



# SecureTransport

Version 5.4  
2 April 2024

## Web Client Configuration Guide



Copyright © 2019 Axway

All rights reserved.

This documentation describes the following Axway software:

Axway SecureTransport 5.4

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.

---

# Revision history

The following changes are added to the SecureTransport 5.4 ST Web Client Configuration Guide:

SecureTransport version	Document revision number	Topics added/updated
5.4	5.4.01	The following topics are added to the customization options: <ul style="list-style-type: none"><li>• <a href="#">Lite mode on page 28</a></li><li>• <a href="#">Session Management on page 37</a></li><li>• <a href="#">Mailbox settings on page 24</a></li><li>• <a href="#">Custom links on page 38</a></li><li>• <a href="#">Pre-login disclaimer banner on page 42</a></li></ul>
5.4	5.4.02	The following changes are made to the <a href="#">Branding on page 17</a> topic: <ul style="list-style-type: none"><li>• Instructions on how to change an existing style or create a new one are added to the "Change the color theme" subtopic which is now renamed to "Change the color theme and style"</li><li>• <a href="#">Set up a custom layout on page 21</a> subtopic added</li></ul>
5.4	5.4.03	In the <a href="#">Branding on page 17</a> section, the <i>postcss-cli</i> version is updated
5.4	5.4.04	<ul style="list-style-type: none"><li>• <a href="#">Manage folder sharing on page 32</a> added</li><li>• <a href="#">Branding on page 17</a> updated</li></ul>
5.4	5.4.05	<ul style="list-style-type: none"><li>• <a href="#">Configure upload settings on page 25</a> updated</li><li>• <a href="#">UI performance settings on page 27</a> added</li></ul>
5.4	5.4.06 – current version	<ul style="list-style-type: none"><li>• "Account ID" topic extended with instructions for managing the visibility and the type of user information displayed in the Welcome menu, and renamed to <a href="#">Control the displayed user information on page 29</a></li><li>• <a href="#">Access Control on page 31</a> topic updated with instructions for disabling users' ability to delete files and folders</li></ul>

# Contents

<b>Preface</b> .....	<b>6</b>
Who should read this document .....	6
Related documentation .....	6
Get more help .....	7
Training .....	8
<b>1 Customization overview</b> .....	<b>9</b>
<b>2 Configuration files</b> .....	<b>10</b>
<b>3 Enable and disable features</b> .....	<b>12</b>
<b>4 Add and modify languages</b> .....	<b>14</b>
Add a supported language .....	14
Modify an existing language .....	15
<b>5 Modify date and time format</b> .....	<b>16</b>
<b>6 Branding</b> .....	<b>17</b>
Prerequisites .....	17
Procedure .....	17
Change the logos, header, and disclaimer .....	18
Change the favicon .....	18
Change the color theme and/or the style .....	18
Set up a custom layout .....	21
<b>7 Modify the view configuration</b> .....	<b>23</b>
Change the default view .....	23
Enable or disable a view .....	23

---

<b>8 Mailbox settings</b> .....	<b>24</b>
<b>9 Configure upload settings</b> .....	<b>25</b>
<b>9 UI performance settings</b> .....	<b>27</b>
<b>10 Lite mode</b> .....	<b>28</b>
<b>11 Control the displayed user information</b> .....	<b>29</b>
Hide user information .....	29
Change the displayed user information .....	29
<b>12 Access Control</b> .....	<b>31</b>
<b>13 Manage folder sharing</b> .....	<b>32</b>
Disable Folder sharing from ST Web Client .....	32
Set the default folder sharing options .....	32
<b>14 Password Management</b> .....	<b>35</b>
<b>15 Session Management</b> .....	<b>37</b>
<b>16 Custom links</b> .....	<b>38</b>
Enable Custom Links feature .....	38
Custom links without i18n translations .....	38
Custom links with i18n .....	39
<b>17 Pre-login disclaimer banner</b> .....	<b>42</b>
Enable the disclaimer feature .....	42
Set up the disclaimer banner .....	43
Add translations for other languages .....	44
<b>18 Customizations after a patch or upgrade</b> .....	<b>45</b>

---

# Preface

The *ST Web Client Configuration Guide* describes how to configure and customize the ST Web Client user interface.

The possible customizations are:

- Enabling and disabling features
- Adding and modifying languages
- Branding the user interface
- Changing the user interface view
- Changing the Transfer Queue settings
- Enabling Lite mode
- Modifying access to file and folder operations
- Modifying the "Forgot password" and "Change password" links
- Managing Session settings
- Adding custom links to the user menu
- Adding a disclaimer banner on the login page

## Who should read this document

This guide is intended for administrators who want to customize the ST Web Client user interface.

This guide assumes the reader is familiar with creating, editing, and modifying JSON files and generating CSS files.

## Related documentation

SecureTransport provides the following documentation:

- *SecureTransport Administrator's Guide* – This guide describes how to use the SecureTransport Administration Tool to configure and administer your SecureTransport Server. The content of this guide is also available in the Administration Tool online help.
- *SecureTransport REST API documentation* – The portal published API documentation derived from the API swagger documents. To access the administrator API documentation, go to [SecureTransport Administrator API v1.4](#). To access the end-user API documentation, go to [SecureTransport End-User API v1.4](#).

- *SecureTransport Appliance Guide* – This guide provides the SecureTransport Appliance installation, configuration, and operation instructions. It also provides SecureTransport installation and upgrade instructions for Axway Appliances.
- *SecureTransport Capacity Planning Guide* – This guide provides information useful when planning your production environment for SecureTransport.
- *SecureTransport Developer's Guide* – This guide provides the descriptions and usage of the plugable information for the SecureTransport Pluggable Transfer Site and how to implement a Pluggable Transfer Site. It also provides Swagger REST API integration instructions and custom Address Book source implementation instructions and custom plugins/exits source implementation instructions.
- *SecureTransport Getting Started Guide* – This guide explains the initial setup and configuration of SecureTransport using the SecureTransport Administrator setup interface.
- *SecureTransport Installation Guide* – This guide explains how to install and uninstall SecureTransport on UNIX-based platforms and Microsoft Windows.
- *SecureTransport Release Notes* – This document contains information about new features and enhancements, information received after the finalization of the rest of the documentation, and a list of known and fixed issues.
- *SecureTransport Security Guide* – This guide provides security information necessary for the secure operation of the SecureTransport product.
- *SecureTransport Software Development Kit (SDK)* – A set of software development tools and examples that allow extending SecureTransport by consuming and implementing available APIs.
- *SecureTransport Upgrade Guide* – This guide explains how to upgrade SecureTransport on UNIX-based platforms and Microsoft Windows.
- *ST Web Client Configuration Guide* – (This document) This guide describes how to configure and customize the ST Web Client user interface.
- *ST Web Client User Guide* – This guide describes how to use the ST Web Client.

Go to Axway Support at [support.axway.com](https://support.axway.com) to view or download documentation. The website requires login credentials and is for customers with active support contracts.

## Get more help

Go to Axway Support at [support.axway.com](https://support.axway.com) to get technical support, download software, documentation and knowledgebase articles. The website requires login credentials and is for customers with active support contracts.

The following support services are available:

- Official documentation
- Product downloads, service packs, and patches
- Information about supported platforms
- Knowledgebase articles
- Access to your cases

When you contact Axway Support with a problem, be prepared to provide the following information for more efficient service:

- Product version and build number
- Database type and version
- Operating system type and version
- Service packs and patches applied
- Description of the sequence of actions and events that led to the problem
- Symptoms of the problem
- Text of any error or warning messages
- Description of any attempts you have made to fix the problem and the results

## Training

Axway offers training across the globe, including on-site instructor-led classes and self-paced online learning. For details, go to [training.axway.com](https://training.axway.com)

---

# Customization overview

# 1

Most of the Web Client customizations are performed by modifying the `custom/stwebclient.config.json` configuration file. This file is not installed by default, it must be created after installation.

The available customizations are:

- Enabling/Disabling features
- Adding/Modifying languages
- Changing the branding
- Modifying the view configuration
- Changing the Transfer Queue settings
- Enabling Lite mode
- Modifying access to file and folder operations
- Modifying the "Forgot password" and "Change password" links
- Managing Session settings
- Adding custom links to the user menu
- Adding a disclaimer banner on the login page

The directory paths used in this guide are relative to the ST Web Client installation folder:

```
[SecureTransport_installDir]/share/ftdocs/html/skin/ric
```

---

# Configuration files

# 2

The settings in the custom `/custom/stwebclient.config.json` configuration file override the settings in the default `C/config/default.config.json` configuration file.

**Note** You must not modify the `C/config/default.config.json` configuration file. You must make your changes in the `/custom/stwebclient.config.json` configuration file.

The default configuration file has the following top-level keys:

```
{
  "features": {
    ... // enabling/disabling features
  },
  "i18n": {
    ... // adding/modifying languages (internationalization)
  },
  "branding": {
    ... // branding
  }
  "transferQueue": {
    ... // transfer queue settings
  },
  "views": {
    ... // views configuration
  }
  "liteMode": {
    ... // enabling/disabling liteMode
  },
  "passwordManagement": {
    ... // forgot password and change password settings
  },
  "accessControl": {
    ... // enabling/disabling access to some UI functions
  },
}
```

```
"sessionManagement": {
  ... // client session management settings
},
"disclaimer":{
  ... // disclaimer settings
}
}
```

The custom `stwebclient.config.json` configuration file must have the same structure as the `default.config.json` configuration file, but including a key in the custom configuration file is optional. If a key is not present in the custom configuration file, the key in the default configuration file is used.

For example, the following custom `stwebclient.config.json` configuration file only has a branding key.

```
{
  "branding": {
    ...
  }
}
```

In this example, the features and internationalization configurations are provided by the keys in the `default.config.json` file, and the branding configuration is provided by the key in the `stwebclient.config.json` file. Keys not included in the `stwebclient.config.json` file are inherited from the `default.config.json` file.

# Enable and disable features

# 3

The `features` key in the `default.config.json` configuration file defines all the ST Web Client features that can be enabled or disabled. To enable or disable a feature, adding a "features" section in the custom `stwebclient.config.json` configuration file, and change the value of the corresponding key.

The `default.config.json` file contains the following features:

```
{
  ...
  "features": {
    "addressBook": {
      "enabled": null,
      "collaborationAllowed": null,
      "favoritesEnabled": true
    },
    "i18n": {
      "enabled": true
    }
  },
  ...
}
```

Where:

`true` - The feature is enabled.

`false` - The feature is disabled.

`null` - The feature is controlled by the SecureTransport server. Server controlled features can be disabled, but not enabled.

Keys not included in this file are inherited from the `default.config.json` file.

To disable the internationalization (language localization) feature, edit the `stwebclient.config.json` file and add the following:

```
{
  ...
}
```

```
"features": {  
  "i18n": {  
    "enabled": false  
  }  
},  
...  
}
```

To disable the Address Book feature, even if it is enabled on the SecureTransport server, edit the `stwebclient.config.json` file and add the following:

```
{  
  ...  
  "features": {  
    "addressBook": {  
      "enabled": false  
    }  
  },  
  ...  
}
```

To make the feature changes effective, clear your browser cache and reload ST Web Client. No server restart is required.

---

# Add and modify languages

# 4

ST Web Client is installed with some predefined languages. The translations for the predefined languages are located in the `C/locales` folder.

Each translation has a language code folder which contains a JSON file with the translated labels. For example, the English translation is in the `C/locales/en/translation.json` file.

## Add a supported language

The default translations that are part of ST Web Client installation should not be altered. You cannot add new languages in the same location as the default ones. They must be added in the `custom` folder.

To add the Bulgarian language:

1. Copy the content from the English translation (`C/locales/en/translation.json`) file into the `custom/locales/bg/translation.json` file.

2. Translate the values from English to Bulgarian.

If a value is not translated, it will default to the English translation.

3. Edit the `stwebclient.config.json` file and specify the additional language in the `i18n` section:

```
{
  ...
  "i18n": {
    "additionalLocales": [{
      "id": "bg",
      "nameEng": "Bulgarian",
      "name": "Български"
    }]
  }
  ...
}
```

4. Refresh the ST Web Client and open the *Preferences* pane. The new language should be present in the *Language* drop-down menu.

## Modify an existing language

You can change the labels for any (default or additional) language.

To change the Welcome label to "Hello Mate:"

1. Create the `custom/locales/en/override.json` file.
2. Using the same structure as the English translation (`C/locales/en/translation.json`) file, add the labels that need changed:

```
{
  "header": {
    "user_welcome": "Hello mate, "
  }
}
```

3. Edit the `stwebclient.config.json` file and specify the override for the English language in the `i18n` section:

```
{
  ...
  "i18n": {
    "overridenLocales": ["en"]
  }
  ...
}
```

4. Refresh the ST Web Client. The overridden label should be displayed.

---

# Modify date and time format

# 5

You can modify the displayed date and time format. The default format is: M/D/YYYY, h:mm:ss A

1. In the `stwebclient.config.json` file, specify the preferred format in the `i18n` section:

```
{  
  ...  
  "i18n": {  
    "dateFormat": "MMMM Do YYYY, h:mm:ss a"  
  }  
  ...  
}
```

2. Refresh the ST Web Client.

For a list of date and time formatting options, refer to [Formatting options](#).

Changing the branding means modifying the following aspects of the Secure Transport Web Client:

- Color theme and style
- Logo on the login page and on the pages a logged in user can access
- Text header next to the logo
- Disclaimer on the login page
- Favicon
- Add custom content via Layout Injector plug-in

## Prerequisites

On a separate machine than the SecureTransport Server, install NodeJS from the official website <https://nodejs.org/> or through the package manager if you're working on a Linux machine.

Then install the LESS compiler (version 3.9.0), postcss-cli (version 5.0.1), autoprefixer (version 9.4.5), clean-css (version 4.2.1), and postcss-clean (version 1.1.0):

```
npm install -g less@3.9.0 postcss-cli@5.0.1 autoprefixer@9.4.5 clean-  
css@4.2.1 postcss-clean@1.1.0
```

**Note** Do not install NodeJS on the production server where SecureTransport is installed. Use a separate machine for building the branding assets.

## Procedure

The application is styled and branded by the following files and folders found in the ST Web Client installation folder:

- `C/assets/wap.css` - Controls the color theme and the style. It is generated from several `.less` files delivered in the branding assets archive.
- `custom/stwebclient.config.json` - Controls the logos, the text header displayed next to the logo and the disclaimer on the login page.

## Change the logos, header, and disclaimer

This is done by editing the custom configuration file as described in section [Configuration files on page 10](#).

Copy the “branding” section from `C/config/default.config.json` into `custom/stwebclient.config.json`, and do the modifications as desired.

The `logo` and `header` parameters specify the logo and header text displayed on the main application pages (the `list.html` template) and in the **About** dialog. All the [HTML templates](#) except `list.html` use the `logoLogin` and `headerLogin` parameters.

**Note** The logo paths are relative to `[SecureTransport_installDir]/share/ftdocs`. If you modify them, make sure to create them on the server.

## Change the favicon

This is done by adding an ICO file in `[SecureTransport_installDir]/share/ftdocs`. The ICO file must respect these conditions:

- Must be named `favicon.ico`
- Must be 64x64 pixels in size

This customization is not preserved between installations or after applying a patch or service pack.

The `favicon.ico` file must be added again after these operations.

## Change the color theme and/or the style

1. Copy the branding assets archive `C/assets/branding/branding_assets.zip` on the system where you installed NodeJS and unzip it. After unzipping, it should have the following directory structure:

```
assets
_508.less
_a5.less
_book.less
_branding.less
_download.less
_fonts.less
_header.less
_icons.less
_media-queries.less
_mixins.less
```

```

_reset.less
_spinner.less
progress.less
wap.less

```

2. To change the color theme, follow the steps bellow. If you want to change an existing style or create a new one, go to step 3.

2.1 Open `_branding.less` in a text editor. The color theme is defined by a set of variables:

- `toolbarBgImage` - [null | path]
  - background image for the toolbar component. Paths are relative to `[SecureTransport_installDir]/share/ftdocs`.
- `baseColor` - [color]
  - main color of the file explorer
- `headerColor` - [color]
  - header background color.
- `sidebarColor` - [color]
  - sidebar background color.
- `highlightColor` - [color]
  - active element color.
- `secondaryColor` - [color]
- `headerHighlightColor` - [color]
  - custom highlight color for active elements in the header.
- `headerUsesFlatColor` - [auto|true|false]
  - header background style. Set to true for flat color, false for gradient or leave auto.

2.2 You can either modify the default theme or use one of the predefined alternative themes. In order to use one of the predefined themes, comment the default theme and uncomment the desired predefined theme. A theme is commented by adding `/*` before the first variable in the group and `*/` after the last one. It is uncommented by removing the lines containing `/*` and `*/` at the beginning and the end of its variable group.

3. To override an existing style or create a new one, follow the steps:

3.1 Create a file called "custom.less".

3.2 Define all the new style definitions and overrides in this file. You can reuse definitions and variables from the other LESS files, the compiler imports them automatically.

Example: removing the background image

```

/* the original style definition in wap.less at line 3104 */
body.Login {

```

```

color: contrast(@baseColor);

background-color: @baseColor;

background-image: url(textures/login_background_
1920x1080.jpg);

background-position: center center;

background-size: cover;

background-attachment: fixed;

}

/the modified style definition in * custom.less */

body.Login {

/* Set background-image to 'none' */

background-image: none;

}

```

3.3 Copy `custom.less` next to the other branding artifacts (`wap.less`, `progress.less`, etc. ) to the unzipped folder.

4. After modifying `branding.less`, you need to generate the CSS files containing the new theme. This is done using the LESS compiler which should be installed beforehand.

4.1 Open the assets folder in a terminal and type:

```
lessc wap.less wap.css && lessc progress.less progress.css
```

If the compilation is successful, you will see a `wap.css` and `progress.css` files next to `wap.less` and `progress.less`.

4.2 In the same folder, create the postcss configuration file: `postcss.config.js`. Paste the following content inside and save.

```

module.exports = {
  plugins: {
    autoprefixer: { browsers: ['last 1 version', 'ie >= 11'] },
    'postcss-clean': {
      1: {
        removeQuotes: false,
        normalizeUrls: false,
      },
    },
  },
};

```

4.3 To clean, minify, and add browser prefixes, type:

```
postcss wap.css -r && postcss progress.css -r
```

This will overwrite the existing CSS files with their minified versions.

5. Copy both CSS files to the server in `C/assets` by overwriting the old `wap.css` and `progress.css`.

Make sure to put the image used for `@toolbarBgImage` on the server .

## Set up a custom layout

This feature enables you to add custom content on the ST Web Client login page and on the pages a logged in user can access.

**Note** Your customizations must NOT modify any ST Web Client file inside `<FDH>/share/ftdocs/html/skin/ric/C`. These kinds of customizations are not supported.

1. Collect all your changes in an HTML template. Analyze them and make sure that you can apply these changes after STWC initialization.
2. Create the plug-in by following the rules:
  - must be placed in its own folder, e.g., `/my-layout`.
  - must contain an `index.html` file (this is the entry point), e.g., `/my-layout/index.html`.
  - the plug-in folder can contain resources like images and fonts, e.g., `/my-layout/mylogo.png`.
  - the layout name can contain letters, numbers, underscores, and dashes; If the name does not comply with these rules, the layout will be ignored.
3. When the plug-in is ready, add it to ST Web Client by copying its folder to `/custom`.
4. Edit `stwebclient.config.json` to tell ST Web Client to load the plug-in:

```
{
  "branding": {
    "layoutInject": {
      "my-layout": ["login", "list"]
    }
  }
}
```

The values on the right are the target HTML template names without the extension. In this case, the plug-in is applied to the login and the main application pages.

The templates are placed in `<FDH>/share/ftdocs/html/skin/ric/C`.

The available templates are: `account_expired`, `download_public`, `download`, `list`, `lockout`, `login`, `logout`, `message`, `password_expired`, `password_fail`, `password_msg`, `password_reset`, `password`, `secretQuestion`, `timeout`.

**Note** The custom HTML is inserted at the beginning of the `<body>` tag. You can use CSS to style it and change its position.

**HINTS** When creating your plug-in, you can:

- use the CSS style definitions in `wap.css` and `progress.css`.
- add styles in the plug in HTML file (`index.html`). If the styles are overriding existing ones, flickering may occur.

---

# Modify the view configuration 7

The ST Web Client has two views:

- File view
- Mailbox view

The default view can be changed and a view can be disabled.

To modify the view configuration, copy the `views` key from the `C/config/default.config.json` file into the `custom/stwebclient.config.json` file, and edit the custom configuration file as described in [Configuration files on page 10](#).

The view configuration is located in the `views` key.

## Change the default view

To set the default view, change the `defaultView` parameter. The parameter values are:

- `allFiles` - File view
- `mailbox` - Mailbox view

## Enable or disable a view

To enable a view, set the `enable` parameter for the view to `true`. To disable a view, set the `enable` parameter for the view to `false`.

# Mailbox settings

# 8

When the ST Web Client sends an e-mail, it checks for a certain period of time whether the message has been sent successfully or it has been blocked because it contained sensitive information. In both scenarios, a message status notification is displayed to the user. If ST Web Client cannot obtain the message status before the allotted time expires, the user needs to refresh the folder to see details.

This behavior is controlled by two configuration options:

- `sendPollingInterval` (milliseconds) specifies the interval between successive checks
- `sendPollingRetries` specifies how many times ST should check for the status of the message

The default values are listed below.

```
{
  ...
  "mailbox": {
    "sendPollingInterval": 500,
    "sendPollingRetries": 10
  }
  ...
}
```

# Configure upload settings

# 9

You can manage the ST Web Client client upload settings using the `transferQueue` property.

The following code snippet shows the upload defaults:

```
{
  ...
  "transferQueue": {
    "failedLastXHours": 24,
    "transferOptions": {
      "uploadChunkSize": 104857600,
      "readChunkSize": 262144,
      "maxRetryCount": 3,
      "maxChunkRetryCount": 3,
      "timeoutBeforeRetry": 10000,
      "maxSimultaneousJob": 3
    },
    "transferHistoryLimit": 100
  },
  ...
}
```

To change a default setting, in the custom configuration file, add a "transferQueue" section and overwrite the desired value.

Listed below are all the accepted properties and what they affect.

Property (type)	Description
<b>failedLastXHours</b> (integer)	The number of hours a failed transfer is counted in the failed transfers notification. Default value: 24, which means that any transfer that failed more than 24 hours ago will not be counted in the failed transfers notification.
<b>transferOptions</b> (object)	Controls chunk upload

<b>Property</b> <i>(type)</i>	<b>Description</b>
<b>uploadChunkSize</b> <i>(integer)</i>	The size of the upload chunk, in bytes. Default value: 104857600 (100 MB). Higher values may increase the upload speed but will make the upload progress less responsive and may cause high CPU load.
<b>readChunkSize</b> <i>(integer)</i>	The size of the read chunk, in bytes. default value: 262144 (0.25 MB) Higher values may increase the upload speed but more CPU will be consumed because of the incremental MD5. We recommend a value less than 1 MB.
<b>maxRetryCount</b> <i>(integer)</i>	The number of maximum retry attempts for the transfer job. Default value: 3
<b>maxChunkRetryCount</b> <i>(integer)</i>	The maximum number of retry attempts for the current upload chunk after which the chunk will fail and the job will be restarted (see <b>maxRetryCount</b> ) Default value: 3
<b>timeoutBeforeRetry</b> <i>(integer)</i>	The period before retrying the current upload chunk in case of error, in milliseconds. Default value: 10000
<b>maxSimultaneousJob</b> <i>(integer)</i>	The number of simultaneous upload jobs. Default value: 3 If you select more than three (files for upload), the first three will start and the rest will wait. For Chrome and Firefox, the value can be up to 20. For Edge and IE, we recommend a value less than 10.
<b>transferHistoryLimit</b> <i>(integer)</i>	The number of transfers to get from the transfers API on application load (if the <b>Allow this account to submit transfers using the Transfers RESTful API</b> option is enabled for the user). Default value: 100 Higher values increase the number of history entries in the <a href="#">Uploads monitor</a> but take more time to load.

# UI performance settings

# 9

The following code snippet shows the ST Web Client UI performance defaults:

```
{
  ...
  "uiPerformance": {
    "refreshThrottleTime": 1000,
    "addressBook": {
      "minimumBatchSize": 100
    }
  }
  ...
}
```

To change the UI performance default setting, in the custom configuration file, add a "uiPerformance" section and overwrite the desired property.

Listed below are all the accepted properties and what they affect.

Property (type)	Description
<b>refreshThrottleTime</b> (integer)	Determines the interval between the <a href="#">Uploads monitor</a> refreshes, in milliseconds. Default value: 1000
<b>addressBook</b> (object)	
<b>minimumBatchSize</b> (integer)	Specifies the maximum number of records to include in a single request to the address book API. Default value: 100

Lite mode is a lightweight version of the ST Web Client. It offers all basic functionality you would expect from the Web Client in a simplistic view.

Because of this, certain features and views in the Lite mode are disabled. For example, in Lite mode you can have only All files view, while the Mailbox view is disabled. As a result, the Address book feature is disabled and you cannot access your contacts.

Other disabled features include the File sharing action, Transfer queue, navigation tabs, ASCII transfer mode, and some Accessibility shortcuts. Also, in Lite mode only the basic navigation with breadcrumbs is available. These changes also apply to the mobile view.

To enable Lite mode, add the following key to the custom `stwebclient.config.json` configuration file.

```
{
  ...
  "liteMode": {
    "enabled": true
  }
}
```

**Note** When you enable Lite mode, it overrides the settings of certain features and views, regardless how they are configured in the custom `stwebclient.config.json` configuration file.

---

# Control the displayed user information

# 11

By default, ST Web Client displays the name of the logged-in user in the **Welcome** menu located in the upper right corner. As an administrator, you can control whether and what user information is displayed there.

## Hide user information

Modify the `stwebclient.config.json` configuration file to add `displayAccountId` property and set `enabled` to `false`.

```
{
  "features": {
    "displayAccountId" : {
      "enabled": false
    },
  },
  ....
}
```

Then, refresh the browser to apply the changes.

## Change the displayed user information

You can specify the user information to be displayed under "Welcome" by adding the `preferredAttribute` property in the `stwebclient.config.json` configuration file and setting its value to your most preferred option.

It takes the following values ordered from highest to lowest precedence:

- `loginName` - displays the login name of the account (corresponds to the **Login Name** field in the account settings)
- `accountName` - displays the name of the account (corresponds to the **Account Name** field in the account settings)
- `emailContact` - displays the email of the logged-in user (corresponds to the **Email Contact** field in the account settings)

As **Email Contact** is an optional field, if it is not filled-in, then the next-highest precedence value will be taken. For example, for the following code snippet where the value is set to `emailContact`:

```
"displayAccountId": {  
    "preferredAttribute": "emailContact"  
}  
....  
}
```

The result would be: If an email address is filled-in in the **Email Contact** field in the account settings, it will be displayed in the **Welcome** menu. If the field is blank, then the login name of the account will be displayed.

After you set the parameter, you need to refresh the browser to apply the changes.

# Access Control

# 12

Use the "fileOperations" section to control the end-users' ability to create, move, rename, or delete files and folders.

All options are of type Boolean with default value `true` (no restrictions).

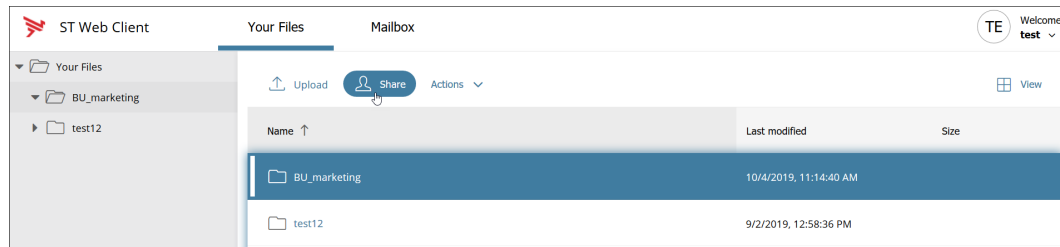
```
{
  ...
  "accessControl": {
    "fileOperations": {
      "createFolder": {
        "enabled": true
      },
      "moveFile": {
        "enabled": true
      },
      "renameFileOrFolder": {
        "enabled": true
      },
      "deleteFileOrFolder": {
        "enabled": true
      }
    }
  },
  ...
}
```

**Note** The above section duplicates server settings and must be maintained together with all related settings in the SecureTransport Administration Tool. The restrictions target only the ST Web Client user interface - the users have other channels to bypass them if the server has more permissive settings.

# Manage folder sharing

# 13

*Share Folder* is a [server-controlled](#) feature. Its availability for end-users is controlled by the `SharedFolders.AllowCollaboration` configuration option in SecureTransportAdministration tool. When it set to `true`, *Share Folder* is enabled. When set to `false`, *Share Folder* is disabled and the **Share** button is not visible.



## Disable Folder sharing from ST Web Client

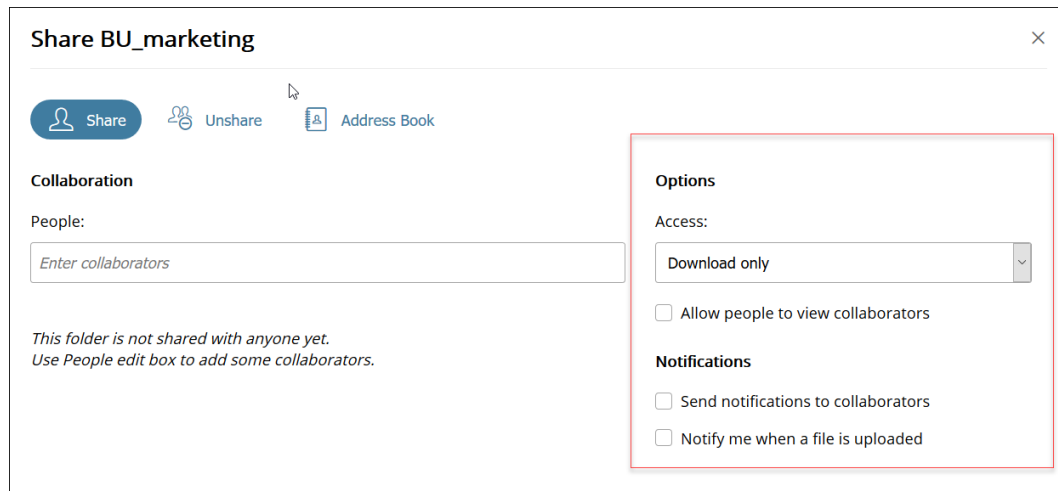
Folder sharing can be disabled from the ST Web Client using the `shareAction` parameter. To do so, add the following section in the custom [configuration file](#):

```
{
  "features": {
    ...
    "shareAction": {
      "enabled": false
    }
    ...
  }
}
```

## Set the default folder sharing options

The *Share* dialog gives end-users the ability to share their folders with other people and perform the following operations:

- set the collaborators access level
- control collaborators' visibility to each other
- configure notification criteria for the shared folder.



As an administrator, you can customize the default setting of the drop-down and the default states of the checkboxes located under **Options** and **Notifications**.

The following code snippet shows their default values in the ST Web Client configuration file; the defaults are applied when a folder is shared for the first time. For already shared folders, the *Share* dialog opens with the existing sharing settings pre-selected.

```
{
  "shareAction": {
    "default": {
      "access": {
        "shareRights": "download",
        "showCollaboratorsToAll": false
      },
      "notifications": {
        "collaborators": false,
        "owner": false
      }
    }
  }
}
```

To change a default setting, in the custom configuration file, add a "shareAction" section and change the value of the corresponding key.

Listed below are all the accepted properties and what they affect.

<b>Property</b> <i>(type)</i>	<b>Description</b>
<b>shareAction</b> <i>(object)</i>	Controls the availability and the default option settings for the Share Folder feature
<b>default</b> <i>(object)</i>	Specifies the default settings for the shared folder's options
<b>access</b> <i>(object)</i>	Controls the collaborator's access level
<b>shareRights</b> <i>(enumeration)</i>	Sets the default value of the <b>Access</b> drop-down Default value: <code>download</code> ; Enumeration values: <code>download</code> , <code>upload</code> , <code>overwrite</code>
<b>showCollaboratorsToAll</b> <i>(Boolean)</i>	Sets the default state of the <b>Allow people to view collaborators</b> checkbox; Default value: <code>false</code> (the check-box is not selected)
<b>notifications</b> <i>(object)</i>	Controls user notifications for the shared folder
<b>collaborators</b> <i>(Boolean)</i>	Sets the default state of the <b>Send notifications to collaborators</b> checkbox Default value: <code>false</code> (the check-box is not selected)
<b>owner</b> <i>(Boolean)</i>	Sets the default state of the <b>Notify me when a file is uploaded</b> checkbox Default value: <code>false</code> (the check-box is not selected)

# Password Management

# 14

Use the "passwordManagement" section to override few restrictions for password management options on the sign-in screen.

```
{
  "features": {
    ...
    "passwordManagement": {
      "enabled": false
    }
  },
  ...
  "passwordManagement": {
    "forgotPasswordButton": {
      "visible": true,
      "customUrl": ""
    },
    "changePassword": {
      "visible": true,
      "customUrl": ""
    }
  },
  ...
}
```

The Password management customization options include:

- "passwordManagement" is the option to enable customizing the Forgotten password options.
- "forgotPasswordButton" is the "**Forgot your password?**" link on the sign-in page.
- "changePassword" is the "**Password**" menu entry in the user menu.

Feature	Description and usage
<b>passwordManagement</b> "enabled": Boolean	This option allows you to enable visibility of all password management options. <ul style="list-style-type: none"> <li>• False (default value) – the section is disabled, all other settings ignored.</li> <li>• True – the settings are applied.</li> </ul>

Feature	Description and usage
<b>forgotPasswordButton</b> "visible": Boolean	This option allows you to enable visibility of the <b>Forgot your password?</b> link on the sign-in page. <ul style="list-style-type: none"> <li>• True (default value) – the link is visible.</li> <li>• False – the link is hidden, all other settings related to <code>forgotPasswordButton</code> are ignored.</li> </ul>
<b>forgotPasswordButton</b> "customUrl": String	This option allows you to set a custom <b>Forgot your password?</b> link on the sign-in page. <ul style="list-style-type: none"> <li>• Default value is an empty string.</li> <li>• If "visible" is true and the value is empty, the setting is ignored and the application has default behavior.</li> <li>• If "visible" is true and the value is a valid URL, changes the behavior of the link to redirect to the specified custom URL. (IDP, etc.)</li> </ul>
<b>changePassword</b> "visible": Boolean	This option allows you to enable visibility of the Password (Change your password) entry in the Welcome menu. <ul style="list-style-type: none"> <li>• True (default value) – the change password entry option in the user menu is visible.</li> <li>• False – the Change password is hidden, all other settings related to it are ignored.</li> </ul>
<b>changePassword</b> "customUrl": String	This option allows you to set a custom URL for the Password (Change your password) web page. <ul style="list-style-type: none"> <li>• Default value is an empty string.</li> <li>• If "visible" is true and the value is empty, the setting is ignored and the application has default behavior.</li> <li>• If "visible" is true and the value is a valid URL, changes the behavior of the change password menu entry to redirect to the specified custom URL.</li> </ul>

---

# Session Management

# 15

Use the "downloadServerPolling" section to prevent UI session timeouts when large files are being downloaded.

To enable "downloadServerPolling":

```
{
  ...
  "sessionManagement": {
    "downloadServerPolling": {
      "enabled": true
    }
  }
  ...
}
```

**Note** Download polling depends on transfers API. The `_Allow this account to submit transfers using the Transfers RESTful API_` option must be enabled for the user.

Custom links appear between last menu item and the 'Logout' one. A separator is inserted before the custom links (can be disabled in configuration). The custom links feature supports translations with i18n module.

## Enable Custom Links feature

To enable Custom Links, in the custom configuration file, add `incuserMenuCustomLinks` and set the value to `true`.

```
{
  "features": {
    ...
    "userMenuCustomLinks": {
      "enabled": true
    },
    ...
  },
  ...
}
```

## Custom links without i18n translations

Add the feature section to the custom configuration file. `insertSeparator` is set to `true` by default; set it to `false` to disable the separator. Each menu item has two mandatory properties - `label` and `url`, and one optional - `useSameWindow`. For security reasons, the `url` property must contain a valid URL (RFC 1738). The default link behavior is to try to open in a new tab (window), depending on browser settings. Set `useSameWindow` property to `true` to open the link in the same window.

A sample configuration with two links and no separator:

```
{
  ...
  "userMenuCustomLinks": {
```

```

    "insertSeparator": false,
    "items": [{
      "label": "Google search",
      "url": "http://google.com"
    }, {
      "label": "Company policy",
      "url": "https://mycompany.com/policy",
      "useSameWindow": true
    }]
  }
  ...
}

```

## Custom links with i18n

To enable translations via i18n module, follow the steps:

1. Add an `i18n` section to `userMenuCustomLinks` configuration and set `enabled` to `true`.
2. Define the section name using the `sectionName` property.  
This section name will be used in translation files in the next steps. We recommend that you follow the `snake_case` naming convention which is also used in `translation.json` file.
3. Add custom link definitions. For each item use the `label` property as a translation key - set a unique property value.

**Note** DO NOT USE SPACES OR SPECIAL CHARACTERS AS KEY VALUES! We recommend that you use `snake_case`.

**Good:** "my\_link", "link\_to\_google", "myLink"

**Bad:** "my link", "1stlink", "my-link", "my,link"

Sample configuration:

```

{
  ...
  "userMenuCustomLinks": {
    "i18n": {
      "enabled": true,
      "sectionName": "custom_links"
    },
    "items": [{
      "label": "link1",
      "url": "http://google.com"
    }, {

```

```

    "label": "link2",
    "url": "https://mycompany.com/policy"
  }}
}
...
}

```

4. Setup override for the default translation. Follow the steps as described in [Add and modify languages on page 14](#).

In `custom/locales/en/override.json`, add the previously defined custom links section:

```

{
  ...
  "custom_links": {
    "link1": "Google search",
    "link2": "Company policy"
  }
}

```

When done, the links should appear with their texts in user menu for the default locale.

**Note** If the keys are displayed instead of the texts, double-check all settings.

5. Add a translation for another language - see [Add and modify languages on page 14](#). Use the same section name and keys.

Modify `custom/locales/<your_locale>/translation.json`:

```

{
  ...
  "custom_links": {
    "link1": "Гугъл търсене",
    "link2": "Политика на компанията"
  }
}

```

The final configuration should look like this:

```

{
  "features": {
    "userMenuCustomLinks": {
      "enabled": true
    },
  },
}

```

```
"userMenuCustomLinks": {
  "i18n": {
    "enabled": true,
    "sectionName": "custom_links"
  },
  "items": [{
    "label": "link1",
    "url": "http://google.com"
  }, {
    "label": "link2",
    "url": "https://mycompany.com/policy",
    "useSameWindow": true
  }]
},
"i18n": {
  "additionalLocales": [{
    "id": "bg",
    "nameEng": "Bulgarian",
    "name": "Български"
  }],
  "overridenLocales": ["en"]
},
...
}
```

If all settings are correct, the links' names should be properly displayed for the selected locale.

---

# Pre-login disclaimer banner 17

You can configure a disclaimer banner that is displayed to users before they log in. The banner can contain legal statements that must be accepted before a user can continue to the logon process.

## Enable the disclaimer feature

To enable the disclaimer feature, add a `disclaimer` section to the features configuration and set `enabled` to `true`.

```
{
  "features": {
    ...
    "disclaimer": {
      "enabled": true
    }
    ...
  },
  "disclaimer": {
    "showAlways": false
  }
}
```

The disclaimer appearance is controlled by the `showAlways` parameter. `showAlways` is set to `false` by default which means that the disclaimer appears once per user/browser. The user confirmation is stored locally along with a content hash to ensure the disclaimer will pop again when its content changes.

To configure a disclaimer that appears before every login attempt, set `showAlways` to `true`.

**Note** To prevent accidental confirmation of the disclaimer by users without ST Web Client credentials, the confirmation is stored permanently only after the user logs into ST Web Client.

## Set up the disclaimer banner

The disclaimer banner has two controls: for the text and the button respectively. Since the disclaimer text can be long, it should be placed inside a separate file, and the file name must be added to `i18n` settings.

To set up the disclaimer text, follow the steps below:

1. Put your disclaimer text into a file. The format can be either text or HTML (only basic formatting tags are supported: `p`, `div`, `span`, `br`, `b`, `strong`, `i`, `u`).

`disclaimer.html`

```
<p>Please, read the terms and conditions below....</p>
Additional text
```

2. Put the file in the `custom/locales/en/` folder.
3. Set up override for the default translation by following the steps as described in [Add and modify languages on page 14](#).

In `custom/locales/en/override.json`, define the disclaimer section:

```
{
  ...
  "disclaimer": {
    "button_label": "I agree",
    "content_file_name": "disclaimer.html"
  }
}
```

The final configuration should look like this:

```
{
  "features": {
    "disclaimer": {
      "enabled": true
    }
  },
  "i18n": {
    "overridenLocales": ["en"]
  },
  ....
}
```

Once done, you should see the disclaimer banner on the login page.

## Add translations for other languages

1. Modify `custom/locales/<your_locale>/translation.json`. For more information, see [Add and modify languages on page 14](#).

It should look like this:

```
{
  ...
  "disclaimer": {
    "button_label": "Съгласен съм",
    "content_file_name": "disclaimer_bg.html"
  }
}
```

2. Translate the disclaimer text and place the file next to `translation.json`.  
`disclaimer_bg.html`:

```
<p>Моля, прочетете условията.....</p>
Още съдържание...
```

**Note** The name of the disclaimer content file can be the same because each locale is placed in its own folder; the "\_bg" suffix in the example is used only for clarity.

If everything is correct, you should see the disclaimer properly translated for the chosen locale.

---

# Customizations after a patch or upgrade 18

Most customizations (feature toggles, localization, branding texts, and view configuration) do not have to be recreated after patching or upgrading SecureTransport. The exceptions are the icon and the color theme. To replace the icon, see [Change the favicon on page 18](#). To recreate the color theme, see [Change the color theme and/or the style on page 18](#).

If SecureTransport is uninstalled, the `custom` folder will be removed. Before uninstalling SecureTransport, back up the `custom` folder. The backed up `custom` folder can be used in future SecureTransport installations.