_EXTRACT_DATE (file_att) | time_t   |                    |
| GIK_T_CIPHER_TYPE             | Uchar    |                    |
| GIK_T_KEY_MATERIAL            | Ustring  |                    |
| GIK_T_GROUP_NAME              | String   | ITP_GROUP_NAME_MAX |
| <b>Dynamic information</b>    |          |                    |
| GIK_T_STATE                   | Short    |                    |
| GIK_T_RETRY_COUNT             | Short    |                    |
| GIK_T_X_BYTES                 | Offset_t |                    |
| GIK_T_F_BYTES                 | Offset_t |                    |
| GIK_T_DATE_BEGIN              | time_t   |                    |
| GIK_T_DATE_END                | time_t   |                    |
| GIK_T_LAST_END_REASON         | Int      |                    |
| GIK_T_LAST_END_DIAG           | Int      |                    |
| GIK_T_LAST_END_ERR            | Int      |                    |
| GIK_T_TOTAL_REQUEST           | Short    |                    |
| GIK_T_TOTAL_FILES             | Short    |                    |
| GIK_T_SPROF (Security)        | String   | SPROF_NAME_MAX     |

| FieldIdent                       | Type   | Max String           |
|----------------------------------|--------|----------------------|
| GIK_T_CIPHER_SUITE<br>(Security) | Int    |                      |
| GIK_T_CLIENT_AUTH<br>(Security)  | Uchar  |                      |
| GIK_T_SERVER_AUTH<br>(Security)  | Uchar  |                      |
| <b>User</b>                      |        |                      |
| GIK_T_USERNAME                   | String | USER_NAME_MAX        |
| GIK_T_END_XFER_SCRIPT            | String | FILE_NAME_MAX        |
| GIK_T_RCV_MSG                    | String | USER_MSGX_MAX        |
| <b>Diffusion List</b>            |        |                      |
| GIK_T_MASTER                     | Long   |                      |
| GIK_T_NEXT_XFER                  | Long   |                      |
| GIK_T_NB_ENDED                   | Ushort |                      |
| GIK_T_NB_TO_BEGIN                | Ushort |                      |
| GIK_T_NB_CANCELED                | Ushort |                      |
| GIK_T_NB_SUSPENDED               | Ushort |                      |
| GIK_T_NB_FROZEN                  | Ushort |                      |
| GIK_T_NB_PROGRESSING             | Ushort |                      |
| GIK_T_NB_OTHERS                  | Ushort |                      |
| GIK_T_NB_TOTAL                   | Ushort |                      |
| <b>Protocol information</b>      |        |                      |
| GIK_T_XFER_IDENT                 | String | PROTO_XFER_IDENT_MAX |
| GIK_T_PROTOCOL                   | Int    |                      |
| <b>Status information</b>        |        |                      |

| FieldIdent           | Type   | Max String      |
|----------------------|--------|-----------------|
| GIK_T_CALLED_ADDRESS | String | NET_ADDRESS_MAX |
| GIK_T_CALLED_SAP     | String | NET_SAP_MAX     |
| GIK_T_HTTP_HOST_NAME | String | HOST_NAME_MAX   |

## *Selection parameters*

### *Gateway selection parameters*

| FieldIdent             | Type                                 | Max String                             |
|------------------------|--------------------------------------|----------------------------------------|
| GIK_S_LOCAL_IDENT      | Long                                 |                                        |
| GIK_S_IDENT_INF        | Long                                 |                                        |
| GIK_S_IDENT_SUP        | Long                                 |                                        |
| GIK_S_PROTOCOL         | Int                                  |                                        |
| GIK_S_ORG_ALIAS        | String                               | PROTO_IDENT_MAX                        |
| GIK_S_DEST_ALIAS       | String                               | PROTO_IDENT_MAX                        |
| GIK_S_YDAY_INF         | Short                                |                                        |
| GIK_S_YDAY_SUP         | Short                                |                                        |
| GIK_S_SEQUENCE_INF     | Short                                |                                        |
| GIK_S_SEQUENCE_SUP     | Short                                |                                        |
| GIK_S_FROM_DATE        | time_t                               |                                        |
| GIK_S_TO_DATE          | time_t                               |                                        |
| GIK_S_LIST_TRANS_STATE | short* <b>Buff</b><br>Int (in bytes) | XFER_LIST_LEN (Max number<br>of short) |
| GIK_S_DIRECTION        | Short                                |                                        |
| GIK_S_FILE_TYPE        | String                               | PROTO_FILE_TYPE_MAX                    |
| GIK_S_FILE_NAME        | String                               | PROTO_FILE_NAME_MAX                    |

| FieldIdent           | Type   | Max String               |
|----------------------|--------|--------------------------|
| GIK_S_USERNAME       | String | USER_NAME_MAX            |
| GIK_S_APPLI          | String | APPLI_NAME_MAX           |
| GIK_S_USER_STATE     | Uchar  |                          |
| GIK_S_TYPE           | Short  |                          |
| GIK_S_MASTER         | Long   |                          |
| GIK_S_PROTO_IDENT    | String | ITP_PROTO_ID_MAX         |
| GIK_S_FIRSTREC       | Short  |                          |
| GIK_S_MODE           | Short  |                          |
| GIK_S_BEGIN_PRIORITY | Short  |                          |
| GIK_S_END_PRIORITY   | Short  |                          |
| GIK_S_BEGIN_BEFORE   | time_t |                          |
| GIK_S_BEGIN_AFTER    | time_t |                          |
| GIK_S_END_BEFORE     | time_t |                          |
| GIK_S_END_AFTER      | time_t |                          |
| GIK_S_PERMANENT      | Uchar  |                          |
| GIK_S_PARAM1         | String | ITP_USER_PARAM1_NAME_MAX |
| GIK_S_PARAM2         | String | ITP_USER_PARAM2_NAME_MAX |
| GIK_S_ITEM           | String | ITP_FILE_NAME_MAX        |

Axway Gateway: Customizing Gateway Processes

## Code conversion and file formatting

[Overview](#)[Defining code conversion and formatting parameters](#)



[Code conversion](#)

[Formatting and padding](#)

[Examples](#)

## Overview

This topic describes how to use code conversion and file formatting utilities in Gateway. These features enable you to change the data code of file contents or the record size in the file.

- **Code conversion** involves translating the character code. Use code conversion when you transfer a file between two platforms that support different character codes (for example, UNIX/ASCII to MVS/EBCDIC).
- **Formatting** involves changing file record:
  - Format (from fixed to variable or vice versa)
  - Size (adding or removing padding characters where necessary)

You define code conversion and formatting requirements when you create an Application or a Transfer Request. Gateway then converts and formats the file dynamically during the transfer.

This topic also describes the two code conversion utilities supplied with Gateway:

- [trncc](#) – to compile a text file table
- [trnxcode](#) – enables you to convert the contents of a file manually

## Defining code conversion and formatting parameters

You define code conversion and formatting parameters in the Application object and in Transfer Requests that do not use an Application.

Gateway dynamically requests code conversion and/or formatting when the following local file attributes are different from the corresponding transfer attributes:

- Record format
- Record length
- Data encryption
- Padding character

The functions operate differently depending on whether you are sending or receiving a file.

## *Sending a file*

When you send a file, the *local file attributes* are the file attributes defined by the file management system of the sending platform. Define these attributes in the **Local attributes** section of a Transfer Request or Application.

The *transfer attributes* and the *padding character attributes* are the attributes to apply to the file contents during the transfer. Define these attributes in the **Transfer attributes** section of a Transfer Request or Application.

If you do not specify the transfer attributes, they default to the local file attributes and there is no code conversion or formatting.

## *Receiving a file*

When you receive a file, the *transfer attributes* are the attributes of the received file. On transfers using PeSIT, the transfer attributes are completed by the protocol. For all other protocols you must define the attributes in the Application object.

You apply the *local file attributes* and the *padding character attributes* to the file on the receiving platform. Define these attributes in the incoming Transfer Request or in the Application object.

If you do not specify the local file attributes, they default to the transfer attributes and there is no code conversion or formatting.

## Code conversion

A code is the numeric representation of a character set (alphabet + other symbols). All computers use a given code to represent the characters displayed (ASCII, EBCDIC...). When you transfer a text file between two machines, it is recommended that the character set of the received file corresponds to the code of the receiving machine. If the codes used on the sending and receiving machines are different, code conversion is required.

To convert codes, Gateway uses conversion tables. By default, Gateway supplies the following tables:

- `ascii.ebcdic` table (to convert from ASCII to EBCDIC)
- `ebcdic.ascii` table (to convert from EBCDIC to ASCII)

Additionally, you can define up to 12 code conversion tables of your own.

The conversion tables are described in the `table.trn` text file. You can add or change the data contained in this file using a text editor.

Before using conversion tables, use the `trncc` utility to compile the `table.trn` text file.

## How conversion tables work

A conversion table maps the hexadecimal representation of characters in the input code to an output code. For example, the `ascii.ebcdic` table maps `41` to `C1` (respectively, the ASCII and EBCDIC codes for the alphabetic character `A`).

## Table organization

A table is basically a one-dimensional array. If the value of the character in the input code is  $n$ , its value in the output code is in position  $n$  in the table. For example the EBCDIC code for `A` is in position `65` (`41` in hexadecimal) of the `ascii.ebcdic` table.

## Syntax of code conversion tables

For each conversion table, the `table.trn` text file contains blocks of information in the following format:

```
table <codein.codeout>
comments
default = 3F,N # 3F is the ASCII code of the character '?'
fill = 40,N # 40 is the ASCII code of the character '@'
error = 21,N # 21 is the ASCII code of the character '!'
data
[code table]
end
```

Where:

- `codein` is the input code name
- `codeout` is the output code name
- `table`, `data` and `end` are mandatory keywords
- `default`, `fill` and `error` are optional keywords
- The only acceptable code names are:
  - `ascii`
  - `ascii1`
  - `ascii2`
  - `ascii3`
  - `ascii4`
  - `ascii5`,
  - `ebcdic`
  - `ebcdic1`
  - `ebcdic2`

- ebcdic3
  - ebcdic4
  - ebcdic5
  - oempc
  - iso
- Characters following the hash symbol ( # ) in a line, contain comment information
  - `fill`: The value defined in `fill` represents the code attributed to all non-specified characters. In other words any characters encountered during the transcoding operation that are not defined in the table used.
  - `default`: The value defined in `default` represents the code attributed to all characters for which the table assigns the value ,`U`.
  - `error`: The value defined in `error` corresponds to the code attributed to all characters for which the table assigns the value ,`R`.
  - The code table contains 32 lines of eight codes. The beginning of each line gives the offset of the first code in the line.

```

<offset> : <line of eight codes>
 00 : xx xx xx xx xx xx xx xx # 1st line
of codes
 08 : xx xx xx xx xx xx xx xx
.
.
.
 F0 : xx xx xx xx xx xx xx xx
 F8 : xx xx xx xx xx xx xx xx # 32nd
line

```

In addition to the input codes, the tables can contain:

- ,`T` (Transparent): The character code is not modified between the input and output character sets.
- ,`U` (Undefined): The character code has no equivalent in the input character set. The code attributed corresponds to the value defined in `default`.
- ,`N` (Normal): The character code corresponds to the value of the character in the input character set. The transcoding is performed in any case, with or without this attribute.
- ,`R` (Error): The character code attributed corresponds to the value defined in `error`.

### Example of table using these codes

```

table EBCDIC.ASCII1
conversion table : EBCDIC --> ASCII

```

```

default = 3F,N # Gaps
are replaced by the character ? (3F in ASCII)
data
00 : ,T ,T ,T ,T ,U 09,N ,U 7F,N
08 : ,U ,U ,U ,T ,T ,T ,T ,T
10 : ,T ,T ,T ,U ,U ,U 08,N ,U
18 : ,T ,T ,U ,U ,U 1D,N ,U ,T
20 : ,U ,U 1C,N ,U ,U 0A,N 17,N 1B,N
28 : ,U ,U ,U ,U 60,N 05,N 06,N 07,N # `
...
end

```

## Conversion table examples

The following examples provide a description of the conversion table from ASCII to EBCDIC and from EBCDIC to ASCII in the `table.trn` file (the basic ASCII code set).

You can customize the description to specify codes for undefined characters.

### ASCII to EBCDIC conversion table

```

table ASCII.EBCDIC
conversion table : ASCII --> EBCDIC

Attention : 0x13(ASCII DC3) -> 0x15 (EBCDIC NL)
data
00 : 00 01 02 03 37 2D 2E 2F
08 : 16 05 25 0B 0C 0D 0E 0F
10 : 10 11 12 15 3C 3D 32 26
18 : 18 19 3F 27 22 1D 35 1F
20 : 40 5A 7F 7B 5B 6C 50 7D # ! " # $ % & '
28 : 4D 5D 5C 4E 6B 60 4B 61 # () * + , - . /
30 : F0 F1 F2 F3 F4 F5 F6 F7 # 0 1 2 3 4 5 6 7
38 : F8 F9 7A 5E 4C 7E 6E 6F # 8 9 : ; < = > ?
40 : 7C C1 C2 C3 C4 C5 C6 C7 # @ A B C D E F G
48 : C8 C9 D1 D2 D3 D4 D5 D6 # H I J K L M N O
50 : D7 D8 D9 E2 E3 E4 E5 E6 # P Q R S T U V W
58 : E7 E8 E9 AD E0 BD 5F 6D # X Y Z [\] _
60 : 2C 81 82 83 84 85 86 87 # ` a b c d e f g
68 : 88 89 91 92 93 94 95 96 # h i j k l m n o
70 : 97 98 99 A2 A3 A4 A5 A6 # p q r s t u v w
78 : A7 A8 A9 C0 6A D0 A1 07 # x y z { | } ~
80 : 6F 6F 6F 6F 6F 6F 6F 6F
88 : 6F 6F 6F 6F 6F 6F 6F 6F
90 : 6F 6F 6F 6F 6F 6F 6F 6F
98 : 6F 6F 6F 6F 6F 6F 6F 6F
A0 : 6F 6F 6F 6F 6F 6F 6F 6F

```

```

A8 : 6F 6F 6F 6F 6F 6F 6F 6F
B0 : 6F 6F 6F 6F 6F 6F 6F 6F
B8 : 6F 6F 6F 6F 6F 6F 6F 6F
C0 : 6F 6F 6F 6F 6F 6F 6F 6F
C8 : 6F 6F 6F 6F 6F 6F 6F 6F
D0 : 6F 6F 6F 6F 6F 6F 6F 6F
D8 : 6F 6F 6F 6F 6F 6F 6F 6F
E0 : 6F 6F 6F 6F 6F 6F 6F 6F
E8 : 6F 6F 6F 6F 6F 6F 6F 6F
F0 : 6F 6F 6F 6F 6F 6F 6F 6F
F8 : 6F 6F 6F 6F 6F 6F 6F 6F
end

```

In the ASCII--->EBCDIC table above:

- ASCII hexadecimal character 20 is converted into hexadecimal 40 in EBCDIC
- ASCII hexadecimal character 21 is converted into hexadecimal 5A in EBCDIC

### *EBCDIC to ASCII conversion table*

```

table EBCDIC.ASCII
conversion table : EBCDIC --> ASCII

data
00 : 00 01 02 03 00 09 00 7F
08 : 00 00 0A 0B 0C 0D 0E 0F
10 : 10 11 12 00 00 00 08 00
18 : 18 19 00 00 00 1D 00 1F
20 : 00 00 1C 00 00 0A 17 1B
28 : 00 00 00 00 60 05 06 07 # `
30 : 00 00 16 00 00 1E 00 04
38 : 00 00 00 00 14 15 00 1A
40 : 20 00 00 00 00 00 00 00
48 : 00 00 00 2E 3C 28 2B 00 # . < (+
50 : 26 00 00 00 00 00 00 00 # &
58 : 00 00 21 24 2A 29 3B 5E # ! $ *) ;
60 : 2D 2F 00 00 00 00 00 00 # - /
68 : 00 00 7C 2C 25 5F 3E 3F # | , N % _ > ?
70 : 00 00 00 00 00 00 00 00
78 : 00 00 3A 23 40 27 3D 22 # : # @ ' = "
80 : 00 61 62 63 64 65 66 67 # a b c d e f g
88 : 68 69 00 00 00 00 00 00 # h i
90 : 00 6A 6B 6C 6D 6E 6F 70 # j k l m n o p
98 : 71 72 00 00 00 00 00 00 # q r
A0 : 00 7E 73 74 75 76 77 78 # ~ s t u v w x
A8 : 79 7A 00 00 00 5B 00 00 # y z [
B0 : 00 00 00 00 00 00 00 00

```

```

B8 : 00 00 00 00 00 5D 00 00 #]
C0 : 7B 41 42 43 44 45 46 47 # { A B C D E F G
C8 : 48 49 00 00 00 00 00 00 # H I
D0 : 7D 4A 4B 4C 4D 4E 4F 50 # } J K L M N O P
D8 : 51 52 00 00 00 00 00 00 # Q R
E0 : 5C 00 53 54 55 56 57 58 # \ S T U V W X
E8 : 59 5A 00 00 00 00 00 00 # Y Z
F0 : 30 31 32 33 34 35 36 37 # 0 1 2 3 4 5 6 7
F8 : 38 39 00 00 00 00 00 00 # 8 9
end

```

## How Gateway applies code conversion tables

When you start a Transfer, Gateway detects whether the code of the local file attributes is the same as the code of the transfer attributes in the Application or in the Transfer Request. If the codes are different, Gateway initiates conversion based on the table specified in the Application.

### Example

The Application includes the following parameters:

#### Local file attributes

- **Data encoding: ASCII**

#### Transfer attributes

- **Data encoding: EBCDIC**

When you send a file using this Application, Gateway converts it from ASCII to EBCDIC using the `ascii.ebcdic` table.

When a file is received using this Application, Gateway converts it from EBCDIC to ASCII using the `ebcdic.ascii` table.

## Compiling conversion tables: TRNCC

---

|               |                              |
|---------------|------------------------------|
| <b>Syntax</b> | <code>trncc table.trn</code> |
|---------------|------------------------------|

---

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b>  | <p>Use the <code>trncc</code> utility to compile a text file table, in order to obtain binary tables that the code conversion module can use.</p> <p><code>trncc</code> creates one compiled table per text conversion table. Each compiled table is named:</p> <p><b>TRN&lt;code_in&gt;&lt;code_out&gt;</b></p> <p>Where:</p> <p>&lt;code_in&gt; and &lt;code_out&gt; are hexadecimal values of the initial and resulting data codes (for example, TRN0102).</p> <p><code>table.trn</code> can contain several conversion tables. The compiled tables are created in the sub-directory defined by an environment parameter (<code>\$p_database</code> for UNIX-like platforms).</p> |
| <b>Return value</b> | <ul style="list-style-type: none"> <li>• 0 = Compilation completed successfully</li> <li>• Any other return value = Error</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Diagnostic</b>   | <ul style="list-style-type: none"> <li>• <i>file open fail</i>: File opening failed</li> <li>• <i>compiler fail</i>: File compilation failed</li> <li>• <i>table write fail</i>: Cannot save the compiled table</li> <li>• <i>bad name</i>: Incorrect name</li> <li>• <i>table not found</i>: Table not found</li> <li>• <i>read input error</i>: Buffer read error</li> <li>• <i>write output error</i>: Buffer write error</li> </ul>                                                                                                                                                                                                                                              |

## Converting files: TRNXCODE

|                     |                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>       | <code>trnxcode &lt;codein.codeout&gt; &lt;infile&gt;<br/>&lt;outfile&gt;</code>                                                                                                                                                                                                                                                                                      |
| <b>Description</b>  | This utility is used to convert a file before or after a transfer.                                                                                                                                                                                                                                                                                                   |
| <b>Parameters</b>   | <p>&lt;codein.codeout&gt; indicates the conversion table to use (see example below). This conversion table was compiled using the <code>trncc</code> utility.</p> <p>&lt;infile&gt; indicates the file to be converted</p> <p>&lt;outfile&gt; is the file resulting from the conversion. The output file is created with the converted content of the input file</p> |
| <b>Return value</b> | <ul style="list-style-type: none"> <li>• 0 = Conversion completed successfully</li> <li>• Any other return value = Error</li> </ul>                                                                                                                                                                                                                                  |



**Diagnostic**

- *file open fail*: File opening failed
- *compiler fail*: File compilation failed
- *usage trntrns....*: Cannot save the compiled table
- *bad name*: Incorrect name
- *table not found*: Table not found
- *read input error*: Buffer read error
- *write output error*: Buffer write error

**Example**

The following command converts an ASCII file (named inascii) to an EBCDIC file (named outebcdic):

```
trnxcode ascii.ebcdic inascii outebcdic
```

## Formatting and padding

### About formatting

The purpose of **formatting** is to change the format of a file, that is, to change the structure of the records in the following ways:

- Convert fixed length records to variable length records or vice versa
- Change the record size
- Convert stream files to fixed record format files

### About padding and truncating

**Padding** involves adding characters at the end of each record in a file to make the record a specified fixed size.

The padding function enables you to convert fixed or variable size records into larger fixed size records.

**Truncating** is the inverse of padding. Truncating involves removing characters from the end of each record to convert a padded fixed file into a non-padded variable file. To remove characters, set the final record size to a size equal to or smaller than the initial size. The records are first truncated and then the padding characters at the end of the record are deleted.

In the transfer file attributes you can:

- Enter a hexadecimal value to specify the padding character
- Specify whether truncating is allowed

## Controlling the file format

### *Via command line interface*

To control file format, set the following parameters in the [peladm create appli](#) or [peltrans](#) commands:

- When sending a file, use:
  - `-rec_fmt` and `-rec_len` parameters to control the initial format (variable, fixed or stream) of the file stored on the system
  - `-x_rec_fmt`, `-x_rec_len` and `-x_padding` parameters to control the format (variable, fixed or stream) of the file being transferred
  - `-truncating_allowed` to allow truncating of characters at the end of records
    - When receiving a file, use:
      - `-rec_fmt` and `-rec_len` parameters to control the final format (variable, fixed or stream) of the file you want to store
      - `-x_rec_fmt`, `-x_rec_len` and `-x_padding` parameters to control the format (variable, fixed or stream) of the received file
      - `-truncating_allowed` to allow truncating of characters at the end of records

### *Via the GUI*

Control file format via either of the following objects:

- Transfer Request: **Attributes** tab
- Application: **Physical file attributes** tab and **Transfer attributes** tab

Complete the fields on these tabs as follows:

- In the **Local file attributes** section, use the **Record format** and **Record length** fields to control the initial or final format (Variable, Fixed or Stream) of the file
- In the **Transfer attributes** section, use the **Record format** and **Record length** fields to control the format (Variable, Fixed or Stream) of the transferred or received file
- In the **Format** section, use the **Padding character** and **Truncating allowed** fields to specify the padding character to use and whether to allow truncating of characters at the end of records

## Changing the record format to fixed length

When changing the record format to fixed length:

- If the output record length is greater than the input record length, Gateway adds padding characters to the input record until it equals the length of the output record
- If the output record length is smaller than the input record length and:

- The option `truncating_allowed` is set to *Yes*, the input record is truncated to the length of the output record
- The option `truncating_allowed` is set to *No*, Gateway returns an error

## *Changing the record format to variable length*

When changing the record format to variable length:

- Gateway removes all trailing characters from the input record
- If the resulting record length is greater than the output record length and:
  - The option `truncating_allowed` is set to *Yes*, Gateway truncates the record
  - The option `truncating_allowed` is set to *No*, Gateway returns an error

## *Text format files*

Text file records end with a line-feed (LF) character (and possibly an additional carriage return character). Since the line-feed character differs between platforms, the following processes take place:

- **File sending:** the records are sent without the line-feed character. On the receiving platform, the file is then reconverted to text format by adding the line-feed character at the end of each record
- **File reception:** if the local file is defined as a text file, the line-feed character is added at the end of each record

Set the `text file` attribute in either the Transfer Request or the Application.

## *Changing file format from stream to fixed*

When you send a stream format file, you can change the file format from *Stream* to *Fixed*. Define the following attributes in either the Application or the Transfer Request:

### **Local attributes**

- Record format = Stream
- Organization = Sequential

### **Transfer attributes**

- Record format = Fixed
- Record length = required length between 1 and 32 000 bytes
- Padding character = required padding character
- Organization = Sequential

## Examples

In these examples, the exact names of the parameters depend on whether you use the GUI or online commands.

### *Send request*

Enter the following values:

- Local file attributes
  - Record format: Variable
  - Record length: 80
  - Data code: ASCII
  - TEXT file: yes
  - Padding character: 32
- Transfer attributes
  - Record format: Fixed
  - Record length: 90
  - Data code: EBCDIC

### *Results*

The format of the file stored on the local disk is variable with a maximum record length of 80 characters. It is a text file in ASCII code.

When Gateway sends the file, it converts it to EBCDIC using the `ascii.ebcdic` table. It deletes the line-feed characters and pads each record with hexadecimal character 32 up to a fixed length of 90 characters.

### *Receive request*

Enter the following values:

- Local file attributes
  - Record format: Variable
  - Record length: 90
  - Data code: Ascii1
  - TEXT file: yes
  - Padding character: 32
- Transfer attributes

- Record format: Fixed
- Record length: 90
- Data code: EBCDIC

### *Results*

The file sent by the Remote Site is fixed, has a record length of 90 characters and is in EBCDIC code.

On reception, Gateway converts the file into ASCII via the `ebcdic.ascii` table. It removes the hexadecimal 32 padding characters at the end of each record to reconvert the file to a variable file.

In addition, since it is a text file, Gateway adds the line-feed character at the end of each record.

# Managing Audit files

[Viewing Archived Audits](#)

[Archiving the current Audit](#)

[Filtering the Audit message display](#)

[Refreshing the Audit display](#)

[Exporting the Audit display](#)

Gateway stores the actions performed by users relating to the Gateway configuration in a dedicated audit file, to provide traceability of changes done to the product's configuration. This concerns changes both to global and to object configuration.

This functionality cannot be disabled, and the audit file can only be written by the audit process inside Gateway.

The content can be exported in a readable text format using the command line or UI. When Gateway is stopped, the audit can still be consulted using a dedicated command line tool.

You can archive the current audit file. Gateway then creates an archive file from the current `audit.dat` file by adding the archiving timestamp. To automatize the process of audit file archiving, you can create a scheduler to call a script calling the archive command.

## How to read the audit file

The audit file is stored in binary mode, unlike the `log.dat` file, stored in text mode. The audit file is located at `%GTW_installation_dir%\Gateway\run_time\audit\audit.dat`.

There are two ways of getting information from the audit file:

- directly from the GUI, at any time (in the same way as the log file)
- by exporting the audit file to a text file using `pelbase export_audit` or `pelauditexport` command line commands

## Content of the audit file after export

Here is what you will find in the audit after it is exported:

- The user login from remote command line commands and GUI. Note that successful user login from local command line commands is not logged.

- User action on objects. If the object has been modified, both the old and new value of the updated parameters are present in the audit. See below the general syntax for an entry:  
%tag to linked entries% %timestamp% username:%user% resource\_type:%object type%  
resource name:%name of the object% action\_type:%action name% action\_result:%result%  
[Changed:%parameter% from %old\_value% to %new\_value%]  
where:
  - Resource type: Site, Trading Partner, Configuration, etc
  - Action type: Update, Create, Delete, Start, Stop, Resume, etc
  - Action result: Attempt, Success, Failed, Denied. For each action initiated by a user you'll see at least 2 entries in the audit file: the 'Attempt' and the actual result of the action.
  - For filling %parameter%, %old\_value%, %new\_value% - the terminology from the command line commands have been used.

## Actions not logged to the audit

- Gateway start, status and stop actions and commands which are using these actions (ex: pelmon), because they are not linked with a Gateway's user.
- Users 'view' actions on objects, because all the other actions imply also a 'view' and the audit would be polluted with lots of 'view' actions. This means that the commands which perform view/export operations are not present into the audit file.
- 'peluconf standalone' actions, because they are done when Gateway is stopped and no user is logged
- pelmig, pelimport, pelexport, because Gateway is stopped when the commands are used
- local command lines commands: user login/logout
- pelencpass - does not require user authentication/authorization
- peluconf get (similar with the view/export on the objects)
- XMS connector in, xms connector out (deprecated)

Other notes:

- when creating a new transfer, on the 'Attempt' action, the transfer identifier is unknown, that is why the resource\_name will be "0".
- pelpur: only the purge action and the number of transfers deleted are added into the audit; the effective deletion of the transfers is not audited because it is performed by an internal process with admin rights.

## Actions logged into audit without user action

Some actions performed by the Gateway core process (`supervisor`) may appear in the audit file, because they are triggered by user actions: for example the creation of a transfer implies an update done internally, which is also present in the audit file.

## Actions for Audit files

Actions for audit files are listed in [Working with Audits \(command line\)](#) on page 2398.

## Working with Audits (command line)

- [pelbase export\\_audit](#) on page 2398
- [pelctl archive\\_audit](#): Archive the audit on page 2399
- [pelauditexport](#): export audit file when Gateway is stopped on page 2400

### pelbase export\_audit

---

|               |                                                                                                                                                                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | <b>pelbase export_audit</b> [-audit_file(-af)] [-from_date(-fd)] [-to_date(-td)] [-username(-user)] [-resource_type(-rt)] [-resource_name(-rn)] [-action_type(-at)] [-action_result(-ar)] [-output(-f) output_filename] [-append(-ap)] |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                    |                                                                               |
|--------------------|-------------------------------------------------------------------------------|
| <b>Description</b> | Exports the audit file; see <a href="#">Managing Audit files</a> on page 2396 |
|--------------------|-------------------------------------------------------------------------------|

---



**Parameters***all optional*`-audit_file(-af) full_path_to_audit_filename``-from_date(-fd) start date of exported data``-to_date(-td) end date of exported data``-resource_type(-rt)`

The resource can be one of the following: Application | CGate | CGateGroup | SGate | SGateGroup | DecisionRule | RuleTable | DiffusionList | Log | Model | Profile | User | Proxy | Site | LocalSite | Transfer | TradingPart | VFD | Config | Certificate | Key | TlsProfile | NetworkProfile | SshProfile | Statistics | Trace | PurgeModel | PPropertyList | ConnectedUser | Audit | Connection

`-resource_name(-rn)` specify the name of a resource to export audits for

`-action_type(-at)`

Can be one of the following: Read | Create | Delete | Update | Admin | Login | Logout | Start | Stop | Cancel | Suspend | Resume | Block | Unblock | Archive | Append | Rotate | Truncate | Load | Rebuild | Mount | Unmount | Purge | Disconnect | Reset | AcquireSwQueue | ReleaseSwQueue | CreateSwOutputChannel | DeleteSwOutputChannel | OpenSwOutputChannel | CloseSwOutputChannel | CreateSwInputChannel | DeleteSwInputChannel | OpenSwInputChannel | CloseSwInputChannel | Invalidate

`-action_result(-ar)`

Can be one of the following: A(ttempt) | S(uccess) | F(ailed) | D(enied)

`-output(-f) specify the filename of the audit output``-append(-ap)`**Example**

```
pelbase export_audit -f my_audit_file.txt
```

## pelctl archive\_audit: Archive the audit

**Syntax**

```
pelctl archive_audit
```

|                    |                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b> | <p>This command archives the audit file (similarly to <code>archive_log</code> command) in order to limit the size of the audit file. To automatize the process of audit file archiving, you can create a scheduler to call a script in which the archive command it run.</p> <p><b>Recommendation:</b> Archive the audit file regularly to avoid the file becoming too large.</p> |
| <b>Parameters</b>  | None                                                                                                                                                                                                                                                                                                                                                                               |

## pelauditexport: export audit file when Gateway is stopped

|                    |                                                                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>      | <pre>pelauditexport [-audit_file(-af)][-from_date(-fd)][-to_date(-td)][-username(-user)][-resource_type(-rt) ][-resource_name(-rn)][-action_type(-at) ][-action_result(-ar) ][-output(-f) output_filename][-append(-ap)]</pre> |
| <b>Description</b> | <p>Exports the audit file. It is similar to the <code>pelbase export_audit</code> command, except it can also be used when Gateway is stopped.</p>                                                                             |

**Parameters***all optional*

---

`-audit_file (-af)` audit file name (specify the full path)

---

`-from_date (-fd)` specify the start date of exported data

---

`-to_date (-td)` end date of exported data

---

`username (-user)` export only audits for a given user

---

`-resource_type (-rt)`

The resource can be one of the following: Application | CGate | CGateGroup | SGate | SGateGroup | DecisionRule | RuleTable | DiffusionList | Log | Model | Profile | User | Proxy | Site | LocalSite | Transfer | TradingPart | VFD | Config | Certificate | Key | TlsProfile | NetworkProfile | SshProfile | Statistics | Trace | PurgeModel | PPropertyList | ConnectedUser | Audit | Connection

---

`-resource_name (-rn)` specify the name of a resource to export audits for

---

`-action_type (-at)`

Can be one of the following: Read | Create | Delete | Update | Admin | Login | Logout | Start | Stop | Cancel | Suspend | Resume | Block | Unblock | Archive | Append | Rotate | Truncate | Load | Rebuild | Mount | Unmount | Purge | Disconnect | Reset | AcquireSwQueue | ReleaseSwQueue | CreateSwOutputChannel | DeleteSwOutputChannel | OpenSwOutputChannel | CloseSwOutputChannel | CreateSwInputChannel | DeleteSwInputChannel | OpenSwInputChannel | CloseSwInputChannel | Invalidate

---

`-action_result (-ar)`

Can be one of the following: A(ttempt) | S(uccess) | F(ailed) | D(enied)

---

`-output (-f)` specify the filename of the audit output

---

`-append (-ap)` append audit at the end of an existing audit file

---

## Viewing and managing Audit

### Viewing Audits

To view a complete list of all audit messages contained in the current audit file:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
2. Click the **Audit** command

Gateway displays all audit messages in a pair of tables in the display pane (to the right). Each row of the main table represents an audit message. Selecting a row from the main display loads the second display to show what changes have been made.

The main table columns contain the following information for each message.

| Column heading       | Contents                                                                                                | Data format          |
|----------------------|---------------------------------------------------------------------------------------------------------|----------------------|
| <b>Action Result</b> | The result of the audited action                                                                        | Icon and text        |
| <b>Date</b>          | Date and time when the message was audited                                                              | dd:MM:yy<br>HH:mm:SS |
| <b>User</b>          | The user who altered the configuration                                                                  | Text                 |
| <b>Resource Type</b> | The resource type that was altered                                                                      | Text                 |
| <b>Resource Name</b> | The name of the resource that was altered                                                               | Text                 |
| <b>Action Type</b>   | The action type that was performed on the altered resource                                              | Text                 |
| <b>Changes</b>       | Number of changes done with this action. You can see all the changes in the table from the bottom part. | Text                 |

### *Customizing the main display*

By default, Gateway displays the audit messages with each of the above columns visible and with the messages displayed from top to bottom in the reverse order in which they were generated.

**To change the order of the display**, click one of the column headers cells to sort according to that column. Click again to sort in reverse order.

The second table columns contain the following information for each message.

| Column heading        | Contents                           | Data format |
|-----------------------|------------------------------------|-------------|
| <b>Id</b>             | Id of the change                   | Text        |
| <b>Parameter Name</b> | The parameter that was altered     | Text        |
| <b>Old Value</b>      | Old value of the altered parameter | Text        |
| <b>New Value</b>      | New value of the altered parameter | Text        |

## Customizing the second display

Click one of the column headers cells to sort according to that column. Click again to sort in reverse order.

**To change the size of displayed columns**, right-click any column.

Gateway displays the following context menu.

Select the command:

- **Autosize the column**, to adjust the selected column size to the column contents
- **Hide the column**, to remove the entire column from view
- **Autosize all the columns**, to adjust all columns to the size of their contents

## Viewing Archived Audits

To view a complete list of all archived audit files:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
2. Click the **Archived Audit**

Gateway displays all archived audit files in a table in the display pane (to the right). Each row of the table represents an audit file. The table columns contain the following information for each message:

| Column heading | Contents                                 | Data format   |
|----------------|------------------------------------------|---------------|
| File Name      | Archived Audit file name                 | Icon and text |
| Date           | Date when was the last writing into file | dd.MM.yy      |
| Time           | Date when was the last writing into file | HH:mm:ss      |

3. Double-click the archived file name to view it. A sub-tree item is created, containing the file name for each archived log the user loads.

**Note** Pay attention to how many archived audit you are loading.

## *Archiving the current Audit*

To archive the current Audit file:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
2. Right-click the **Audit**  
Gateway displays a context menu.
3. From the context menu select **Archive**

## **Filtering the Audit message display**

To select filter criteria for the display of current audit messages:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
2. Right-click the **Audit** command  
Gateway displays a context menu.
3. From the Audit context menu select **Filtering**

Gateway displays the *Select Audit Messages* dialog box. Use the fields of the *Select Audit Messages* dialog box to create selection criteria for the display of Audit messages. You can use one or more of the fields described in the following table.

---

| Field                 | Use                                                           |
|-----------------------|---------------------------------------------------------------|
| <b>Maximum number</b> | Enter the maximum number of message results you want to view. |

---

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Date and start time</b> | <p>You can select either starting date or starting time criteria, or combine the two. You can combine starting time and/or starting date criteria with ending time and/or ending date criteria.</p> <p>To <b>set a starting date</b> as selection criteria:</p> <ol style="list-style-type: none"> <li>1. Click the down arrow on the starting date field displaying the current date. Gateway displays the calendar selection tool.</li> <li>2. Use the calendar selection tool to navigate to the start month and start date of your choice.</li> </ol> <p>To <b>set a starting time</b> as selection criteria:</p> <ol style="list-style-type: none"> <li>1. Click the check box next to the starting time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated.</li> <li>2. Click to select the <b>time unit</b> you want to modify: hour, minute or second. Gateway highlights the time unit.</li> <li>3. Type a valid number or use the up and down arrows to set the time unit.</li> <li>4. Select and set the other time units (hour, minute, second) as required.</li> </ol> |
| <b>Date and end time</b>   | <p>You can set either ending date or ending time criteria, or combine the two. You can combine ending time and/or ending date criteria with starting time and/or starting date criteria.</p> <p>To <b>set an ending date</b> as selection criteria:</p> <ol style="list-style-type: none"> <li>1. Click the down arrow on the ending date field displaying the current date. Gateway displays the calendar selection tool.</li> <li>2. Use the calendar selection tool to navigate to the end month and end date of your choice.</li> </ol> <p>To <b>set an ending time</b> as selection criteria:</p> <ol style="list-style-type: none"> <li>1. Click the check box next to the ending time selection field. Gateway converts the time display from grayed out to normal text to indicate that it is activated.</li> <li>2. Click to select the <b>time unit</b> you want to modify: hour, minute or second. Gateway highlights the time unit.</li> <li>3. Type a valid number or use the up and down arrows to set the time unit.</li> <li>4. Select and set the other time units (hour, minute, second) as required.</li> </ol>                  |
| <b>Username</b>            | Enter the name of the user you want to display                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Resource Type</b>       | <p>To select a resource type as a display selection criteria:</p> <ol style="list-style-type: none"> <li>1. Click the down arrow located in the <b>Resource type</b> field. Gateway displays a drop-down list: Application, Audit, Certificate, CGate, CGateGroup, Config, ConnectedUser, DecisionRule, DiffusionList, Key, LocalSite, Log, Model, NetworkProfile, Profile, PropertyList, Proxy, PurgeModel, RuleTable, SGate, SGateGroup, Site, SshProfile, Statistics, TlsProfile, Trace, TradinPartner, Transfer, User, VFD, and others.</li> </ol> <p>Select the resource type you want to use as display selection criteria.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Resource Name</b> | Enter the name of the resource you want to filter                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Action Type</b>   | <p>To select an action type as display selection criteria:</p> <p>2. Click the down arrow located in the <b>Action type</b> field.</p> <p>Gateway displays a drop-down list:</p> <p>Admin, Append, Archive, Block, Cancel, Create, Delete, Load, Login, Logout, Mount, Purge, Read, Rebuild, Resume, Rotate, Start, Stop, Suspend, Truncate, Unblock, Unmount, Update, and others.</p> <p>Select the action type you want to use as display selection criteria.</p> |
| <b>Action Result</b> | <p>To select an action result as a display selection criteria:</p> <p>3. Click the down arrow located in the <b>Action result</b> field.</p> <p>Gateway displays a drop-down list:</p> <ul style="list-style-type: none"> <li>· Attempt</li> <li>· Denied</li> <li>· Failure</li> <li>· Success</li> </ul> <p>Select the action result you want to use as display selection criteria.</p>                                                                           |

#### To create a new filter:

1. Click **Add New Filter**, to create a blank template for the new filter.
2. Enter a name for the Filter.
3. Choose whether the filter is available only for this server, or whether it is global. (This option cannot be changed later).
4. Configure Selection criteria.
5. Add a text comment for this filter. (optional)

#### To delete a filter:

1. Select the filter.
2. Click **Delete Filter**.

#### To modify a filter:

1. Select the filter.
2. Configure Selection criteria.
3. Click **Save and apply**.

## Refreshing the Audit display

To refresh the display of Audit messages, taking into account the most recent network events:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
2. Right-click the **Audit** command



Gateway displays a context menu.

3. From the Audit context menu select **Refresh**

Gateway regenerates the display, adding any new audit messages.

## Exporting the Audit display

To export Audit messages:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
2. Right-click the **Audit** command  
Gateway displays a context menu.
3. From the Audit context menu select **Export...**  
Gateway then exports the view to the selected file.

## Exporting the Archived Audit display

To export Archived Audit messages:

1. In the left pane of the GUI main window, click **+** to expand the nodes:  
**Transfer Management > Monitoring**
1. Click on the **Archived Audit** node to expand the nodes  
**Transfer Management > Monitoring**
2. Right-click the desired Archived audit node  
Gateway displays a context menu.
3. From the Audit context menu select **Export...**  
Gateway then exports the view to the selected file.

## Archiving audit files automatically

### *About automatic audit archiving*

With automatic audit archiving, Gateway can run without interruption while archiving audit data. You can set automatic audit archiving to be triggered either:

- at regular intervals in time,
- or once a set number of lines of audit is reached.

When it's time to archive an audit file, Gateway renames the file currently in use, appending a timestamp; then it creates and opens a new empty audit file.

## *How to set up automatic audit file archiving*

To set up automatic archiving of audit files you can use the following parameters from the audit section:

- `arc_active` values: *0* (inactive) or *1* (active)
- `auto_arc_lines` the number of lines at which you want to trigger audit file archiving.
- `auto_arc_period` the period, in the format: `MM:DD hh:mm:ss`
- `start_arc_period` start date and time, in the format: `MM:DD hh:mm:ss`

You can alter these parameters using the `peluconf` command. Gateway takes the changes into account after the next restart.

**Example:** to set an automatic archive for the audit file every morning at 9:00 AM, enter the following command:

```
peluconf set -s audit arc_active 1 auto_arc_period "00:01 00:00:00"
start_arc_period "00:00 09:00:00"
```

## *Disabling Sentinel audit replication*

You can disable Sentinel replication when SWIFT protocol is not in use. To do that you can set the `sentinel_replication` parameter from audit section to 0. This parameter is ignored when SWIFT protocol is in use: in this case, Sentinel replication remains on.

Command: `peluconf set -s audit sentinel_replication 0`

**Note** You must restart Gateway before this takes effect, as this is a static parameter.