CIFEX 10.01 Admin Manual

Executive Summary

CIFEX (CISD File EXchanger) allows internet users to share large files using only a modern web browser (such as Firefox, Safari or Internet Explorer) on the client side. It was inspired by the ELF system (see http://elf.sourceforge.net) in terms of features, but it supports uploading of files larger than 2GB, should be easier to maintain for the administrator, and should give the users more information and control about the files they have uploaded and the temporary accounts they have created.

On the server side, CIFEX requires only a Java Runtime Environment (Sun Java 5.0 or newer) and access to an SMTP server so that the CIFEX server can send out emails. You can make it use a PostgreSQL database, but you don't have to.

Table of Contents

Functionality

CIFEX allows you to:

- Upload files in a simple Web GUI, Java GUI or command line client and define users who can download them. This can be users with a regular (permanent) account or users you, as a user of CIFEX, just create for the purpose to allow them the download.
- Download files that someone else uploaded for you, using the Web GUI, Java GUI, or command line client.
- Create a (temporary) user account for a person you want to have upload something for you.

All files and temporary user accounts expire automatically after some time, so the server will clean up after you. For typical ways to use the system, see Use Cases below.

User Roles

There are 3 user roles in the application. If you don't have a user account, you can not access the system.

The roles are the following:

Regular User

A regular user has permanent access to the system:

- Upload Files for any existing user of the CIFEX instance
- Download the files, which another user has uploaded for him or her
- Renew the expiration date of files that he or she has uploaded
- Delete files that he or she has uploaded prior to its expiration time
- Change the list of people that a file is shared with
- Create a temporary user account
- Renew the expiration data of a temporary user that he or she created
- Delete a temporary user account that he or she created prior to the account's expiration time
- Edit the profile (password, email address and full name) of any temporary users that he or she created (with or without user notification)
- Edit their own profile (password, email address and full name) – note: this is only true if the user has not been authenticated by an external authentication service
- Switch to external authentication method (if he or she has an account in such a service and it is connected to CIFEX) If a user provides an email address during file upload that does not belong to any known user in the system, a temporary user account will be created automatically by the CIFEX server.

Administrator

An administrator has full control over the system. He or she can use the system as a regular user (see above) but additionally he or she can:

- Create another administrator or regular user
- Edit the profile (Role, Password, Username, Email) of any user that has not been authenticated via an external authentication service (with or without user notification); the profile of a user can contain special settings for the file retention time and the max upload size which overrides the server default
- Activate or de-activate users

- Renew the expiration date of any file
- Delete any file
- Renew the expiration date of any temporary user
- Delete any user
- Change account name (user code)

Temporary User

A temporary user has a limited life time (the length of which can be configured, see Configuration). After this time, the user expires and can't access the system anymore. A temporary user can do the following in the system:

- Upload Files and share them with any existing user of this CIFEX server instance
- Change the list of people that a file is shared with
- Download the files, which another user has uploaded for him or her
- Renew the expiration date of files that he or she has uploaded
- Delete files that he or she has uploaded
- Switch to external authentication method (if he or she has an account in such a service and it is connected to CIFEX)

If a regular user or an administrator uploads files and inserts an email address, which does not belong to an already existing user, the system will create a temporary user account automatically and send an email with the access information. Temporary users are not permitted to create other temporary users.

Installation

Quick Installation

The quick installation procedure assumes that a Java Runtime Environment (Sun JRE 5.0 or later has been tested) is installed, that a local SMTP server is running on the server, and that the port 8443 (for HTTPS) is available and accessible (no firewall is blocking this port). (Standard port that CIFEX is running on may be changed in `jetty.properties` file, which is located in `bin/` sub-directory of server directory and linked to in the `etc/` sub-directory).

1. unzip distribution and change to unzipped distribution folder
2. run (where `<server-dir>` is the directory you want to install the server to):

```
./install.sh <server-dir>
```

Prerequisites

You need to have the following prerequisites for a server installation:

- A Posix environment (it has been tested on Sun Solaris, Redhat Linux, openSUSE and Apple MacOSX)
- Java Runtime Environment (tested with Sun JRE 5.0 and 6.0) installed on your system
- Accessible SMTP server (for the configuration: see Email Configuration below)

Configuration

A properties file (named `service.properties`) contains all configuration parameters (except the logging configuration parameters, see below). Without specifying such a file at installation the following parameters are used:

```
############################################################
# Template of service.properties for CIFEX
############################################################

# Supported: no-authentication-service, crowd-authentication-service (see
http://www.atlassian.com/software/crowd)
external-authentication-service = no-authentication-service

# Relative or absolute path to the file store.
file-store = file-store
# Default file retention time (in days). Needs to be smaller than max-file-
retention.
file-retention = 7
# Default user retention time for temporary users (in days). Needs to be smaller
than max-user-retention.
user-retention = 7
# Maximum user retention time for temporary users (in days).
max-user-retention = 14
# Maximum file retention time (in days).
max-file-retention = 14
# Maximum size (in megabytes) of uploaded files per user (more precisely per
quota group), leave empty for no limit.
quota-file-size = 5000
# Maximum number of uploaded files per user (more precisely per quota group),
leave empty for no limit.
quota-file-count = 100
# Whether new externally authenticated users will start as 'active' ('true') or
'inactive' ('false')
new-externally-authenticated-user-start-active = true
# The number of trigger permits for this server. This determines the number of
asynchronous triggers
# that can run in parallel. (More expensive triggers may use up more than one
permit.)
trigger-permits = 4
# A negative time (in minutes) indicates the session should never timeout.
session-timeout = 15

script-folder = .

# Supported: postgresql, h2
database.engine = h2
# BE CAREFUL! If this property is changed to 'true', the next restart of the
server will scrub the CIFEX database!
database.create-from-scratch = false
database.script-single-step-mode = false
# Leave empty to get the default for the db engine
database.url-host-part =
database.kind = productive
# Credentials of the database user which should own the database. Leave empty to
use the OS user
database.owner =
database.owner-password =
# Credentials of a database user which is able to create a new database. Leave
empty to use the db engines default
database.admin-user =
```

```
database.admin-password =

# The name of the Crowd server
crowd.service.host = <crowd server>
# The port that the Crowd service can be reached at
crowd.service.port = 8443
# Name of the application when logging in into Crowd
crowd.application.name = cifex
# Password for which the Cifex service is registered at Crowd
crowd.application.password = <application password>

# The IPs from which changing the session user by an API call is allowed (comma-
separated list)
allowed-ips-for-set-session-user =

# SMTP properties
# Default values are 'localhost' for 'mail.smtp.host' and 'cifex@localhost' for
'mail.from'
mail.smtp.host = localhost
mail.from = cifex@localhost
mail.smtp.user =
mail.smtp.password =
# Set this if you want to override the base URL shown in the emails
mail.url.override =

# Administrator email address
administrator.email = root@localhost
```

The configuration of the logging parameters is contained in a file `log.xml`. Most of the content of this file can remain unchanged. However, the logging system should be enabled to send messages to an administrator (*notification log messages*). To this end, the email configuration needs also to be given in this file. The default `log.xml` file contains the following section:

```
  <appender name="EMAIL" class="org.apache.log4j.net.SMTPAppender">

    <param name="BufferSize" value="512" />
    <param name="SMTPHost" value="localhost" />
    <param name="From" value="cifex@localhost" />
    <param name="To" value="root@localhost" />
    <param name="Subject" value="ATTENTION: CIFEX" />

    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d %-5p [%t] %c - %m%n" />
    </layout>

  </appender>
```

This section is suitable if a local SMTP server has been set up on the server running CIFEX and the `root` user on this host is supposed to get the notification log emails. Otherwise the parameters of the `SMTPAppender` in `log.xml` need to be adapted accordingly.

Email Configuration

The easiest way to configure the email (smtp host, smtp user and smtp password) is to provide the correct parameters in `service.properties` and `log.xml` **prior to the installation**. Should you need to change the files after the installation, you will find a link to these files in the `etc/` sub-directory of the server directory. A restart of the server is required in order to make changes to those files effective.

Legal Disclaimer

The CIFEX server has a link to a legal disclaimer in the footer line. This disclaimer is supposed to protect the CIFEX service provider from claims by users (due to breach of security or loss of data) or other parties (due to unlawful use of the application by some users). This disclaimer should be adapted for a new instance of CIFEX. The text shown as legal disclaimer can be found in the file `disclaimer.html` in the `work/` sub-directory of the server directory. No restart is required to make changes to this file effective.

Quota

CIFEX supports quota for the total size of what a user can upload and for the number of files she can upload. As also regular users can create new (temporary) users, a regular user and all the temporary users she created form a *quota group*, i. e. they share the quota values. If, on the other hand, an admin creates a temporary user, this user will have her own quota group and will not be linked with the admin that created her in terms of quota.

CIFEX supports setting a default quota value that applies to all (non-admin) users and overriding the quota for a particular user. Setting a value of '`unlimited`' as a custom quota removes any quota restriction.

Setting up the CIFEX server

Installation

In order to install the application perform the following steps:

1. Unzip the distribution. A new folder named `cifex` with the following content will be created:
    - CIFEX-10.01-docu.pdf
    - cifex-jetty-ssl.xml
    - cifex-jetty.xml
    - cifex.war
    - cifex_cli.zip
    - COPYING
    - install.sh
    - jetty-version.txt
    - jetty.zip
    - keystore
    - LICENSE
    - log.xml
    - service.properties
    - shutdown.sh
    - startup.sh
2. Edit `service.properties` if needed (e.g. to set a SMTP server).
3. Install the server with the following command:

   ```
   ./install.sh  [--port <https port>] <server folder>
   ```

   A new subfolder will be created (`jetty`), which contains startup and shutdown scripts and the startup script will be executed.

If no <service properties file> and <log configuration file> are provided on the command line, the files service.properties and log.xml in the installation directory will be used.
If the ports are not provided as a command line parameter during installation - standard values are used. The ports can be changed later in jetty.properties file, which is located in bin/ subdirectory of server directory.

The First User

The first user gets created automatically when you login to the system for the first time after installation. It will always have admin permissions. Use this user to create all other users.

- [ ] Be careful when typing the password. If you can't remember the password later on you will not be able to access the server and will have to reinstall it!

Start Application

In order to start the application change to <server folder>/jetty and execute

```
bin/startup.sh
```

Stop Application

In order to stop the application change to <server folder>/jetty and execute

```
bin/shutdown.sh
```

Where is my stuff actually stored on the disk?

CIFEX uses a **file store** (which holds the actual files) and a **database** (which holds the users, the information about which files are available to whom, user and file expiration, etc.).

The root directory of the file store can be configured in service.properties, key file-store. By default this is set to the file-store sub-directory of the server directory (which is the first argument you provide to the install.sh script, appended with /jetty). Below the file-store directory you will find sub-directories for each user who has uploaded files. These sub-directories contain the uploaded files themselves.

For the database engine you have two choices:

- h2 (see http://www.h2database.com), a Java in process database that doesn't require any installation on your system
- PostgreSQL 8.2 or later (see http://www.postgresql.org), a mature Open Source standalone relational database. This database needs to be installed (and configured) separately from CIFEX prior to installation. CIFEX needs a superuser account (typically called postgres) in order to do the schema installation. It will perform the schema installation the first time the application is started up. The database will be called cifex_productive by default. The second part (i.e. "productive") can be configured in the service.properties using the key "database.kind".

By default, the h2 database is configured. In this case the files constituting the database are stored in the sub-directory db/ of the CIFEX server directory.

Upgrade to a new release

When upgrading the CIFEX server, the administrator should distinguish between an upgrade to a new *major release* and an upgrade to a *bugfix release* of the same *major release*. *Major releases* are distinguised by the release date in the format `year.month`, e.g. `8.06` for the release of June 2008. *Bugfix releases*, on the other hand, share the same release month but are distinguished by the patch number. For example `8.06.1` is a bugfix release on `8.06.0`, but they are both major release `8.06`.

Upgrade to a new major release

> **Warning**
> Before upgrading the CIFEX server to a new major version, the administrator is strongly encouraged to backup the CIFEX database in order to be able to restore it in case the automatic database migration fails!

Upgrading a CIFEX server to a new *major release* consists of these straightforward steps:

1. Unpack the installation directory of the new release version of CIFEX.
2. Transfer the custom changes you made to `service.properties` and `log.xml` to the files in the installation directory of the new version. It is recommended not to copy over those files from the old server directory without reviewing them as their structure may have changed from the old to the new version. (This warning is particularly important for the `service.properties` file.)
3. Stop the old version of CIFEX.
4. Backup the CIFEX database. When using h2, this requires `tar`-ing the `db/` sub-directory of the old version of the CIFEX server directory to a safe place. When using PostgreSQL, the tool `pg_dump` may be used to produce a database dump of the database (which by default is called `cifex_productive`).
5. Move the old CIFEX server directory to a safe place and keep it until you know the new version works for you.
6. If using the h2 database, create the anticipated new CIFEX server directory and copy the `db/` sub-directory from the old server directory to the new server directory.
7. Call the `install.sh` script with the new CIFEX server directory, optionally providing a custom port number if required.
8. Check the log file of the new CIFEX server for error messages.
9. Login to the new CIFEX server as admin and check that you see all users and files.

Upgrade to a new bugfix release

For *bugfix releases* the process is simpler as bugfix release are guaranteed not to change the database schema and the configuration file format:

1. Unpack the installation directory of the new release version of CIFEX.
2. Copy over your custom version of `service.properties` and `log.xml` from the old server directory.
3. Stop the old version of CIFEX.
4. Move the old CIFEX server directory out of the way for the new version.
5. If using the h2 database, create the anticipated new CIFEX server directory and copy the

db/ sub-directory from the old server directory to the new server directory.
6. Call the `install.sh` script with the new CIFEX server directory, optionally providing a custom port number if required.
7. Check the log file of the new CIFEX server for error messages.

What is the option allowed-ips-for-set-session-user good for?

The CIFEX remote API has a method `setSessionUser()` which roughly corresponds to the `su` command under Unix, i.e. it allows an admin user to "become another user". This call can be used by another application to perform a data upload on behalf of another user without requiring this user to provide her password. Files uploaded in this way will show up as being uploaded by the user that has been provided to the `setSessionUser()` call. In order to avoid abuse of this function, it is disabled by default and can only be enabled by the person configuring CIFEX on a "per IP address" bases. The white list of IPs from which the call should be allowed has to be provided in the configuration option `allowed-ips-for-set-session-user`.

Triggers

Triggers are Java classes that can be bound to a certain user account in CIFEX (which is then called a "trigger user"). If a file gets uploaded for a trigger user, a method of this class is called that can process the file that was uploaded. This file is also referred to as a "request".

Configuring triggers

- copy the `jar` file(s) containing the triggers to the folder `jetty/work/webapp`
- write a file `etc/triggers.txt` that maps users to triggers (see below)
- restart the server

The file `triggers.txt` is a tab-separated file with either 2 or 3 columns. The first column contains the user id of the trigger user. The second column contains the fully qualified class path of the trigger class. The third column may contain the name of a properties file. If it does, a `Properties` object filled with the values from this file will be provided to the constructor of the trigger class. If there is no third column for a line, the default constructor of the trigger will be called instead.

Example:

```
echo    ch.ethz.somepackage.Echo        echo.properties
```

Writing triggers

See here for instructions on how to write a trigger: Writing a trigger for CIFEX.

HTTP links to CIFEX server for triggers

There is a handy way of linking to CIFEX when using triggers:

```
https://cifex.your.org/?
comment=This_is_my_comment&recipients=id:first_recipient,user@foreignplace.net
```

This will pre-populate the recipients field and optionally the comment field with the given values. The values specified will not be editable in the web mask.