

[Collapse All](#) | [Expand All](#)

- › [Home](#)
- › [Archived](#)
- › [Acknowledgements](#)
- [Blog and News](#)
- › [Documentation v2-0-alpha](#)
- ▼ [Documentation v1-6](#)
 - [Features](#)
 - [Upgrade notes](#)
 - [Known issues](#)
 - [Requirements](#)
 - Administrator reference manual**
 - [Installation - Linux Source](#)
 - [Installation - Debian Ubuntu](#)
 - [Installation - RPM](#)
 - [Mail Bounce Handling](#)
 - [Configuring SRS with Exim \(Debian and Ubuntu\)](#)
 - [Setting up usage statistics](#)
 - [FAQ](#)
 - [End User Documentation for v1-6](#)
- › [Documentation v1-5](#)
- › [Documentation v1-1](#)
- › [Download](#)
- › [Developer](#)
 - [Existing installations](#)
 - [Feature requests](#)
 - [Online Demo](#)
- › [Project organisation](#)
 - [Roadmap](#)
- › [Release Schedule](#)

[Release status and life cycle](#)

[Reviews and prototypes](#)

[Support and Mailinglists](#)

[Twitter](#)

[FileSender's workflows](#)

[Installation - Linux Source 2-0-Alpha-from-svn](#)

Administrator reference manual



Version 36, last updated by [meijer](#) at 2015-02-06

This is the Administrator Reference Manual for FileSender version 1.6.

- [1. Administering FileSender](#)
 - [1.1 Installation and initial configuration](#)
 - [1.2 Regular tasks](#)
 - [1.3 The Administrator web interface](#)
 - [1.4 Troubleshooting](#)
- [2. Authenticating users](#)
 - [2.1 IdP Attributes](#)
- [3. Security considerations](#)
 - [3.1 Warning - Encryption](#)
- [4. Customising the look and feel of your FileSender installation](#)
 - [4.1 Banner](#)
 - [4.2 Footer](#)
 - [4.3 Favicon website icon](#)
 - [4.4 Site name](#)
 - [4.5 Landing page splash text](#)
 - [4.6 Various UI settings toggles](#)
 - [4.7 Date formats](#)
 - [4.8 Terms and Conditions \(AuP\) controls](#)
 - [4.9 Custom help and about](#)
 - [4.10 Custom logout URL](#)
 - [4.11 Email customisation](#)
 - [4.12 Customising SimpleSAMLphp](#)
- [5. Dealing with language files and localisation](#)
 - [5.1 Internationalisation and character encoding](#)
 - [5.2 Technical details about the language file mechanism](#)
 - [5.3 Customising languages](#)
- [6. Technical details about pause - resume functionality](#)
- [7. Configuration file directives](#)
 - [General settings](#)
 - [User Interface \(UI\) settings](#)
 - [Debug settings](#)
 - [SAML attribute settings](#)

- [Acceptable Use Policy \(AUP\) settings](#)
- [Server settings](#)
- [Advanced server settings](#)
- [Site URL settings](#)
- [Support links](#)
- [File and directory location settings](#)
- [Database related settings](#)
- [Cron settings](#)
- [Email related configuration directives](#)
- [Email template variables](#)
- [Not implemented/unused](#)

1. Administering FileSender

1.1 Installation and initial configuration

Consult the relevant installation guide:

- [Installation - Linux Source](#)
- [Installation - Debian Ubuntu](#)
- [Installation - RPM](#)

1.2 Regular tasks

1. **Make sure the daily expiry cron job runs.** This is needed to delete files and vouchers that have expired, as well as to remove stale database entries from cancelled uploads. The debian and RPM packages will install a daily cronjob for you in `/etc.cron.daily/filesender`. When installed from source make sure you have installed a cronjob as per the [Configure cron](#) section in the Installation Guide.
2. **Clean temp_filestore and log directories.** FileSender does not do any of this automatically so you as a server admin need to ensure this happens. The temp_filestore tends to get filled with files that are leftover when a user cancels an upload. Log files are created on a daily basis but are not automatically purged.
3. **Check logfiles for errors.**
4. **Database cleanup:** every upload to FileSender results in one or more database entries. Not all of these are deleted which could lead to performance degradation in the long run. You should check your database what the situation is and whether it needs cleanup of very old entries.

1.3 The Administrator web interface

The administrator web interface is accessed through the regular FileSender user interface. An 'Administration' button becomes visible next to MyFiles when you log on with an account that has a GUID (like `eduPersonTargetedId` or `eduPersonPrincipalName`) configured in `$config['admin']`. The administrator web interface provides an easy overview of the use of your FileSender installation.

1.4 Troubleshooting

Relevant logfiles to check

Relevant logging can appear at three places:

1. the FileSender logdirectory as configured in your config file with `$config['log_location']`
2. your webserver logs
3. your PHP logs (as per the relevant php.ini settings, often your `/var/log/messages` or `/var/log/syslog` file)

! If there is no logging in your FileSender log directory, make sure the web server can write to it.

! If nothing works you might have issues with connecting to your database. Increase database logging to study interaction between your database and FileSender.

Getting more debugging information

In case of problems you can turn on debugging in the FileSender configuration (see [Debug settings](#) for the relevant directives). Setting `$config['debug'] = true` will turn on extra logging (both for FileSender in the FileSender

log-directory and PHP) and additionally will show some extra technical information.

Note that turning on the debug settings will decrease performance so don't forget to turn off the debugging after investigation.

2. Authenticating users

FileSender **has no built-in local user database**. It uses the [SimpleSAMLphp](#) software to connect to your authentication source of choice. With SimpleSamlPHP you can connect to most authentication sources you can imagine. RADIUS, LDAP, SQL, flat file, SAML 2.0 IdPs, Shibboleth, YubiKey, Twitter, Google: whatever you need to chances are SimpleSAMLphp will let you use it, out of the box. Check [SimpleSAMLphp's list of supported authentication modules](#) for details.

The default FileSender installation and configuration uses Feide OpenIDP. This ensures you'll be up and running within the hour and can test your FileSender install. To connect your FileSender installation to your authentication source(s) of choice, please check the [SimpleSAML Service Provider Quickstart](#). If you have a question about how to configure SimpleSAMLphp, please consider asking your question on [the SimpleSAMLphp mailinglists](#)

In SimpleSAMLphp terms, your FileSender + SimpleSAMLphp installation is a **Service Provider (SP)**. It uses one or more **Identity Providers (IdP)** to do the authentication and provide attributes to FileSender. This Identity Provider will supply the FileSender software with information about the user: attributes.

2.1 IdP Attributes

FileSender **needs** two attributes to work:

- something that uniquely identifies every user (saml_uid_attribute)
- something that supplies the user's email address (saml_email_attribute)

An error "*Your Identity Provider is not providing the required attributes*" will be thrown if FileSender doesn't receive these attributes when a user logs in. If you are using Shibboleth as an IdP please check the note on Shibboleth below.

You may supply an **optional** third attribute:

- something that supplies the user's "friendly name" (saml_name_attribute)

Please consult `$config['displayUserName']` for details.

How to configure IdP Attributes

You configure the attribute name mappings in the config.php file, the [SAML attribute settings](#) section). The default configuration is:

```
$config['saml_uid_attribute'] = 'eduPersonTargetedID'; // Attribute to uniquely ident
$config['saml_email_attribute'] = 'mail'; // Attribute used for email address
$config['saml_name_attribute'] = 'cn'; // Attribute used to get the user's name
```

Example

A typical change you might want to consider is using 'eduPersonPrincipalName' as the saml_uid_attribute in stead of eduPersonTargetedId, as it makes debugging easier. Please be aware you open your installation for the risk of session hijacking through rogue IdPs. Check with your login providers

```
$config['saml_uid_attribute'] = 'eduPersonPrincipalName'; // Attribute to uniquely id
```

More information on IdP attributes

You can check which attributes are received by your FileSender installation on the SimpleSAMLphp pages, in most cases that will be at <https://<your-server>/simplesaml/module.php/core/authenticate.php>. Login from that page to get a list of known attributes and their names and values.

If your IdP doesn't send you any attribute uniquely identifying a user, you can add an authproc filter to the metadata of

that Identity provider to locally generate an eduPersonTargetedID attribute. See http://simplesamlphp.org/docs/1.8/core:authproc_targetedid for the details.

A note on using Shibboleth as an IdP

Shibboleth supplies attribute names not as strings but as object identifiers in URN notation. For example the email attribute will be 'urn:oid:0.9.2342.19200300.100.1.3'.

You can deal with this in two ways. Either you create an attribute mapping in your simplesaml configuration which says oid2name, or you just tell FileSender that it's to expect OIDs rather than names for attribute identifiers. Example:

```
$config['saml_email_attribute'] = 'urn:oid:0.9.2342.19200300.100.1.3'; // Attribute used for ema
```

```
$config['saml_name_attribute'] = 'urn:oid:2.5.4.42'; // Attribute used to get the user's name
```

```
$config['saml_uid_attribute'] = 'urn:oid:2.16.840.1.113730.3.1.241'; //Attribute to uniquely ide
```

3. Security considerations

3.1 Warning - Encryption

FileSender itself does **not** offer protection against eavesdropping on the contents of uploaded/downloaded files.

Sensitive files can currently be protected in two ways:

- in flight: protect files during upload/download using SSL. Ensure your FileSender webserver is only reachable over HTTPS.
- in rest: files stored on the FileSender server should be encrypted by the user **before** uploading them.

4. Customising the look and feel of your FileSender installation

With FileSender version 1.5 you can customise more than with version 1.1. Customisations not mentioned in this section will require changing the code and will not survive upgrades or re-installs unless you keep a good backup of all your modifications. Config file directives you find in config/config.php and start with \$config. Language file directives you find in the language files and start with \$lang.

4.1 Banner

A banner picture is automatically displayed at the top of all web pages.

default banner

```
<your_filesender_dir>/www/banner.png
```

custom banner

You override the default banner with a custom banner image named **banner.png** placed in the config directory. This image has to be in PNG format and should be sized to 800x60 pixels

```
<your_filesender_dir>/config/banner.png
```

4.2 Footer

As of version 1.6 you can configure your own footer. This is done in the language files. Check the language file section on how to create localised language files that won't be overwritten during an upgrade.

Example:

```
// Footer to display
$lang["_SITE_FOOTER"] = "";
Example: $lang['_SITE_FOOTER']='Version '.FileSender_Version::VERSION;
```

4.3 Favicon website icon

A **favicon** is the website icon usually displayed in the URL bar right next to the website's address.

The **favicon.ico** is found in <your_filesender_dir>/www/favicon.ico. It can currently not be customised in the same way as the banner can. This is functionality planned for version 2.0.

4.4 Site name

Use the **\$config['site_name']** directive to specify a custom friendly name to your FileSender instance. The site_name is used at various places, amongst others in the default splash text, the help, the title bar and Subject: line and text body of all email messages.. Check [Configuration file directives](#) for details.

4.5 Landing page splash text

You can display a custom text on the main login page, also known as the landing page. This text is specified in the language file(s) active on your installation. Use **\$lang["_SITE_SPLASHHEAD"]** to specify a custom header for your splash text, and **\$lang["_SITE_SPLASHTEXT"]** to specify the splash text.

Example:

```
// Login Splash text
$lang["_SITE_SPLASHHEAD"] = "<p style='text-align:center;'>" . htmlspecialchars($con
$lang["_SITE_SPLASHTEXT"] = "<p style='text-align:center;'><i><font size='"+1">sha
Logon to upload your files or invite people to send you a file.<br><br></p>
<p><b>Latest news:</b><br>14-11-2012:CloudStor was upgraded to a new software version
```

4.6 Various UI settings toggles

You can toggle the version number (**\$config["versionNumber"]**), total up/download counters (**\$config['site_showStats']**) and the "Welcome <user>" friendly username (**\$config['displayUserName']**). Check the [User Interface \(UI\) Settings](#) section for details.

4.7 Date formats

There are three places where date formats are configured. If you want to customise the date format make sure to change **all three** to ensure a consistent date format experience for your users.

- config.php: **\$config['datedisplayformat']** = "d-m-Y"; // used for date format in emails
- language file (e.g. en_AU.php): **\$lang['datedisplayformat']** = "d/m/Y"; // Format for displaying date/time, use PHP date() format string syntax
- language file (e.g. en_AU.php): **\$lang["_DP_dateFormat"]** = 'dd/mm/yy';

4.8 Terms and Conditions (AUP) controls

FileSender supports an AuP tick box. You can make clicking this tick box mandatory, you can configure it to be ticked by default or you can switch it off entirely. The AuP text is specified in the language file(s) active on your FileSender instance. Check the [Acceptable Use Policy \(AUP\) settings](#) section for \$config options.

Example AUP text:

```
$lang["_AUPTERMS"] = "1. Your use of the UNINETT CloudStor Service (Service) is gover

2. The Service is governed by Norwegian law. You will only use this Service for purpo

3. You will refrain from any use that threatens the availability of the Service<br><b

4. The Service is provided \"as-is\" and delivered on a best-effort basis, any warran
```

4.9 Custom help and about

method 1: Modal inline popup (default method)

When a user clicks on "help" or on "about", by default a window will pop up with text drawn from the relevant language file. The text is specified in `$lang["_HELP_TEXT"]` and `$lang["_ABOUT_TEXT"]`.

NOTE: the help and about windows are modal inline dialog windows, not popups that trigger popup blockers.

If you want to customise the help and/or about text for this method, you need to create a local override language file and specify a customised `$lang["_HELP_TEXT"]` and/or `$lang["_ABOUT_TEXT"]` in this customised language file.

NOTE: when you change the help and about text, you probably will want to limit the available language files to only those containing your customised help and about!

method 2: Link to specific (external) HTML pages

Like with previous versions you can still simply link a help and/or about page. Check the [Support links](#) for details.

4.10 Custom logout URL

The `$config['site_logouturl']` setting (in the [Site URL settings](#) section) can be used to point to a custom page to return to after the authenticated user has logged out. Note that the actual use of this page also depends on the Identity Provider and/or Federation of the authenticated user, not all of them redirect back to the requested logout page.

4.11 Email customisation

You can customise all the emails sent out by FileSender. The email templates are configured in the configuration file.

check the [Email related configuration directives](#) and the [Email template variables](#) section for details.

4.12 Customising SimpleSAMLphp

In case you use the local SimpleSAMLphp 'Identity Provider' selection page you can have a look at the [themefilesender module](#). This module will give SimpleSAMLphp the same look&feel as FileSender.

5. Dealing with language files and localisation

As of version 1.5, FileSender supports multiple languages. All web-UI texts are taken from the language file matching the browser's indicated language preference.

- out-of-the-box (distribution-provided) the following language files are included: **Czech, Croatian, Dutch, English (Australian), French, German, Hungarian, Italian, Norwegian (Bokmål), Serbian, Slovenian and Spanish.**
- **NOTE: do not change these distribution-provided language files!** This will give you a head-ache when upgrading. Read section 5.2 on customising language tags.
- the filesender-core team maintains Dutch, English and Norwegian, the other files are maintained by contributors.
- the distribution-provided language files are located in the directory **<filesender-root>/languages**, one file per language.
- if a browser language preference is sent for which no language file is present, the default language file is selected.

5.1 Internationalisation and character encoding

FileSender encodes all input and output text in UTF-8. There should be no problems encoding any characters or languages.

5.2 Technical details about the language file mechanism

The language directory contains the distribution files for the various language definitions as well as a file "locale.php" controlling which language files can be served up. Definitions and mappings in these files can be overruled by a corresponding file (with the same name) in the main ./config directory. Local customisations should be done in these files in the ./config directory. See <http://www.assembla.com/spaces/filesendercontrib/> on how to report errors in the translations or contribute new translations. The "./scripts/check_language_tags.sh" script can be used to identify untranslated items in customised language files.

The naming scheme of the language files in this directory and mapping of browser tags to language files is based on Best Current Practices as defined by [IETF BCP 47: Tags for Identifying Languages](#) and related ICU/ISO guidelines.

File naming

<language>_<COUNTRY>.php

<language> is the shortest two/three letter ISO 639 language code in lowercase
<COUNTRY> is the ISO 3166-1 country/region/territory code in UPPERcase

Use of "_" (underscore) in filenames follows the convention for defining 'locales' on modern Linux/Unix systems and defined/allowed by the related Unicode/ICU definitions, see http://www.unicode.org/reports/tr35/#Unicode_Language_and_Locale_Identifiers and <http://userguide.icu-project.org/locale>).

Browser tags mapping

Mapping from browser language tags to language files is done in locale.php. Note that browser tags for language preference use the "-" hyphen as defined in [IETF BCP 47: Tags for Identifying Languages](#). Tags specified in locale.php should be all lowercase.

5.3 Customising languages

You can easily adapt relevant language labels to your local needs in an upgrade-friendly way, for example to localise the splash screen text, the AUP text, the help text, etc. When you adapt certain language labels you will probably want to limit the available languages on your site to only those that you in fact translated. This is also easily done.

Warning: the texts specified in a language file are sent to the user's browser as-is. Unfiltered. Make sure you run your customised language files through a HTML validator!

Limiting what languages your installation supports

Copy language/locale.php to your ./config/ directory and edit it to support only the languages you want. For example, UNINETT currently supports two languages: English and Norwegian - Bokmål:

```
root@cloudstor:/var/www/filesender/# cat config/locale.php
<?php

//The below array maps locales provided by the browser to your local translation.
//Put your custom translated file in the "config" subdirectory.
//Add relevant entries for browser locales here, and your translation will automagically work.

// Uninett supports English and Norwegian - Bokmål.

$locales = array(
    "en" => "en_AU.php",
    "en-au" => "en_AU.php",
    "no" => "no_N0.php",
    "nb" => "no_N0.php",
    "nb-no" => "no_N0.php",
);
?>
```

Customising certain language texts

All language texts come in the form of a language label \$lang["<label>"] = "text to be displayed".

You probably do not want to change all language labels unless you'd like to add a new language. To customise

certain language labels, create a file in your `.config/` directory **with exactly the same name as the language file you'd like to over-ride** and put the language labels + texts you'd like to be used in that file. Files in the `.config` directory are NOT overwritten when upgrading.

As an example here's the over-ride file UNINETT uses for its FileSender production service:

```
root@cloudstor:/var/www/filesender/config# more en_AU.php
<?php

/* -----
 * Overrides to en_AU Language File
 * -----
 */

$lang["_LOGON"] = 'Logon with <br><br> <br />Please contact support@example.com.";
$lang["_DISK_SPACE_ERROR"] = "There is not enough drive space on this service. This s
support@ecampus.no and/or upload a smaller file.";
$lang["_ERROR_ATTRIBUTES"] = "Your institution is not providing the required identity
t support@example.com";
// vouchers
$lang["_SEND_NEW_VOUCHER"] = "A Voucher allows someone to send you a file.<br />
To create a voucher, enter an email address then select Send Voucher.<br />
An email will be sent to the recipient with a link to use the Voucher.";

// Login Splash text
$lang["_SITE_SPLASHHEAD"] = "<p style=\"text-align:center;\">>". htmlspecialchars($co
$lang["_SITE_SPLASHTEXT"] = "<p style=\"text-align:center;\">><i><font size=\"+1\">sha
font></i><br>
Logon to upload your files or invite people to send you a file.<br><br><br></p>";
// <p><b>Latest news:</b><br>14-11-2012:CloudStor was upgraded to a new software vers
us.no/2012/11/14/cloudstor-oppgradert/\">>the eCampus blog article</a> for details! <
href=\"mailto:support@example.com\">support@example.com</a>.</i> </p>";

// site about
$lang["_ABOUT_TEXT"] = ' <div align="left" style="padding:5px">'. htmlspecialchars($co
lation of FileSender (<a rel="nofollow" href="http://www.filesender.org/" target="_bl
is developed to the requirements of the higher education and research community.</di

// site AUP terms
$lang["_AUPTERMS"] = "1. Your use of the UNINETT CloudStor Service (Service) is gover
tions<br><br>

2. The Service is governed by Norwegian law. You will only use this Service for purpo
aws and regulations, in particular those laws and regulations pertaining to intellect
<br>

3. You will refrain from any use that threatens the availability of the Service<br><br>

4. The Service is provided \"as-is\" and delivered on a best-effort basis, any warran
urpose are disclaimed.<br>";

?>
```

6. Technical details about pause - resume functionality

Placeholder to document additional technical details about pause - resume functionality. For now we'll suffice to say it works based on the file name and the file size, and that it only works with the HTML5 uploads.

7. Configuration file directives

All configuration of FileSender is done by editing the configuration file:

config/config.php

This file is actual PHP code so be careful to adhere to the PHP syntax:

- keep the closing ; at the end of a \$config specification
- enclose string/text values in single or double quotes
- boolean values can be specified as true or false (without quotes)
- remarks/comments start with // or are enclosed in /* ... */
- make sure you don't have any extra characters after the ?> at the end of the php code

This section describes all the available configuration settings in config.php.

General settings

\$config['admin']

description: list of UIDs of administrators, separated by comma. Administrators have access to the Administrator Interface. A UID is the value of the attribute defined in `$config['saml_uid_attribute']` in the [SAML attribute settings](#) section.

default: none

example: '31a7bf626eb4fa2d425f611c0490940e71e1c42c' (with `eduPersonTargetedID` as UID attribute)

example: 'john.doe@mysite.dom' (with `mail` as UID attribute)

\$config['adminEmail']

description: Email address(es, separated by ,) to receive administrative messages (low disk space warning)

default: none

\$config['Default_TimeZone']

description: Set this to your local timezone according to timezone specifications in <http://php.net/manual/en/timezones.php>

default: 'Australia/Sydney'

example: 'Europe/Berlin'

\$config['Default_Language']

description: Set the default language to use when the user's browser wants a language that's not available. For available languages see the language directory on your FileSender installation. In case you make a typo here, there is a last resort hard-coded fallback to `en_AU`.

default: 'en_AU'

example: 'no_NO'

\$config['site_name']

description: Friendly name used for your FileSender instance. Used on the main login page, the title bar, the help and in emails.

default: 'FileSender'

example: 'AARNet CloudStor'

\$config['site_splashtext']

deprecated. You now set this in the language files using `$lang["_SITE_SPLASHHEAD"]` and `$lang["_SITE_SPLASHTEXT"]`.

User Interface (UI) settings

`$config['datedisplayformat']`

description : Format for displaying date/time. Use the [PHP date \(\) format string syntax](#).

1.5 default : "d-m-Y"

1.5 example : "d-m-Y"

Note : the semantics of this directive changed with version 1.5.

`$config["versionNumber"]`

description : show FileSender version number in the upper right corner

values : true/false

default : true

`$config['site_showStats']`

description : show upload/download statistics up in the upper right corner.

default : false

`$config['displayUserName']`

description : show 'Welcome user' in the upper left corner. The friendly name of the authenticated user is taken from the SAML attribute as specified by `$config['saml_name_attribute']` or will be 'Guest' for users using a voucher.

values : true/false

default : true

\$config["autocomplete"]

description : provides an auto-complete function in the recipient email field (To:). It shows previous recipients **this** authenticated user has sent files to before. For this it queries the Files table (!).

values : true/false

default : true

note : available since version 1.6

\$config["autocompletehistory"]

description : max. number of results displayed in auto-complete.

values : "" for unlimited, "<integer>" for limit

default : true

note : available since version 1.6

Debug settings

\$config["debug"]

description: if set to true server side logging will be more verbose and the UI will display extra technical information of what's happening. This setting also increases the PHP loglevel.

values: true/false

default: false (true in beta releases)

\$config['dnslookup']

description : if set to true the logging includes a reverse DNS lookup of the originating IP-address of the user. On really busy servers this may impact performance.

default : true

values : true/false

SAML attribute settings

\$config['saml_email_attribute']

description: . attribute used as From: email address of the authenticated user

default: mail

\$config['saml_name_attribute']

description: attribute used to get the friendly name of the authenticated user. If this attribute does not exist or has no value (?) then FileSender will use the first part of the email attribute before the @ as the 'name_attribute' value.

default : cn

\$config['saml_uid_attribute']

description: attribute to uniquely identify the user.

default : eduPersonTargetedID

Acceptable Use Policy (AUP) settings

\$config["AuP_default"]

description : when set to false (the default) the user has to explicitly tick the 'I accept the AuP' checkbox before uploading a file. When set to true the checkbox will already be ticked.

default : false

\$config["AuP"]

description : a text box with an AuP is displayed

default : true

Server settings**\$config['default_daysvalid']**

description : Maximum number of days before a file or a voucher is expired

default : 20

\$config['ban_extension']

description : Possibly dangerous file extensions that are disallowed

default : = 'exe,bat'

\$config["max_email_recipients"]

description : maximum email addresses allowed to send at once for voucher or file sending, a value of 0 allows unlimited emails.

default : 100

\$config["confirmation_to_downloader"]

description : if true, sends a copy to the file recipient of each "your file has been downloaded" confirmation email that the Sender gets when the recipient downloads a file. Defaults to true as that's the behaviour in version 1.5.

values : true/false

default : true

availability : since version 1.6

\$config['max_flash_upload_size']

description : the maximum files size when uploading with Flash (browser with no HTML5 support), specify in number of bytes.

default : '2147483648' (2 GigaByte)

remarks : the value specified will be compared with the PHP settings `post_max_size` and `upload_max_filesize`. The minimum of these values will be used as actual maximum upload size for Flash uploads

\$config['max_html5_upload_size']

description : the maximum file size when uploading with HTML5, specify in number of bytes

availability : version 1.5

default : '107374182400' (100 GigaByte)

```
$config["upload_chunk_size"] = '2000000';//
```

description: uploads using HTML5 happen one chunk at a time. This directive specifies how big each chunk is. When using long latency paths and/or when sending very large files an increased chunk size might lead to an improved transfer speed experienced.

availability: version 1.5
default: '2000000';

```
$config["terasender"]
```

description : enables the TeraSender fast upload module. teraSender uses HTML5 web workers to do parallel chunk uploads. Multi-threading fast uploads.

values : true/false

default: false

availability: since version 1.6

```
$config["terasenderadvanced"]
```

description : display advanced terasender settings in UI for user to change. Useful when trying to figure out the best settings for a particularly demanding user.

values : true/false

default: false

availability: available since version 1.6

```
$config["terasender_chunksize"]
```

description : size of each TeraSender worker thread chunk.

values : integer

default: 5

availability: available since version 1.6

```
$config["terasender_workerCount"]
```

description : worker count TeraSender will use. Check what the number of workers is that's actually supported in a particular browser.

values : integer

default: 6

availability: available since version 1.6

```
$config["terasender_jobsPerWorker"]
```

description: number of jobs per TeraSender worker. Testing early 2013 did not indicate a larger job size has any influence on throughput so your mileage may vary.

values: integer

default: 1

availability: available since version 1.6

Advanced server settings

This section contains settings that under normal circumstances don't need changing. **Do not change them unless you have a very good reason.**

\$config['db_dateformat']

description: Date/Time format for your database. For PostgreSQL, use [PHP date format specifier syntax](#)

default: 'Y-m-d H:i:sP'

\$config["crlf"]

description: controls whether lines in emails sent out by FileSender are terminated with CRLF or with LF. What makes sense depends on your PHP and email setup. Linux/Unix hosts will need "\n". Possible values are "\n" or "\r\n". Keep this at the default unless you have some weird setup.

default: '\n'

\$config['voucherRegEx']

description: When someone uses a voucher, the voucher is in the format of 8 numbers/letters followed by - 4numlet-4numlet-4-12numlet unique id that's standard flash and windows GUID (generated unique id). The regex is to make sure that a presented voucherid adheres to this GUID format. Anything else will result in a 'bugger off'. Also makes sure that generated vouchers are constructed correctly. Reason for putting it in config is because the PHP uniqueid is a different format so you'd need a different RegEx. So this is future proofing the vouchers.

default: '[a-zA-Z0-9]{8}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{4}-[a-zA-Z0-9]{12}'

\$config['voucherUIDLength']

description: See above. Make sure no-one tries anything 'interesting' with vouchers.

default: '36'

\$config['emailRegEx']

description: Regular expression to check the syntax of email-addresses

default: '[a-z0-9!#\$%&*+/?^_{}~]+(?:\\.[a-z0-9!#\$%&*+/?^_{}~]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?'

Site URL settings

These settings define the basic URL's FileSender will use. The configuration file will in general automatically define them based on the server name of the webserver which is automatically set in the global PHP variable `$_SERVER['SERVER_NAME']`. The following piece of code in config.php handles the automatic configuration:

```
if ( isset($_SERVER['SERVER_NAME']) ) {
    $prot = isset($_SERVER['HTTPS']) ? 'https://' : 'http://';
    $config['site_url'] = $prot . $_SERVER['SERVER_NAME'] . '/filesender/'; // URL to FileSender
    $config['site_simplesamlurl'] = $prot . $_SERVER['SERVER_NAME'] . '/simplesaml/';
    $config['site_authenticationSource'] = "default-sp";
    $config['site_logouturl'] = $config['site_url'] . '?s=logout';
}
```

Settings within this block can be modified as long as you keep those settings within the block.

\$prot

description : variable containing the automatically determined current protocol (http or https) of the connection

default : `isset($_SERVER['HTTPS']) ? 'https://' : 'http://'`

remark : do not change

\$config['site_url']

description : URL on your webserver where the FileSender instance can be reached

default : `$prot . $_SERVER['SERVER_NAME'] . '/filesender/'`

remarks : URL is automatically generated based on the SERVER_NAME and a manually specified local path ('/filesender/' by default). If you have your FileSender running on another URL you can change the local path part. At a minimum, this part should contain '/' which means FileSender is running at the root of your webserver.

example : `$prot . $_SERVER['SERVER_NAME'] . '/'`

\$config['site_simpleamlurl']

description : URL on your webserver where the SimpleSAMLphp instance can be reached

default : `$prot . $_SERVER['SERVER_NAME'] . '/simpleaml/'`

remark : URL is automatically generated based on the SERVER_NAME and a manually specified local path ('/simpleaml/' by default). If you have your SimpleSAMLphp instance running on another URL you can change the local path part. If you have your FileSender instance reachable with http (no forced SSL) but want to force authentication via simpleaml to use SSL change the \$prot part to 'https://' (including quotes).

example : `'https://' . $_SERVER['SERVER_NAME'] . '/simpleamlphp/'`

\$config['site_authenticationSource']

description : authentication source on your SimpleSAMLphp instance to use

default : "default-sp"

example : "example-userpass" ; Point FileSender to the 'example-userpass' authentication source defined in the SimpleSAMLphp authsources.php

\$config['site_logouturl']

description : URL to go to when the user logs out (will be used by the IdP to redirect to)

remark : should be an absolute URL including protocol and host, by default this will be the URL part as generated in \$config['site_url'] plus a local path part (default '?s=logout')

default : `$config['site_url'] . '?s=logout'`

example : `$config['site_url'] . 'custom-logout.php'`

\$config['site_downloadurl']

description : in beta release before 0.1.16 this was used for download URLs. Obsoleted as of beta 0.1.16

status : Obsoleted

The forceSSL setting is outside the `if {}` block but is related to the server URL settings:

\$config['forceSSL']

description : when set to true (default, recommended) FileSender will force connections to always use SSL

values : true/false

default : true

Support links

These settings control what happens when a user clicks on the "Help" or "About" buttons.

`$config['aboutURL']`

description : if "", a modal inline popup dialogue is shown with the contents of `$lang['_ABOUT_TEXT']`. Alternatively a URL can be used to point to a specific (local or external) HTML page.

default : ""

example : "http://support.example.org/about-filesender.aspx"

`$config['helpURL']`

description : if "", a modal inline popup dialogue is shown with the contents of `$lang['_HELP_TEXT']`. Alternatively a URL can be used to point to a specific (local or external) HTML page.

default : ""

example : "http://support.example.org/help-for-filesender.aspx"

File and directory location settings

These settings define the directory locations where various files can be found or stored. Directory values are absolute filesystem paths on your server and should be readable and writable (except for the `site_simplesamllocation`) for the webserver user.

`$config['site_filestore']`

description: the directory where all uploaded files are stored

remarks : this directory should have ample disk space and can fysically be stored on a NAS or other big storage device.

default: `'/usr/share/filesender/files/'`

example: `'/data/filesender/files/'`

`$config['site_simplesamllocation']`

description : directory where the SimpleSAMLphp software can be found. Needed to find the required SimpleSAMLphp libraries.

default : `= '/usr/share/simplesamlphp/'`

`$config['log_location']`

description : directory to keep FileSender specific logfiles.

remarks : logfiles are created per day (and per user/voucheruid for `client_specific_logging`). FileSender has no internal mechanism to clean up old logs, you might want to do a (automated) regular cleanup of this directory.

default : `'/usr/share/filesender/log/'`

Database related settings

`$config['db_type']`

description: specifies what type of database you're using. Values can be "pgsql"// pgsql or mysql

default: 'pgsql'

`$config['db_host']`

description: the hostname of the host where the FileSender database server resides. Simplest setup is on the FileSender server

default: 'localhost'

`$config['db_database']`

description: Name of the Postgresql database that will be used to store FileSender's meta data. Files are stored directly on disk.

default: 'filesender'

\$config['db_port']

description: TCP port of the Postgresql database that File Sender will connect to

default: '5432'

\$config['db_username']

description: username used to authenticate to the Postgresql database

default: none

example: 'filesender'

\$config['db_password']

description: password to authenticate to the Postgresql database

default: none

example: 'databasepassword'

To accomodate more elaborate PDO database settings a DSN style database setting can be configured which will override the db_* settings:

```
//Optional DSN format overrides db_ settings
//$config['dsn'] = "pgsql:host=localhost;dbname=filesender";
//$config['dsn'] = 'pgsql:host=localhost;dbname=filesender';
//$config['dsn'] = 'sqlite:/usr/share/filesender/db/filesender.sqlite';
//$config['dsn_driver_options'] = array();
// dsn requires username and password in $config['db_username'] and $config['db_p
```

\$config['dsn']

description: the 'URL' pointing to the FileSender database

default: none

Cron settings

\$config['cron_exclude_prefix']

description : exclude deletion of files with the prefix character listed (can use multiple characters eg '._' will ignore .xxxx and _xxxx)

default: '._'

\$config['cron_shred']

description : Instead of simply unlinking, overwrite expired files so they are hard to recover

availability: version 1.5

default: false;

`$config['cron_shred_command']`

description: the command used to shred files

availability: version 1.5

default: `/usr/bin/shred -f -u -n 1 -z`;

`$config["cron_cleanupdays"]`

description: number of days to keep TeraSender chunk files in the temp_filestore. TeraSender chunks are sent in parallel. This means they might arrive out-of-order. They are appended to the uploaded file in-order. This means at any time a certain amount of out-of-order chunks are stored in the temp_filestore. In case an upload is cancelled for whatever reason these stale chunks need to be cleaned up. This cronjob setting indicates after how many days possibly stale chunks are removed.

values: integer

default: 7

availability: available since version 1.6

Email related configuration directives

FileSender sends out 5 types of email to users. In the configuration file you can define templates for these emails. These templates can contain template variables that will be filled when the actual email is created. This section details the configuration file directives for those email templates and the email template variables.

**** finished except for the mapping of which variables can be used with which templates ****

\$config['default_emailsubject']

description: If a FileSender user refrains from entering an email subject then this subject will be used

usable template variables: {siteName}, {fileName}

default value: "{siteName}: {fileName}";

\$config['filedownloadedemailbody']

description: email sent to the *sender* of a file when a *recipient* has downloaded that file.

usable template variables:

\$config['fileuploadedemailbody']

description: email sent to the *recipient(s)* of an uploaded file that the file is available for download.

usable template variables:

\$config['voucherissuedemailsubject']

description: default subject for the email sent to a recipient of a guest usage voucher.

usable template variables:

\$config['voucherissuedemailbody']

description: email sent to a recipient of a voucher that a voucher is available to upload a file.

usable template variables:

\$config['defaultvouchercancelled']

description: email sent to a recipient of a voucher that said voucher is now cancelled.

usable template variables:

\$config['defaultfilecancelled']

description: email sent to a recipient of a file that said file is now cancelled.

usable template variables:

Email template variables

In email templates you can use certain variables that will be expanded by FileSender when the emails are actually created. The following template variables can be used in the email subject and the email body:

{fileto}

- will be replaced with the recipient address of the mail

- usable in emails: **fileuploadedemailbody**, **filedownloadedemailbody**

{filefrom}

- will be replaced with the sender address of the mail

- usable in emails: **fileuploadedemailbody**, **filedownloadedemailbody**

{filemessage_start} and **{filemessage_end}**

- markers for begin and end of the (optional) personal message
- usable in emails: **fileuploadedemailbody**

{siteName}

- will be replaced with the site name as specified in `$config['site_name']`

- usable in emails: **filedownloadedemailbody**, **fileuploadedemailbody**, **voucherissuedemailbody**, **defaultvouchercancelled**, **defaultfilecancelled**

{filename} {fileoriginalname}

- will be replaced with the original filename.

- usable in emails: **fileuploadedemailbody**, **filedownloadedemail body**

{htmlfilename} {htmlfileoriginalname}

- usable in the HTML part of emails: **fileuploadedemailbody**, **filedownloadedemail body**
- Same as {filename} but with html encoding of unsafe characters. **Always** use this variable to specify the filename in HTML parts of a mail

{serverURL}

- will be replaced with URL as defined in `$config['site_url']`

- usable in emails:

{filevoucheruid}

- *will be replaced with the voucher ID of the file or voucher (?)*

- *usable in emails:*

{fileexpirydate}

- *will be replaced with the expiry date of the file*

- *usable in emails:*

{filesize}

- *will be replaced with the filesize of the file*

- *usable in emails:*

{CRLF}

- *used to insert a line break in emails.*

{charset}

- *will be replaced with the charset detected in the actual message*

- *used in the MIME headers of emails: **fileuploadedemailbody**, **filedownloadedemailbody***

Not implemented/unused

\$config['site_icon'] = 'cloudstor.png';

\$config['site_css'] = '';

\$config["server_drivespace_warning"]

description : the percentage of free space on the storage device at (or below) which a warning message is mailed to the administrator(s) as specified in \$config['adminEmail'].

default : 20 (= send a warning when 20% or less space is left)

\$config["client_specific_logging"]

description : turn on/off client specific logging. If set to true the client (Flash UI) will send debug information to the server which will be logged in separate user-specific files in the 'log_location' directory.

remark : turning on client specific logging will impact performance, obsolete since 1.5 (no Flash UI), setting this to *true* in 1.5+ has the same effect as setting \$config["debug"] to *true*. Setting will most likely be removed in a future version.

default : false (*true in beta releases*)

values : true/false

\$config["client_specific_logging_uids"]

description : A list of UIDs, separated by comma. When client specific logging is turned on the logging will be restricted to the UIDs in this list. The value "" (default) means 'log all clients'. A UID can be the UID of an authenticated user (as specified by the saml_uid_attribute setting below) or a voucher ID. Not used anymore since 1.5.

default : ""

example : '31a7b626eb4fa2d425f611c0490940e71e1c42c, 6B30BA68-B883-8565-36B5-73CDBE0C672C'

Comments are disabled for this space. In order to enable comments, Messages tool must be added to project.
[You can add Messages tool from Tools section on the Admin tab.](#)