Collapse All | Expand All

- Home
- Archived
- Acknowledgements
- Blog and News
- Documentation v2-0-alpha
  - Installation - Linux Source 2-0-Alpha-from-git
  - Development upgrade notes to 2-0
  - **Admin - Configuration directives v2-0**
  - Admin - Reference guide v2-0
  - RESTful API v2-0
  - Language handling and localisation reference v2-0
  - Detailed feature list v2-0
- Documentation v1-6
- Documentation v1-5
- Documentation v1-1
- Download
- Developer
- Existing installations
- Feature requests
- Online Demo
- Project organisation
- Roadmap
- Release Schedule
  - Release Schedule until v1-0
- Release status and life cycle
- Reviews and prototypes
  - Monash University User Experience Review
  - Intersect UX Review
- Support and Mailinglists
- Twitter
- FileSender's workflows

Installation - Linux Source 2-0-Alpha-from-svn

Admin - Configuration directives v2-0

Version 66, last updated by jornane at about 1 year

How to read this document

- This document is a work in progress. If you are keen on seeing FileSender 2.0 released offer to help with (testing) the documentation.
- mandatory configuration settings are marked in red
- sections marked in orange need to be double checked.

Table of contents

General settings

Backend storage

Database

Language and internationalisation

Email

- email_reply_to_name
- email_return_path
- email_use_html
- email_newline
- relay_unknown_feedbacks

General UI

- autocomplete
- autocomplete_max_pool
- autocomplete_min_characters
- upload_display_bits_per_sec

Transfers

- aup_default
- aup_enabled
- ban_extension
- chunk_upload_security
- default_days_valid
- max_days_valid
- allow_transfer_expiry_date_extension
- force_legacy_mode
- legacy_upload_progress_refresh_period
- max_legacy_file_size
- max_transfer_size
- max_transfer_files
- max_transfer_recipients
- transfer_options (email receipt control)
- upload_chunk_size
- user_quota

TeraSender (high speed upload module)

- terasender_enabled
- terasender_advanced
- terasender_worker_count
- terasender_start_mode
- stalling_detection

Download

- download_chunk_size
- mac_unzip_name
- [mac_unzip_link)(#mac_unzip_link)

Guest use

- guest_options
- default_guest_days_valid
- max_guest_days_valid
- max_guest_recipients

Authentication

- auth_sp_type
- session_cookie_path
- **SimpleSAMLphp**
  - auth_sp_saml_authentication_source
  - auth_sp_saml_simplesamlphp_url
  - auth_sp_saml_simplesamlphp_location
  - auth_sp_saml_email_attribute
  - auth_sp_saml_name_attribute
  - auth_sp_saml_uid_attribute
- **Shibboleth**

  - auth_sp_shibboleth_uid_attribute
  - auth_sp_shibboleth_email_attribute
  - auth_sp_shibboleth_name_attribute
  - auth_sp_shibboleth_login_url
  - auth_sp_shibboleth_logout_url

- **SP_Fake**
  - auth_sp_fake_authenticated!!
  - auth_sp_fake_uid!!
  - auth_sp_fake_email!!
  - auth_sp_fake_name!!

Maintenance and logging

- failed_transfer_cleanup_days
- log_facilities!!
- maintenance mode
- statlog_lifetime
- auth_sp_additional_attributes
- auth_sp_save_user_additional_attributes
- statlog_log_user_additional_attributes
- auth_sp_fake_additional_attributes_values
- auditlog_lifetime
- report_format

Webservices API

- auth_remote_application_enabled
- remote_applications
- auth_remote_user_autogenerate_secret
- rest_allow_jsonp

Other

- host_quota
- config_overrides (experimental feature, not tested)

Configuration directives

General settings

admin_email

- **description:** email address of FileSender administrator(s). Separate multiple addresses with a comma (',').
  Emails regarding disk full etc. are sent here. You should use a role-address here.
- **mandatory:** yes. There must be at least one email address defined.
- **type:** string.
- **default:** -
- **available:** since version 1.0
- **1.x name:** adminEmail

admin

- **description:** UIDs (as per the configured saml_uid_attribute) of FileSender administrators. Accounts with these UIDs can access the Admin page through the web UI. Separate multiple entries with a comma (',').
- **mandatory:** yes. Can be empty but then no-one has access to the admin page.
- **type:** string
- **default:** -
- **available:** since version 1.0

site_name

- **description:** friendly name for your FileSender instance. Used in site header in browser and in email templates.
- **mandatory:** no. If you don't define it, every place it's used will initialise to NULL which results in an empty string being displayed.
- **type:** string
- **default:** FileSender
- **available:** since version 1.0

force_ssl

- **description:** enforce use of SSL. Set this to true and FileSender won't work if the user doesn't have a SSL session. Useful to retain security in case of web server misconfigurations.
- **mandatory:** no. if you don't set it it will be evaluated to false? What about the default of 'true'?)
- **type:** boolean
- **default:** true
- **available:** since version 1.0
- **1.x name:** forceSSL

session_cookie_path

- **description:** Explicitly sets the session.cookie.path parameter for the authentication cookies. You typically need this if you use SimpleSAMLphp for authentication and have multiple FileSender instances using the same SimpleSAMLphp installation. Shibboleth has its own session identifier mechanism and you probably won't need to change the session_cookie_path when using Shibboleth.
- **mandatory:** no
- **type:** string
- **default:** if(!$session_cookie_path) $session_cookie_path = $site_url_parts['path'];
- **available:** since version 2.0
- **1.x name:**
- **comment:** Testing, ticket #1198
- **comment:** Be careful to include the entire URL path, like http://yourdomain.dom/!
- **comment:** When do you set this? If you use SimpleSAMLphp for authentication there is one common scenario where you need to set this parameter: the URL space for your FileSender instance and your SimpleSAMLphp instance do not overlap. This happens when you have multiple FileSender instances (one production, one beta) sharing the same SimpleSAMLphp installation. For example: http://yourdomain.dom/filesender-beta and http://yourdomain.dom /simplesamlphp. Because SimpleSAMLphp and FileSender are both written in PHP they use the same mechanism for session identifiers. They can share session identifiers but only if this is allowed by the session_cookie_path. When you log on with SimpleSAMLphp a session identifier is created. If this can not be shared with your FileSender instance you will notice a user can log on, only to be presented with the same logon form again. A silent failure. In this scenario you will either need to ensure your SimpleSAMLphp instance is available within the FileSender URL space, or you set the session cookie parameter to for example "http://yourdomain.dom/". Another workaround is to use memcache for SimpleSAMLphp's session identifiers but that would mean an extra package on your server.

auth_remote_signature_algorithm

- **description:** which remote signature algorithm to use. Used in API? Should be in API section probably? Which other permissible values?
- **mandatory:** no
- **type:** string, permissible values: "sha1".
- **default:** "sha1"
- **available:** since version 2.0

default_timezone

- **description:** used to set default timezone of PHP. Used to convert dates. Dates are loaded from database and converted to PHP timestamps on the fly. Times in database are stored in GMT dates. Used to present localised time information. Audit logs use time? Also: include link to PHP timezone values
- **mandatory:** yes (doublecheck)
- **type:** string
- **default:** Europe/London
- **available:** since version 1.0
- **1.x name:** Default_TimeZone

default_language

- **description:** if there are no end-user overrides then this is the default language to use in the UI (and email?). If the user picks a language that doesn't exist, or if a language directive isn't translated in the language served up to the user, the directive will in stead be taken from the language defined here as default_language. If all else fails, English (en) is a hard coded default.
- **mandatory:** no. Hard-coded default of last resort: English ("en")
- **type:** string
- **default:** en
- **available:** since version 1.6
- **1.x name:** site_defaultlanguage
- **comment:** if the default_language is not one of the available (configured) languages, the configuration validator will thrown an error.

site_url

- **description:** Site URL. Used in emails, to build URLs for logging in, logging out, build URL for upload endpoint for web workers, to include scripts etc.
- **mandatory:** yes
- **type:** string
- **default:** -
- **available:** since version 1.0

site_logouturl

- **description:** $_GET parameters for the logout page; this is where user gets redirected to after logout. Is given to the SP logout end-point.
- **mandatory:** ?
- **type:** string
- **default:** $config['site_url'].'?s=logout'
- **available:** since version 1.6

about_url

- **description:** if set to "", a modal inline popup dialogue is shown with the contents of _ABOUT_TEXT from the relevant language file. Alternatively a URL can be used to point to a specific (local or external) HTML page. check if this is a modal inline popup
- **mandatory:** no
- **type:** string
- **default:** not set = empty string = popup
- **available:** since version 1.0
- **1.x name:** aboutURL

help_url

- **description:** if set to "", a modal inline popup dialogue is shown with the contents of _HELP_TEXT from the relevant language file. If set to an URL it will open the referenced (local or external) HTML page in a new tab. check if this is a modal inline popup
- **mandatory:** no doublecheck
- **type:** string
- **default:** not set = empty string = popup
- **available:** since version 1.0
- **1.x name:** helpURL
- **comment:** when configured with a mailto: address that points to e.g. support@yourdomain.dom the email bounce handler will use this address to send unprocessable email bounces to. include link to email bounce handling config directives/help

---

Backend storage

---

storage_type

- **description:** type of storage you used for storing files uploaded to FileSender.
- **mandatory:** no
- **type:** string. Permissible values: **filesystem**.
- **default:** filesystem
- **available:** since version 2.0
- **comment:** each supported storage type will have a specific class defined in classes/storage. Each is named Storage.class.php, for example StorageFilesystem.class.php for the type

filesystem. The values for "Foo" are the permissible values for this directive. For now the only permissible value and supported storage type is filesystem. Future storage types could include e.g. **object**, **amazon_s3** and others.

storage_filesystem_path

- **description:** when using storage type **filesystem** this is the absolute path to the file system where uploaded files are stored until they expire. Your FileSender storage root.
- **mandatory:** no
- **type:** string
- **default:** ['filesenderbase'].'/files'
- **available:** since version 1.0
- **1.x name:** site_filestore
- **comment:**

storage_filesystem_df_command

- **description:** Command used to determine available disk space on file system. Used to perform per-transfer check for sufficient disk space and to trigger disk space usage warnings to the FileSender Admin
- **mandatory:** ?
- **type:** string
- **default:** ??
- **available:** since version 2.0
- **1.x name:**
- **comment:**

storage_filesystem_file_deletion_command

- **description:** Command used to delete files, when they expire or are cleaned in routine cleaning of stale files.
- **mandatory:** ?
- **type:** string
- **default:** ?
- **available:** since version 1.1
- **1.x name:** cron_shred_command
- **comment:**

storage_usage_warning

- **description:** percentage of drive space left that will trigger an email warning to the admin.
- **mandatory:** no. If not set, evaluates to zero and you get no warnings.
- **type:** int
- **default:** 20
- **available:** since version 1.0
- **1.x name:** server_drivespace_warning
- **comment:**

###storage_filesystem_hashing * **description:** Aggregate several directories into a virtual FileSender file store without using LVM. Directories can be on different file systems which can be on different block devices and hard drives. Allows you to pool several hard drives into one virtual FileSender file store without any external software.
* **mandatory:** no * **type:** **int** or **callable**. When integer indicates number of characters used in hash. When callable "file que l'on veit stocker et doit retourner le chemin dans le stockage" * **default:** 0 * **available:** since version 20 * **1.x name:** * **comment:** not tested * **comment:** basically integer. use fileUID (which is used to create name on hard drive) + as many characters as the hashing value (if you set hashing to 2 you take the 2 first letters of the fileUID (big random string) and use these two characters to create a directory structure under the storage path. This avoids having all files in the same directory. If you set this to 1 you have 16 possible different values for the directory structure under the storage root. You'll have 16 folders under your storage root under which you'll have the files. This allows you to spread files over different file systems / hard drives. You can aggregate storage space without using things like LVM. If you set this to two you have 2 levels of subdirectories. For directory naming: first level, directory names has one letter. Second level has two: letter from upper level + own level. Temporary chunks are stored directly in the final file. No temp folder (!!) Benchmarking between writing small file in potentially huge directory and opening big file and seeking in it was negligable. Can just open final file, seek to location of chunk offset and write data. Removes need to move file in the end. It can also be "callable". We call the function giving it the file object which hold all properties of the file. Reference to the transfer as well. The function has to return a path under the storage root. This is a path related to storage root. For example: if you want to store small files in a small file directory and big files in big directory. F.ex. if file->size < 100 MB store on fast small disk, if > 100 MB store on big slow disk. Can also be used for functions to store new files on new storage while the existing files remain on existing storage. Note: we need contributions for useful functions here :)

Database

db_type

- **description:** type of database
- **mandatory:** yes
- **type:** string, keyword
- **permissible values**: mysql, pgsql, sqlite (taken from PDO drivers documentation, need to check with Etienne)
- **default:** pgsql
- **available:** since version 1.0
- **1.x name:**
- **comment:**

db_host

- **description:** database host address or name. Typically 127.0.0.1 or localhost.
- **mandatory:** yes
- **type:** string
- **default:** -
- **available:** since version 1.0
- **1.x name:**
- **comment:**

db_port

- **description:** port used by database server
- **mandatory:** yes
- **type:** string
- **default:** -
- **available:** since version 1.0
- **1.x name:**
- **comment:**

db_username

- **description:** database username
- **mandatory:** yes
- **type:** string
- **default:** -
- **available:** since version 1.0
- **1.x name:**
- **comment:**

db_password

- **description:** database password
- **mandatory:** yes
- **type:** string
- **default:** -
- **available:** since version 1.0
- **1.x name:**
- **comment:**

db_table_prefix

- **description:** table prefix to use. Allows you to have several filesender instances in one database. For example if you buy hosting with 1 database and still want multiple filesender instances.
- **mandatory:** ? Would think not?
- **type:** string
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:**

Language and internationalisation

FileSender includes a translation engine which allows for flexible user language detection and customisation. For more details check the Translating FileSender 2.0 documentation

User language detection is done in the following order: 1. From the url (`lang` url parameter) : allows for user language switching (only if `lang_url_enabled` set to true in config), if `lang_save_url_switch_in_userpref` is enabled in config and a user session exists the new language is saved in the user preferences so that he doesn't need to switch it again the nex time. If no user session is found the new choice is saved in the PhP session. 2. From the browser's `Accept-Language` header : allows for automatic language detection base on the browser config (if `lang_browser_enabled` set to true in config) 3. From `default_language` config parameter 4. From the hard-coded absolute default `en`

lang_browser_enabled](#lang_browser_enabled)

- **description:** detect user's preferred language from browser's Accept-Language header if this header is provided. If a language a user requests is not available, falls back to the default language. If no default language is configured, falls back to English. If a language directive is not available in the selected language, it is taken from the default language file.
- **mandatory:** no
- **type:** boolean
- **default:** true
- **available:** since version 2.0
- **1.x name:**
- **comment:** requires lang_url_enabled to be true.

lang_url_enabled

- **description:** allow explicit language switching via URL (example: ?lang=en)
- **mandatory:** no (required when using lang_browser_enabled)
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:**

lang_userpref_enabled

- **description:** take user's preferred language from user's stored preferences. These preferences are stored in the FileSender database.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:**

lang_selector_enabled

- **description:** display language selector in UI . If your FileSender instance only supports 1 language no selector is displayed and no "translate this email" link is present in emails.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:** requires lang_url_enabled to be true.
- **comment:** if the lang_selector is disabled a user can still select different translations in the email translation page
- **comment:** how is determined which language the lang selector defaults to when a user enters a page? Browser setting? Order in locale.php?

lang_save_url_switch_in_userpref

- **description:** save language switching in user preferences on change (requires lang_url_enabled = true and lang_userpref_enabled = true)
- **mandatory:** no

- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:**

Email

email_from

- **description:** sets the email From: header to either an explicit value or fills it with the sender's email address as received from the identity service provider in the "mail" attribute. Is this the body From:?
- **mandatory:** no
- **type:** string or keyword. Permissible value for keyword: "sender"
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:** To be SPF compliant set this to an address like "filesender-bounces@yourdomain.dom" and use the bounce-handler script to deal with email bounces.

email_from_name

- **description:** pretty name for the email_from address. Use when you explicitly set email_from to an email address like "no-reply@domain.dom".
- **mandatory:** no
- **type:** string
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:**

email_reply_to

- **description:** adds a reply-to: header to emails sent by FileSender. When users reply to such an email usually the reply is then sent to the reply_to address. A user would typically reply to an email to ask a question about a file transfer which should go directly to the sender as the sender is the only one who knows.
- **mandatory:** no
- **type:** string or keyword. Permissible values for keyword: "sender"
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:** To be SPF compliant set this to "sender"

email_reply_to_name

- **description:** pretty name for the email_reply_to address. Use when you explicitly set email_reply_to to an email address like "no-reply@domain.dom".
- **mandatory:** no
- **type:** string
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:**

email_return_path

- **description:** sets the return_path email header to either an explicit value or fills it with the sender's email address as received from the identity service provider in the "mail" attribute. Is this the envelope from??
- **mandatory:** no
- **type:** string or keyword. Permissible value for keyword: "sender"
- **default:** -
- **available:** since version 2.0
- **1.x name:**

- **comment:** To be SPF compliant set this to an address like "filesender-bounces@yourdomain.dom" and use the bounce-handler script to deal with email bounces.

email_subject_prefix

- **description:** the string specified here will be prepended to the subject of all emails sent out.
- **mandatory:** no
- **type:**: string
- **default:** site_name config directive
- **available:** since version 2.0
- **1.x name:**
- **comment:** was equal to site_name in version 1.x

email_use_html

- **description:** if true all emails sent by FileSender will include both HTML and plaintext. For most users this means they will see HTML emails. If false only plain-text emails are sent.
- **mandatory:** no
- **type:** boolean
- **default:** true
- **available:** since version 2.0 (?)
- **1.x name:**
- **comment:**

email_newline

- **description:** specify a different new line character for use in emails. If your FileSender emails look garbled (display raw MIME and HTML source) try setting this to \n as an alternative to reconfiguring your mail server.
- **mandatory:** no
- **type:** string
- **default:** "\r\n" (as per RFC 2822)
- **available:** since version 1.0
- **1.x name:** crlf
- **comment:** the default value in version 1.x was "\n".
- **comment:** Make sure you use double quotes to configure this value in the config file. If you use single quotes the \r and \n will NOT be interpreted!

relay_unknown_feedbacks

- **description:** tells the bounce handler where to forward those messages it can not identify as email bounces but can be related to a specific target (recipient, guest). The received message is forwarded as message/rfc822 attachment.
- **mandatory:** no
- **type:** string or keyword
- **permissible values:** "sender": relay to recipient's transfer owner or guest owner. "admin": relay to admin email address. "support": relay to help_url if the latter is in the form of a mailto: URL ("mailto:someaddress@domain.tld"), "someaddress@domain.tld": an explicit email address to forward these types of mails to.
- **default:** "sender"
- **available:** since version 2.0
- **1.x name:**
- **comment:** this parameter will get a different name

General UI

autocomplete

- **description:** provide autocomplete for email input fields. If set to a positive integer autocomplete is enabled and the value dictates how many results are returned to a user in the autocomplete popup. The result list is limited to recipients this particular user has used.
- **mandatory:** no
- **type:** integer/boolean
- **default:** false
- **available:** since version 1.6
- **1.x name:**
- **comment:** Checks the frequent recipient field (array) in the user preference table. Holds different recipients the user did use. The first one is the last used one. Every time the user sends a file or guest voucher we take recipiients and add them at the top of the array. If they

already exist in the array the address is put to the top. We limit array to max length defined in config.

autocomplete_max_pool

- **description:** how many of the user's recipients are stored in the user's preferences in the database. Should be between 2 and ca. 15 times the "autocomplete" value.
- **mandatory:** no
- **type:** int
- **default:** 5 times the value set for autocomplete
- **available:** since version 1.6
- **1.x name:** autocompleteHistoryMax
- **comment:** the higher this number the larger the number of email recipients you will store over time. This increases your privacy footprint.

autocomplete_min_characters

- **description:** how many characters the user needs to type in an email address field to trigger the autocomplete popup.
- **mandatory:** no
- **type:** int
- **default:** 3 (might 2 be better?)
- **available:** since version 2.0
- **1.x name:**
- **comment:**

upload_display_bits_per_sec

- **description:** if true display upload speed in MBps (megabytes/second). If false display upload speeds as Mbps (megabits/second) Need to test, reality seems different from documentation
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:** does this actually work?

Transfers

---

aup_enabled

- **description:** If set to 'true' the AuP (terms of service) checkbox is visible AND mandatory for the user to tick.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 1.0
- **1.x name:** AuP
- **comment:**

aup_default

- **description:** if set to 'true' the AuP (terms of service) checkbox (if enabled) is already pre-ticked for the user
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 1.0
- **1.x name:** AuP_default
- **comment:**

ban_extension

- **description:** disallow files with the extensions specified here.
- **mandatory:** no
- **type:** string

- **default:** exe, bat
- **available:** since version 1.0
- **1.x name:**
- **comment:**

chunk_upload_security

- this entire parameter needs to be checked with Etienne
- **description:** controls how FileSender behaves when an upload lasts longer than an authenticated user session. If set to "key" the web client will use FileUID as a transfer session key. This transfer session key is valid for as long as the upload lasts independent from the user's login session. So if a user logs in at some identity provider and that session expires after e.g. 8 hours but the upload lasts for 10 hours, the upload will complete. If set to "auth" the user will be required to re-logon if their logon session has expired before an upload completed. **If set to "key" and the user's login session expires before the upload is completed, the user will need to be logged on before redirected to their "My Transfers" page.**
- **mandatory:** no
- **type:** string keyword
- **permissible values:** "key" or "auth"
- **default:** key
- **available:** since version 2.0
- **1.x name:**
- **comment:** When you upload a file it is uploaded in chunks. For each chunk there is a check whether the chunk belongs to a valid session and it ensures the right chunks are appended to the right files. You don't want others to be able to insert chunks in a user's file as it would lead to file corruption. In version 1.x this check was done with the user's session (login) identifier which from a security point of view worked well. The only problem is that sometimes uploads take a long time, depending on file size and upload speeds. A user's login session can then expire before the upload is complete. Most FileSender installations in national research networks use SAML-based authentication. A user logs in to an Identity Provider (IdP), this IdP sends a SAML-token to the Service Provider (SP, your FileSender instance) containing information like the session authentication token. This SAML-token also contains a timestamp which indicates exactly when the user's login session expires. FileSender CAN NOT change this session expiry time as the authentication libraries it uses honour this value. It's the Identiy Provider that makes this choice. In Norwegian higher education for example a login session with the national authentication infrastructure expires after 8 hours. We have seen uploads that last longer than that.

  To solve this we introduced a transfer key in FileSender 2.0. When you start an upload you use the FileUID as a unique transfer session key. If the user session times out before the upload is done, the upload will still continue. The transfer session key expires immediately once the upload is done. The upload is secure: you need an authenticated session to start an upload, only the server and the uploading client have knowledge of the FileUID. Third parties can not inject chunks.

  If you want to find out the expiry timer for your SAML Identity Provider install the SAML tracer add-on in FireFox and log in to your FileSender install. Click on the "SAML" message in SAML tracer.

default_transfer_days_valid

- **description:** specifies the default expiry date value in the "Expiry date" date picker in the Upload form. If a user doesn't do anything this becomes the expiry date for the transfer.
- **mandatory:** no
- **type:** int
- **default:** 10
- **available:** since version 1.0
- **1.x name:** default_daysvalid
- **comment:** Be aware of the changed semantic from 1.6 to 2.0.

max_transfer_days_valid

- **description:** specifies the maximum expiry date for a transfer. A user can not choose a larger value than this.
- **mandatory:** no
- **type:** int
- **default:** 20
- **available:** since version 1.0
- **1.x name:** default_daysvalid
- **comment:** experience shows the vast majority of users simply go with the default expiry time. For some users having a maximum value a long time in the future makes sense, e.g. papers sent out to a research proposal evaluation committee that need to be evaluated by a certain date. Downloads typically start not too long before the due date, but the actual due date can be over a month in the future.

allow_transfer_expiry_date_extension

- **description:** allows a user to extend the expiry date.

- **mandatory:**
- **type:** an array of integers containing possible extensions in days.
- **default:** - (= not activated)
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- **Examples:**

```
    $config['allow_transfer_expiry_date_extension'] = array(5); // Allows a sing
    $config['allow_transfer_expiry_date_extension'] = 5; // Same as above
    $config['allow_transfer_expiry_date_extension'] = array(5, 3); // Allows 2 s
    $config['allow_transfer_expiry_date_extension'] = array(5, 3, 1, true); // A
```

force_legacy_mode

- **description:** Force FileSender into legacy non-HTML5 mode. Multi-file uploads are still possible, but each file is limited to max. 2GB. The help file and certain text labels change as well. The max. number of files and total transfer size limit is the same as for HTML5 mode. This function is available for testing purposes: FileSender will detect automatically if a user's browser supports the necessary HTML5 functionality or not.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:** for testing purposes.

legacy_upload_progress_refresh_period

- **description:** when uploading in legacy mode (non-HTML5 uploads) this indicates in seconds how often the client-side progress bar is refreshed.
- **mandatory:** no
- **type:** int (seconds)
- **default:** 5. Setting this to 0 is not a wise choice as it will make the timer refresh every millisecond (the min. value for a JavaScript timer)
- **available:** since version 2.0
- **1.x name:**
- **comment:** Normally FileSender will use the browser's HTML5 FileAPI functionality for uploading, splitting files in chunks and uploading these chunks. This allows for uploads of any size. Older browsers which you may find in a locked-down environment do not support the necessary HTML5 functionality. For these browsers a legacy fallback upload method is provided. Before version 2.0 a flash component was used for legacy uploads. As of version 2.0 this is replaced by a native HTML upload with a limit of 2GB per file. A user **can** select multiple files but in a less smooth way than with the HTML5 drag & drop box. The upload progress for legacy uploads is polled from the server (via PHP) based on what has arrived (how many bytes) server side. This only became possible as of PHP version 5.x, released in x

max_legacy_file_size

- **description:** maximum size per file for a legacy upload. With a legacy upload users can upload x files per transfer..
- **mandatory:** no
- **type:** int
- **default:** 2147483648 (2GB)
- **available:** since version 1.0
- **1.x name:** max_flash_upload_size
- **comment:** Files are uploaded serially. A hidden iframe and hidden form is created for each file, containing the required data (session key for upload etc.). A single file element is cloned into each hidden form. This form is submitted to the hidden iframe which then uploads the file. At the end of the upload the server sends a bit of javascript which triggers the next upload in the queue. Each file is an "entire file at once" upload rather then the chunked upload used to get over the 2GB limit of 32 bit browsers.

max_transfer_size

- **description:** maximum total size for any transfer (both html5 and legacy transfers)
- **mandatory:** no
- **type:** int

- **default:** 107374182400 (100 GB)
- **available:** since version 1.0
- **1.x name:** max_html5_upload_size
- **comment:**

max_transfer_files

- **description:** maximum number of files that can be sent in one transfer
- **mandatory:** no
- **type:** int
- **default:** 30
- **available:** since version 2.0
- **1.x name:**
- **comment:**

max_transfer_recipients

- **description:** maximum number of recipients a transfer can have.
- **mandatory:** no
- **type:** int
- **default:** 50
- **available:** since version 1.0
- **1.x name:** max_email_recipients
- **comment:**

transfer_options

- **description:** this parameter controls which transfer options are available to the user in the Upload form and how these options behave. Options show up in the right hand side block in the Upload form. Options appear in the order they are specified in the config file. Most options control which email receipts are sent out when and to whom. See below for details.
- **mandatory:** no
- **type:** array
- **default:**
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- ***Standard parameters for all options:***
  - **available**(boolean): if set to true then this option shown in the upload form
  - **advanced** (boolean): if set to true the option is hidden under an "Advanced options" click-out. The user must click "Advanced" to make the option visible.
  - **default** (boolean): if set to true then this option is ticked by default. If set to true while ***available*** is set to false the option is mandatory for all users, it can not be switched off by the user.

- ***Available options:***

  - **email_me_copies:** the sender receives copies (Cc:) of all emails concerning this transfer. This is the "spam-me-plenty" option.
  - **email_me_on_expire:** the sender receives a message when the transfer expires.
  - **email_upload_complete:** send the sender an email once the sender's upload is finished. This allows a sender to start a long upload on a workstation when leaving work and check with a smartphone whether the upload was completed some hours afterwards.
  - **email_download_complete:** notify the sender (owner) of a transfer that someone has downloaded it immediately after the download completes.
  - **email_daily_statistics:** send the sender an overview of all activity on that sender's transfers. Who downloaded what when.
  - **email_report_on_closing:** send the sender an overview of all activity on this particular transfer after that transfer is closed. This is the audit report for that particular transfer. When a sender receives this, the server's audit logs can (in principle) be purged for the records pertaining to this particular transfer thus reducing FileSender's privacy footprint.
  - **enable_recipient_email_download_complete:** this gives the downloader a tick box in the download window which in turn lets the downloader indicate they would like to receive an email once the download is finished. If you want this option available for all downloaders and do not want to bother the uploader with it, simply configure it with 'default' => false as the only parameter. **Warning:** if the recipient of a file is a mailinglist and someone ticks the "send me a message on download complete" box, then all members of that mailinglist will receive that message. That might be a reason why you don't want to make this option

available to your users.
  - **add_me_to_recipients:** include the sender as one of the recipients.
  - **get_a_link:** if checked it will not send any emails, only present the uploader with a download link once the upload is complete. This is useful when sending files to mailinglists, newsletters etc. When ticked the message subject and message text box disappear from the UI. Under the hood it creates an anonymous recipient with a token for download. You can se the download count, but not who downloaded it (obviously, as there are no recipients defined).
  - **redirect_url_on_complete:** When the transfer upload completes, instead of showing a success message, redirect the user to a URL. This interferes with **get_a_link** in that the uploader will not see the link after the upload completes. Additionally, if the uploader is a guest, there is no way straightforward way for the uploader to learn the download link, although this must not be used as a security feature.

- ***Configuration example:***

```
    $config['transfer_options'] = array(
        'email_upload_complete' => array(
            'available' => true,
            'advanced' => false,
            'default' => false
        ),
        'email_me_copies' => array(
            'available' => true,
            'advanced' => true,
            'default' => false
        );
```

upload_chunk_size

- **description:** standard upload for FileSender is chunked upload. This indicates how big each chunk is. There is a certain optimal chunk size which depends largely on your bandwidth-delay product. Usually you shouldn't have to touch this but if you're trying to serve special use cases you might want to experiment with this and see which value gives you the fastest upload times..
- **mandatory:** no
- **type:** int (bytes)
- **default:** 5 * 1024 * 1024 (5242880 bytes (5MB))
- **available:** since version 1.5
- **1.x name:**
- **comment:** Please note that as of version 2.0 the terasender_chunksize and upload_chunk-size have been merged into one parameter.

user_quota

- **description:** set to 0 to disable. If set to a positive value it sets the per-user maximum storage usage. A transfer requiring more space than remains in the user's quota are rejected with an error message in the web-UI.
- **mandatory:** no
- **type:** int (bytes) or function
- **default:** 0
- **available:** since version 2.0
- **1.x name:**
- **comment:** user quote can be implemented in a much more flexible way as well. As we're doing lazy loading of configuration parameters we can change this value (and max. file size) based on user profile. In stead of defining this config parameter with a number you can give a function to it. The value returned by this function is cached for a login session. For example a function that uses eduPersonAffiliation can give a "student" 10 GB and "faculty" 1 TB. You could also change max. days valid based on user profile. The function can use the current application state and user session to compute the value for a logged in user, because the function would run after everything else. Calculated maximum values should have its own chapter to explain, with examples especially for using eduPersonAffiliation.

TeraSender (high speed upload module)

---

terasender_enabled

- **description:** if set to true, enables TeraSender high speed upload module. This leverages client-side webworkers to parallelise uploads; each chunk is sent by a webworker allowing us to

send many chunks in parallel.
- **mandatory:** no
- **type:** boolean
- **default:** true
- **available:** since version 1.6
- **1.x name:** terasender
- **comment:** the default value in version 1.6 was false

terasender_advanced

- **description:** if set to yes the advanced terasender settings (worker count, chunk size) become available for a user in the UI. Use this to easily test which workercount vs. chunk size settings work best for a very specific very demanding user/use case.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 1.6
- **1.x name:** terasenderadvanced
- **comment:**

terasender_worker_count

- **description:** how many client-side workers FileSender fires up when starting a terasender upload. Note that different browsers have different maximum webworker settings which also change over time. As CPU power increases your users will typically be able to support higher number.
- **mandatory:** no
- **type:** int
- **default:** 6
- **available:** since version 1.6
- **1.x name:** terasender_workerCount
- **comment:** we need to check maximum webworker counts for standard browsers and possibly increase the default number

terasender_start_mode

- **description:** progress sequentially or parallel through the file list.
- **mandatory:** no
- **type:** string, keyword
- **permissible values:** "single" or "multiple". When single all workers will work on one single file and move sequentially through the file list. When set to multiple all workers will be spread over all files. The difference is in user experience; in the latter case a user sees progress on all files at once. In reality the total upload time should remain the same. So question is do you want the status to light up light a christmas tree or not.
- **default:** multiple
- **available:** since version 2.0
- **1.x name:**
- **comment:** when looking for a file to put a worker on in multiple mode we look at file which has compbination of least worker and least progress. Try to put available worker on file that is the slowest. In multiple-mode we try to make all files progress at about the same speed. when set to "single" uploads don't work? Bug?

stalling_detection

- **description:** detect whether an upload stalls
- **mandatory:** no
- **type:** boolean
- **default:** true
- **available:** since version 2.0
- **comment:** Has effect on the JavaScript-variables given to the client-side of Terasender.

Download

---

download_chunk_size

- **description:** the maximum amount of data that will be read into (server or client side?) memory

at once during multi-file downloads (not single file?)
- **mandatory:** ?
- **type:** int
- **default:** 5242880 (5MB)
- **available:** since version 2.0
- **1.x name:**
- **comment:**

mac_unzip_name

- **description:** per oktober 2014 the default Mac built-in unzip client is still 32 bits. This leads to problems if the zip file that's downloaded when downloading multiple-files-as-an-archive is larger than 2 GB: a user can click on the zip file but it won't expand into a folder. To prevent help desk calls we alert a user to this problem and give them a place where they can go for the solution. (need double check for Yosemite)
- **mandatory:** ? Should be?)
- **type:** string
- **default:** The Unarchiver
- **available:** since version 2.0
- **1.x name:**
- **comment:**

mac_unzip_link

- **description:** link in download form where user can download a 64 bit unzip utility for Mac OS-X
- **mandatory:** ?
- **type:** string
- **default:** http://unarchiver.c3.cx/unarchiver
- **available:** since version 2.0
- **1.x name:**
- **comment:**

Guest use

---

guest_options

- **description:** are transfer options for guest invitations inherited from transfer_options?this parameter controls which options a user has available in the Guest form to control the behaviour of guest invitations. Options show up in the right hand side block in the Guest form. Options appear in the order they are specified in the config file. See below for details.
- **mandatory:** no
- **type:** array
- **default:**
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- ***Standard parameters for all options:***
  - **available**(boolean): if set to true then this option shown in the Guest form
  - **advanced** (boolean): if set to true the option is hidden under an "Advanced options" click-out. The user must click "Advanced" to make the option visible.
  - **default** (boolean): if set to true then this option is ticked by default. If set to true while ***available*** is set to false the option is mandatory for all users, it can not be switched off by the user.

- ***Available options:***

  - **email_upload_started:** send the guest invitation owner an email when the guest upload is complete.
  - **email_upload_page_access:** send the guest invitation owner an email when the guest accesses the upload page.
  - **valid_only_one_time:** the guest invitation can be used for one transfer only.
  - **does_not_expire:** the guest invitation can be used until it is explicitly expired by the owner. Combine with can_only_send_to_me to create a permanent file upload link that can be put in an email signature.
  - **can_only_send_to_me:** the recipient for this guest invitation is fixed, the guest can not

  choose their own recipients.
  - **email_guest_created:** send the guest invitation owner an email when the guest voucher is created.
  - **email_guest_created_receipt:** send the guest an email when the guest voucher is created.
  - **email_guest_expired:** send the guest an email when the guest voucher is expired.

- ***Configuration example:***

```
    $config['guest_options'] = array(
        'email_upload_started' => array(
            'available' => true,
            'advanced' => false,
            'default' => false
        ),
        'email_upload_page_access' => array(
            'available' => true,
            'advanced' => true,
            'default' => false
        );
```

default_guest_days_valid

- **description:** specifies the default expiry date value in the "Expiry date" date picker in the Guest form. If a user doesn't do anything this becomes the expiry date for the guest invitation. If this value is not configured, it is set to default_transfer_days_valid
- **mandatory:** no
- **type:** int
- **default:** same as default_transfer_days_valid
- **available:** since version 2.0
- **1.x name:**
- **comment:**

max_guest_days_valid

- **description:** specifies the maximum expiry date for a guest invitation. A user can not choose a larger value than this.
- **mandatory:** no
- **type:** int
- **default:** same as max_transfer_days_valid
- **available:** since version 2.0
- **1.x name:**
- **comment:**

max_guest_recipients

- **description:** specifies how many recipients a guest can specify
- **mandatory:** no
- **type:** int
- **default:** 50
- **available:** since version 2.0
- **1.x name:**
- **comment:**

Authentication

---

auth_sp_type

- **description:** which authentication library to use. saml=SimpleSAMLphp, shibboleth=shibboleth, fake uses a local file. Do not use the fakesp in production!
- **mandatory:** no
- **type:** string, keyword
- **permissible values:** "saml", "shibboleth", "fake"
- **default:** saml
- **available:** since version 2.0
- **1.x name:**

- **comment:** to use type "fake" you need ...

session_cookie_path

- **description:** Explicitly sets the session.cookie.path parameter for the authentication cookies. You typically need this if you use SimpleSAMLphp for authentication and have multiple FileSender instances using the same SimpleSAMLphp installation. Shibboleth has its own session identifier mechanism and you probably won't need to change the session_cookie_path when using Shibboleth.
- **mandatory:** no
- **type:** string
- **default:** if(!$session_cookie_path) $session_cookie_path = $site_url_parts['path'];
- **available:** since version 2.0
- **1.x name:**
- **comment:** When do you set this? If you use SimpleSAMLphp for authentication there is one common scenario where you need to set this parameter: the URL space for your FileSender instance and your SimpleSAMLphp instance do not overlap. This happens when you have multiple FileSender instances (one production, one beta) sharing the same SimpleSAMLphp installation. For example: http://yourdomain.dom/filesender-beta and http://yourdomain.dom /simplesamlphp. Because SimpleSAMLphp and FileSender are both written in PHP they use the same mechanism for session identifiers. They can share session identifiers but only if this is allowed by the session_cookie_path. When you log on with SimpleSAMLphp a session identifier is created. If this can not be shared with your FileSender instance you will notice a user can log on, only to be presented with the same logon form again. A silent failure. In this scenario you will either need to ensure your SimpleSAMLphp instance is available within the FileSender URL space, or you set the session cookie parameter to for example "http://yourdomain.dom/". Another workaround is to use memcache for SimpleSAMLphp's session identifiers but that would mean an extra package on your server.

auth_sp_set_idp_as_user_organization

- **description:** saml_sp_idp (simplesaml), shib: (shib_identity_provider environment variable) takes sp identifier from sp if provided and save it in user preferences as organisation property.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:** is this still in use? There is no code associated with it as far as I can tell

Authentication: SimpleSAMLphp

auth_sp_saml_authentication_source

- **description:** which authentication source service provider to use. In SimpleSAMLphp you configure these in the configuration file /config/authsources.php.
- **mandatory:** no
- **type:** string
- **default:** default-sp
- **available:** since version 1.0
- **1.x name:** site_authenticationSource
- **comment:**

auth_sp_saml_simplesamlphp_url

- **description:** which URL to find SimpleSAMLphp.
- **mandatory:** yes, if auth_sp_type is set to 'saml'
- **type:** string
- **default:** -
- **available:** since version 1.0
- **1.x name:** site_simplesamlurl
- **comment:** You will usually have something like 'http:///simplesaml' here where 'simplesaml' is an alias defined as "Alias /simplesaml /usr/local/filesender/simplesaml/www" in your web server config.

auth_sp_saml_simplesamlphp_location

- **description:** file system path to SimpleSAMLphp location
- **mandatory:** yes, if auth_sp_type is set to 'saml'

- **type:** string
- **default:** -
- **available:** since version 1.0
- **1.x name:** site_simplesamllocation
- **comment:**

auth_sp_saml_uid_attribute

- **description:** attribute for user's unique user identifier to get from authentication service provider. Usually you would use either *eduPersonTargetedID* or *eduPersonPrincipalName* (watch the spelling!). ePTID is an anonymous identifier making it hard to link FileSender logging to a specific user which may or may not be what you want. ePTID will protect your users against rogue IdPs. eduPersonPrincipalName will usually give you an identifier like @.
- **mandatory:** no explicit configuration is needed when the default is used. However, this value MUST be received from the Identity Provider, otherwise a user can not log on.
- **type:** string
- **default:** eduPersonTargetedId
- **available:** since version 1.0
- **1.x name:** saml_uid_attribute
- **comment:**

auth_sp_saml_email_attribute

- **description:** attribute for user's mail address to get from authentication service provider
- **mandatory:** no explicit configuration is needed when the default is used. However, this value MUST be received from the Identity Provider, otherwise a user can not log on.
- **type:** string
- **default:** mail
- **available:** since version 1.0
- **1.x name:** saml_email_attribute
- **comment:**

auth_sp_saml_name_attribute

- **description:** attribute for user's name to get from authentication service provider
- **mandatory:** no
- **type:** string
- **default:** cn
- **available:** since version 1.0
- **1.x name:** saml_name_attribute
- **comment:**

Authentication: Shibboleth

---

auth_sp_shibboleth_uid_attribute

- **description:** attribute for user's unique user identifier to get from authentication service provider. Usually you would use either *eduPersonTargetedID* or *eduPersonPrincipalName* (watch the spelling!). ePTID is an anonymous identifier making it hard to link FileSender logging to a specific user which may or may not be what you want. ePTID will protect your users against rogue IdPs. eduPersonPrincipalName will usually give you an identifier like @.
- **mandatory:** no explicit configuration is needed when the default is used. However, this value MUST be received from the Identity Provider, otherwise a user can not log on.
- **type:** string
- **default:**
- **available:** since version 2.0
- **1.x name:**
- **comment:**

auth_sp_shibboleth_email_attribute

- **description:** attribute for user's mail address to get from authentication service provider
- **mandatory:** no explicit configuration is needed when the default is used. However, this value MUST be received from the Identity Provider, otherwise a user can not log on.
- **type:** string
- **default:** mail
- **available:** since version 2.0
- **1.x name:**

- **comment:**

auth_sp_shibboleth_name_attribute

- **description:** attribute for user's name to get from authentication service provider
- **mandatory:** no
- **type:** string
- **default:** cn
- **available:** since version 2.0
- **1.x name:**
- **comment:**

auth_sp_shibboleth_login_url

- **description:** where to find the Shibboleth login URL
- **mandatory:** yes when using Shibboleth as authentication library
- **type:** string
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- **example:** $prot.$_SERVER['SERVER_NAME'].'/Shibboleth.sso/Login?target={target}';

auth_sp_shibboleth_logout_url

- **description:** where to find the Shibboleth logout URL
- **mandatory:** yes when using Shibboleth as authentication library
- **type:** string
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- **example:** $prot.$_SERVER['SERVER_NAME'].'/Shibboleth.sso/Logout?return={target}';

Authentication: SP_fake

auth_sp_fake_authenticated

- **description:**
- **mandatory:**
- **type:** boolean
- **default:**
- **available:**
- **1.x name:**
- **comment:**

auth_sp_fake_uid

- **description:** UID you want to have
- **mandatory:**
- **type:** string
- **default:**
- **available:**
- **1.x name:**
- **comment:**

auth_sp_fake_email

- **description:**
- **mandatory:**
- **type:**
- **default:**
- **available:**
- **1.x name:**
- **comment:**

auth_sp_fake_name

- **description:**

- **mandatory:**
- **type:**
- **default:**
- **available:**
- **1.x name:**
- **comment:**

Maintenance and logging

---

failed_transfer_cleanup_days

- **description:** number of days after which chunks belonging to failed or interrupted uploads will be deleted from disk on the server. If some transfer was created say 7 days ago and still not completed, the associated data is removed after 7 days.
- **mandatory:** no
- **type:** int (days)
- **default:** 7
- **available:** since version 1.5
- **1.x name:** cron_cleanuptempdays
- **comment:**

log_facilities

- **description:** defines where FileSender logging is sent. You can sent logging to a file, to syslog or to the default PHP log facility (as configured through your webserver's PHP module). The directive takes an array of one or more logging targets. Logging can be sent to multiple targets simultaneously. Each logging target is a list containing the name of the logging target and a number of attributes which vary per log target. See below for the exact definiation of each log target.
- **mandatory:** no
- **type:** array of log targets. Each target has a type and a number of parameters
- **default:** array('type' => 'file', 'path' => FILESENDER_BASE.'/log/', 'rotate' => 'hourly'))
- **available:** since version 2.0
- **1.x name:**
- **comment:**

if you define your own log_facilities, you will overwrite the default setting. Make sure to include all log targets you wish to log to. * ***General format of log target:*** array(('type' => string, , => * ***Standard parameters for all options:*** * **'level' (optional):*** restricts loglevel of current facility. Permissible values: debug, warning, info, error * **'process' (optional): allows you to separate logs from different parts of FileSender into separate logfiles, for example the REST logfile gets huge. Permissible values: CLI, GUI, REST, WEB, CRON, FEEDBACK, MISC, INSTALL, UPGRADE. Comma-separated list. *** ' __Available targets:* * 'type' => 'file' **logs to a file. You must specify a path. You can optionally specify log file rotation with 'rotate' => '', where value can be hourly, daily, weekly, monthly, yearly. *** 'type' => 'syslog' **logs to syslog.**
* 'type' => 'errror_log'__ logs to the default PHP log facility as defined in your webserver's PHP module.

__Examples:*__ examples for tpye file with different log rotations examles for type syslog

OR Array(array (type => String (errorlog,syslog,file,callable) path => String rotate => String)) mandatory: no type: array of arrays. Each one is definition of a target. Each target has a type and if needed optional parameters. default: type file.
Note: if you define your own, it will *overwrite* the default setting, not add it to the array. If you want to keep basic logging and add syslog you must add *both*. array ('type' => 'file', (permissible values?) (file, syslog, error_log (log using default php facility, puts logs in apache error logs, callable 'path' => '/logs/' 'rotate' => hourly (permissible values?) (' process' => CLI, GUI or REST (can ask to only get logs from specific parts of FileSender, so you can separate your logs between different componentes. Maybe hourly logs with REST service (they get huge)

mandatory parameter is 'type'. Permissible values file, syslog, error_log

type syslog. indent, facility. Facility sets the syslog facility used. Standard PHP syslog function parameters

callable (advanced): "I give you something you can call to log". There is one mandatory parameter "callback" which must be a php function. That will be called every time you want to log something. Level and process can be set as well. When it's called it will get the message to log and the current process. 1st argument will be message, 2nd argument process type. Can name them A and B. CAn be useful if you're searching for a particular error or for example use remote log facility. Search for particular error: write specific function to catch specific errors and drop an email when it happens.

different options for different types.

maintenance

- **description:** when true, switches the FileSender instance in maintenance mode. This allows to interrupt the service for a database upgrade or webserver restart without breaking ongoing uploads.

  When switched on:

  - all pages are replaced with the maintenance page
  - webservice returns specific exception to all requests
  - clients display a popup explaining what happens
  - clients pause uploads and put all requests they were about to make in a stack
  - clients starts to query the server on a regular basis to see if maintenance ended (server responding with no exception status)
  - when server exits maintenance mode clients restart uploading and run stacked requests and remove maintenance popup
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:**

statlog_lifetime

- **description:** The statlog is kept in the database and contains everything needed to produce usage statistics. This directive defines maximum lifetime of statslog entries (in days) after which they get deleted. point to more text detailing what is actuallly logged in the statlog!
- **mandatory:** no
- **type:** int (days)
- **default:** 0
- **available:** since version 2.0
- **1.x name:**
- **comment:** The statlog is always enabled. If you don't want anything logged, set this lifetime to 0. Use this setting to control the privacy footprint of your FileSender service.

auth_sp_additional_attributes

- **description:** Allows to define additional user attributes that will be asked for, such as organisation, that can then be propagated to the statistic log table in the database for use in creating statistics. This configuration parameter defines the additional attributes to get. definition of additional attributes to get, array of either attributes names or final name to raw attribute name pair or final name to callable getter pair
- **mandatory:** no
- **type:** array of attribute names or name to raw attribute pair or name to callable getter pair
- **default:** - (which means do not get any additional attributes)
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- **example:** need an example here!

auth_sp_save_user_additional_attributes

- **description:** if set to true, the additional user attributes are saved in the userpreferences table.
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:** what was the point of this again?
- **example:** ...

statlog_log_user_additional_attributes

- **description:** if set to yes, the additional attributes defined in auth_sp_additional_attributes are logged in the statlog table. This allows you to do e.g. per organisation statistics or show the use for students, employees, researchers.
- **mandatory:** no
- **type:** boolean

- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- **example:** ...

auth_sp_fake_additional_attributes_values

- **description:** array of name to value pairs for fake sp authentication (testing only)
- **mandatory:** no
- **type:** array of name-value pairs
- **default:** -
- **available:** since version 2.0
- **1.x name:**
- **comment:**
- **example:** needs example!

auditlog_lifetime

- **description:** The auditlog is kept in the database and contains all events for a transfer. This information can be used to tell the user what happened to their transfer when. This directive specifies the maximum lifetime of auditlog entries (in days). If set to 0 we remove data when the transfer is closed, after sending reports (if user indicated they wanted). As long as transfer is live you have this data, as soon as transfer expires the log disappears. If you set it to "false" we don't log anything and a user can't even see the logs when a transfer is live.
- **mandatory:** no
- **type:** boolean/int (days). Set to false to disable.
- **default:** 31
- **available:** since version 2.0
- **1.x name:**
- **comment:** Use this setting to control the privacy footprint of your FileSender service.

report_format

- **description:** A user can ask for an audit report specifying what happened to a transfer when. This can be done when initiating a transfer by ticking the checkbox or explicitly through MyTransfers (view audit log). This setting specifies what type of report will be generated.
- **mandatory:** no
- **type:** keyword (string).
- **permissible values:** inline, PDF.
- **default:** inline
- **available:** since version 2.0
- **1.x name:**
- **comment:** The same information is sent regardless of format. Inline sends an email in plain text and HTML, with all information inline. If PDF is chosen, the report is sent as PDF attachment. Building a PDF is somewhat heavier on the server but won't matter unless you would have a heavily used server. The library used is "dom pdf", included in the code.

Webservices API

---

auth_remote_application_enabled

- **description:** enable or disable remote application authentication. Needed to let remote applications (API applications) authenticate
- **mandatory:** no
- **type:** boolean
- **default:** false (not explicit)
- **available:** since version 2.0
- **1.x name:**
- **comment:** needs to be elaborated more. Consequences of setting to true

remote_applications

- **description:** list of remote applications. This is an array where each entry includes an authentication secret, whether or not the application has admin rights and what methods the application is allowed to use:

- **mandatory:** no
- **type:** array
- **default:**
- **available:** since version 2.0
- **1.x name:**
- **comment:** needs more work. Example: array (idApp => secret(string), isAdmin(bool), acl (array (endpoint(ou *) => boolean OU array (pair de nom de méthode et de valeurs d'accès. ex: get => TRUE, post => FALSE Explained in more detail in API documentation page.

auth_remote_user_autogenerate_secret

- **description:** ask etienne how this works
- **mandatory:** no
- **type:** boolean
- **default:** false
- **available:** since version 2.0
- **1.x name:**
- **comment:**

disclosed

- **description:** the webservice has an endpoint called "info" which discloses information about the FileSender instance. By default it gives the URL of the FileSender instance. This parameter allows you to add more info from the configuration file. E.g. when using a remote client this client needs the chunk size.
- **mandatory:** no
- **type:** boolean/array of strings
- **default:** - (disclose nothing)
- **available:** since version 2
- **1.x name:**
- **comment:** the parameter needs an array of strings. The strings are configuration parameters you want to appear in the "info" webservice endpoint. You can also give it static strings that have a specific meaning for you, like "version 2.0".
- **example:** example comes here.

rest_allow_jsonp

- **description:** Define additional REST-API end points JSONP can be called upon. JSONP is typically used when using FileSender in an iframe. We limit which API end points you can reach in such a scenario but give you the option of enlarging that set of API end points in case you need this.
- **mandatory:** no
- **type:** boolean/array of strings
- **default:** Authorised by default are these api end points: /info, /lang, /file/[0-9]+/whole and /user/@me/remote_auth_config (if remote user authentication is enabled).
- **available:** since version 2
- **1.x name:**
- **comment:**
- **example:** Autorized by default are :

/info : public infos about the instance (name, login url ...) /lang : UI translations getter /file/[0-9]+/whole : legacy upload endpoint /user/@me/remote_auth_config : enabled only if remote user authentication is enabled

Additionnal allowed endpoints can be added through the "rest_allow_jsonp" configuration parameter (array of regexp to match the resource path under rest.php), example :

$config['rest_allow_jsonp'] = array('/transfer/[0-9]+/auditlog');

Other

---

host_quota

- **description:** use this when your FileSender instance needs to share its storage with other applications. If set to a positive value it defines the total amount of storage your FileSender instance can use for storing files. New transfers that require more space than is available are rejected with an error message in the Web-UI. Set to 0 to disable.

- **mandatory:** no
- **type:** int (in bytes)
- **default:** 0
- **available:** since version 2.0
- **1.x name:**
- **comment:**

config_overrides

- experimental feature in 2.0, not tested
- **description:** In version 2.0 you can create virtual FileSender instances (see the administrator guide. Todo: write how to do this in the admin guide!). With the config_overrides directive you specify the list of parameters an admin for a virtual FileSender instance you can override from admin interface. When you set this parameter a "Config" tab becomes visible in the Admin tab in your FileSender UI. If you have one instance you can use this to separate roles between system admin and filesender admin. You can also use this to automate FileSender virtual instance deployment.
- **mandatory:** no
- **type:** array of key-value pairs
- **default:** 0, null, empty string: you won't get the config tab in the admin interface. Any previously done override will be ignored. They're not lost but no longer applied.
- **available:** since version 2.0
- **1.x name:**
- **comment:** example:
  - $config['config_overrides'] = array('site_name_in_header' => 'bool', 'site_name' => array('type' => 'string', 'validator' => 'is_string'), 'terasender_start_mode' => array('single', 'multiple'),);

  In this example the "site_name_in_header" is a checkbox in the UI. For the override "site_name", type string: displays a text field, and runs validator "is_string". You can use existing validators or any other function. The override "terasender_start_mode" displays a dropdown in which you can choose from different predefined values.

Changes are saved in config_overrides.json in the config directory. The config.php file is NOT modified. This keeps overrides separated from the site config. is_string, is_numeric (standard php validators) or a function of your own which returns a boolean indicating if the value is good or not.

#

- **description:**
- **mandatory:**
- **type:**
- **default:**
- **available:**
- **1.x name:**
- **comment:**

#

- **description:**
- **mandatory:**
- **type:**
- **default:**
- **available:**
- **1.x name:**
- **comment:**

Available in 1.x, not in 2.0

cron_shred: consolidated by having a parameter to specify which delete command to use. debug: use log_facilities to set a log level. max_email_recipients: replaced with max_transfer_recipients and max_guest_recipeints

terasender_chunksize: chunksize is now consolidated in 1 parameter for all uploads? terasender_jobsPerWorker: didn't have any practical meaning (doublecheck with Etienne) webWorkersLimit: renamed to terasender_worker_count. before you could launch several workers and each worker would request jobs. There were # jobs per worker. Testing showed having more than 1 job per worker gained nothing. When you have browser process (tab in chrome) and doing async stuff (launch ajax request) get time to do other things. This was not way workers were thought to behave. Worker is not efficient when doing async stuff. Several jobs per worker = async. Theory: several jobs per worker can mean that when one job sends blob, other job can fetch data. No significant gain observed. Code was more complex so simplified.

crlf: now have a constant for that. This parameter was important when windows was not respecting line delimiters in emails. Had to make this configurable in the past when some old Windows clients (Outlook) used different newline format. Really long time since this was a problem.

voucherRegEx: now hardcoded in utilities. app was generating unique Ids with own algorithm that you can't change from the config. Why does the checking regexp be configurable. Changing it you can change the way the unique id looks which is a Bad Idea. You could only really simplify it (make it less strict) thus reducing security.

openSSLKeyLength: generated in utility. Method "generate_uid". don't realy on openssl to generate unique IDs. OpenSSL was used to be sure we had something unique. Added dependency on OpenSSL. Needs to be unique, non-guessable and properly random. Using random_uid_generation (6 calls to mt_rand , build X-string, put dashes. Solved by when generating unique ID. Wwas used to generate random unique ids. Adding dependency on openssl. Was not that much more secure than generating unique IDs. Unique IDs were generated before without collision checking. Now we check for that until we get a real unique one. Removing it removed a dependency. Note: need to double-check how properly random the resulting UIDs are.

emailRegEx: now using PHP built-in facility for checking email address validity which these days works well. Basic function is filter_var. Give it a variable and a filter to use. Using filter FILTER_VALIDATE_EMAIL.

Changed defaults from 1.x to 2.0

email_newline is now "\r\n", before \n terasender_enabled is now "true", before false

Relevant for security audits

library included, dom pdf.

Viewing version 66 created
about 1 year by jornane

Rollback to this version

Recent Versions

Choose two versions to compare, or click the link to view it.

☑ 69. 10 months ago by jornane
☑ 68. 10 months ago by jornane
□ 67. 10 months ago by jornane
□ 66. about 1 year ago by jornane
□ 65. about 1 year ago by jornane
□ 64. about 1 year ago by jornane
□ 63. over 1 year ago by meijer
□ 62. over 1 year ago by meijer
□ 61. over 1 year ago by meijer
□ 60. over 1 year ago by meijer
□ 59. over 1 year ago by

meijer

□ 58. over 1 year ago by meijer

□ 57. over 1 year ago by meijer

□ 56. over 1 year ago by meijer

□ 55. over 1 year ago by jornane

□ 54. over 1 year ago by meijer

□ 53. over 1 year ago by meijer

□ 52. over 1 year ago by jornane

□ 51. over 1 year ago by meijer

□ 50. over 1 year ago by meijer

□ 49. over 1 year ago by meijer

□ 48. over 1 year ago by meijer

□ 47. over 1 year ago by meijer

□ 46. over 1 year ago by meijer

□ 45. over 1 year ago by meijer

□ 44. over 1 year ago by meijer

□ 43. almost 2 years ago by meijer

□ 42. almost 2 years ago by meijer

□ 41. almost 2 years ago by meijer

□ 40. almost 2 years ago by meijer

□ 39. almost 2 years ago by meijer

□ 38. almost 2 years ago by meijer

□ 37. almost 2 years ago by meijer

□ 36. almost 2 years ago by meijer

□ 35. almost 2 years ago by meijer

□ 34. almost 2 years ago by meijer

□ 33. almost 2 years ago by meijer

□ 32. almost 2 years ago by meijer

□ 31. almost 2 years ago by meijer

□ 30. almost 2 years ago by meijer

□ 29. almost 2 years ago by meijer

□ 28. almost 2 years ago by meijer

□ 27. almost 2 years ago by meijer

□ 26. almost 2 years ago by meijer

□ 25. almost 2 years ago
by meijer

□ 24. almost 2 years ago
by meijer

□ 23. almost 2 years ago
by meijer

□ 22. almost 2 years ago
by meijer

□ 21. almost 2 years ago
by meijer

□ 20. almost 2 years ago
by meijer

□ 19. almost 2 years ago
by meijer

□ 18. almost 2 years ago
by meijer

□ 17. almost 2 years ago
by meijer

□ 16. almost 2 years ago
by meijer

□ 15. almost 2 years ago
by meijer

□ 14. almost 2 years ago
by meijer

□ 13. almost 2 years ago
by meijer

□ 12. almost 2 years ago
by meijer

□ 11. almost 2 years ago
by meijer

□ 10. almost 2 years ago
by meijer

□ 9. almost 2 years ago
by meijer

□ 8. almost 2 years ago
by meijer

□ 7. almost 2 years ago
by meijer

□ 6. almost 2 years ago
by meijer

□ 5. almost 2 years ago
by meijer

□ 4. almost 2 years ago
by meijer

□ 3. almost 2 years ago
by meijer

□ 2. almost 2 years ago
by meijer

□ 1. almost 2 years ago
by meijer

Show Diff

Clear All

History Key

- New Content
- ~~Removed content~~