

Freva – BAG – Basic Administration Guide (ALPHA VERSION)

November 19, 2016

1 Introduction - Welcome to Freva

In this guide we will explain the basic workflow to install and administrate Freva. At first we recommend to set up a proper environment [Chapter 2]. We've prepared an easy installation script which installs Freva and its components to be fully operatable in the shell. Therefore we have a Quick Installation Guide [Chapter 3] and a Long Installation Guide [Chapter 4] with more infos. Afterwards we explain how to setup its web-frontend [Chapter 5]. In the end we name basic rules and issues within the connection of Freva with HPCs, SLURM scheduler, etc. to keep an eye on or set up cron jobs [Chapter 6]. For all other informations around Freva, feel free to have a look in the Paper [], the Basic User Guide (BUG), and the Basic Developer Guide (BDG).

2 Environment Setup

Freva is designed to be installed in a Linux/Unix environment. It could/should have access to large databases of standardized data (CMOR, CMIPX, ESGF, etc.) and compute resources for complex evaluation procedures. Probably, this is a High Performance Computer (HPC).

1. Suggestion: We suggest to setup one node/machine for Freva and its purposes. This machine must have access to all databases (e.g. home, scratch, work, etc.). A virtual machine could be possible, depending on the workload of the system. An estimation would be less than 2 million standardized files and less than 100 users at the same time, a virtual machine with 8GB RAM and 8 CPUs should be fine.

2. Suggestion: We suggest to setup a Freva system user. This user installs Freva and its components. It makes the development process much more easy if more than 1 person administrates the system. And if its this 1 person and she leaves, the handover is more complicated. However, due to security issues we recommend not use 'root' to host Freva. Freva is fully capable to be run by a normal/system user to host the full system. During the following installation process, you will need some help of the admins of your machine or root access to install basic software packages, if not already installed.

3. Suggestion: If Freva is an open system to everyone on this machine/node, make

sure that the unix rights are set correctly. If Freva should be exclusively to a specific group, we recommend putting the Freva user into that group and use it as the default group. The unix rights of Freva must be set correctly, that no other user is able to access data of that group. Different research institutes have different requirements. Freva is flexible enough to be adapted, but it is not possible to meet all requirements in defaults.

3 Quick Installation Guide - the fastest way, if you know what wo do

3.1 Step 1: Needed Software

MySQL server: Install, give access to it, and open its port.

SOLR port: Open port 8983.

SLURM scheduler: Install it.

Software: bash, mysql, git, python-dev, java (1.6, 1.7, 1.8), libmysql(-dev), libffi(-dev), libssl(-dev), libsabl(-dev), httpd(-dev), netcdf4, hdf5, cdo, nco, wget, curl

3.2 Step 2: The Install-Script

Download install-script:

```
wget https://gitlab.met.fu-berlin.de/freva/evaluation_system/  
raw/master/build/install.sh
```

Adapt install-script: Fill out the config area and execute it with switched switches

3.3 Step 3: The Basic Setup and Testing

Load Freva: modules or source

```
ls /path/2/project-ces/misc4freva/loadscripts/
```

Test plugin:

```
export EVALUATIONSYSTEM_PLUGINS=/path/2/project-ces/  
plugins4freva/example; freva --plugin ExamplePlugin project=  
bla product=bla institute=bla model=bla
```

Test history:

```
freva --history ENTER
```

Test databrowser:

```
ln -s /path/2/project-ces/misc4freva/db4freva/cmor4freva/  
example_structure4solr/project /path/2/project-ces/  
data4freva/crawl_my_data/project
```

```
/path/2/project-ces/freva/sbin/solr_server path2ingest /path/2/  
project-ces/data4freva/crawl_my_data/project  
freva --databrowser ENTER
```

Test crawl_my_data:

```
mkdir /path/2/project-ces/data4freva/crawl_my_data/user-smith  
ln -s /path/2/project-ces/misc4freva/db4freva/cmor4freva/  
example_structure4solr/project/product /path/2/project-ces/  
data4freva/crawl_my_data/user-smith/product  
freva --crawl_my_data  
freva --databrowser project=user-smith ENTER
```

Test batchmode:

```
freva --plugin ExamplePlugin project=user-smith product=product  
institute=institute model=model experiment=experiment  
time_frequency=time_frequency variable=variable --batchmode=  
True
```

4 Long Installation Guide - more infos but maybe boring

Freva is written in Python 2.7. This is the main software requirement. Therefore we recommend to set up its own Python instance. This could be done automatically by the install script, which is recommended. Other software packages need to be installed beforehand. The following subsection will give an overview of this software.

4.1 Step 1: Needed Software

4.1.1 MySQL Server

Freva saves all configurations and user interactions in a MySQL database. If you already have a MySQL Server, set up a new table with an MySQL-User and its password. If not, please install a MySQL server on e.g. the designated Freva node/machine. Open its port on the machine/node that other machines/nodes in the network are able to connect. The setup of tables will be done by the install-script. So no need to do it now.

4.1.2 Port for Solr Server

The Solr server collects all information of the standardized data in the file system. The databrowser will use this. The Solr server will be installed automatically by the install-script of Freva. However, you have to open the port 8983 on the machine/node where the install will take place. It should be accessible in your network, that other machines/nodes are able to connect.

4.1.3 Slurm Scheduler

Freva is designed to use Slurm as a scheduler in its batch mode. Therefore, the workload can be spreaded over all available machines/nodes - in a smart way. The Freva web-frontend depends on the batch mode. Please install Slurm and connect designated compute machines with it. If you already have another scheduler and want to use this, you have to adapt the scheduling scripts [We've never done that before, but it is possible]. If you don't have any scheduler, and you don't want any scheduler, Freva is able (beta phase) to send jobs to machines using 'nohub' without any control of RAM and CPU usage by several users (not recommended). Of course it is possible, to host Freva without webpage just in its shell part.

4.1.4 Software Packages

Here comes a list of typical needed software packages in the naming convention of Debian/Fedora/etc. If you are on a different Linux/Unix machine (e.g. Suse), make sure to look up its equivalent:

```
bash, mysql, git, python-dev, java (1.6, 1.7, 1.8), libmysql(-dev), libffi(-dev), libssl(-dev), libsabl(-dev), httpd(-dev), netcdf4, hdf5, cdo, nco, wget, curl
```

Side-Note: Basic Linux/Unix packages of different distributions of course can change. Therefore the list could get extended in the future. Please share your experience of this process.

4.2 Step 2: The Install-Script

We provide an install script which takes care of the whole install process. The basic steps are downloading it, adapt it to your needs, start it.

4.2.1 Download

Download the script via git into a temporary direction:

```
wget https://gitlab.met.fu-berlin.de/freva/evaluation_system/raw/master/build/install.sh
```

4.2.2 Adapt

Please adapt the script to your requirements. Freva hosts an evaluation system of a institute, project, or something similiar. Give it a name and a root path. This is an example:

```
NameYourEvaluationSystem=project-ces  
Path2Eva=/path/2/
```

Then Freva will be located here:

```
/path/2/project-ces/freva
```

For example next to its plugins or data:

```
/path/2/project-ces/plugins4freva  
/path/2/project-ces/data4freva
```

Note: No need to set it up like this, will be done automatically. If you have filled out the configuration area, you can start the install-script. This will set up the directory structure and sets unix rights. No real install process will be started as long as the switches at the top are set to 'False'. If you install the whole chain for the first time, we recommend to do it step by step - from top to bottom.

4.2.3 Python via install-script

Set 'makeOwnPython' to True.

Start the script. The install-script downloads, configures, and installs Python 2.7. After that it will install 'pip' and uses 'pip' to install the needed python packages for Freva. Depending on the machine and its software stack, this could be a fast and easy way. If problems occur, have a look into the python log files, located:

```
/path/2/project-ces/misc4freva/python4freva
```

Maybe do the python install process step by step as well, have a look into the install-script. Talk to your admin... If successfully installed, next step.
Set 'makeOwnPython' to False.

4.2.4 Freva Source Code via install-script

Set 'makeFreva' to True.

Start the script. The install-script downloads the Freva source code via git from the gitlab of the Freie Universität Berlin [github?]. Now you should have Freva within the directory:

```
/path/2/project-ces/freva
```

Set 'makeFreva' to False.

4.2.5 Configuration of Freva via install-script

Set 'makeConfig' to True.

Start the script. The install-scripts configures the main configuration file with the informations you provided in the configuration area of the install-script. Additionally, the basic data structure for the standardized database will be setup. The configuration files will be set up here:

```
/path/2/project-ces/misc4freva/conf4freva
```

Set 'makeConfig' to False.

4.2.6 Startscript of Freva via install-script

Set 'makeStartscript' to True.

Start the script. The install-script sets up two kind of start scripts for the project-ces. A source file to be sourced and a module file to be loaded. Depending on what your machine/node having a module system or not. The files are located here:

```
/path/2/project-ces/misc4freva/loadscripts
```

Set 'makeStartscript' to False.

4.2.7 Solr Server via install-script

Set 'makeSOLRSERVER' to True.

Start the script. The install-script downloads, configures, and installs the Apache Solr Server with the informations provided in the main configuration files (already set up by makeConfig). It provides some structure and gets all needed infos from the freva source code. In the last step it sets the Solr cores 'files' and 'latest'. You can check the Solr server by e.g. logging into the website <http://localhost:8983/solr>. If the website is not reachable, maybe the port as not open yet. Maybe, Java is not working correctly within the installation process. You can have a look into log files or do the installation process by hand. The Solr server and its log files should located here:

```
/path/2/project-ces/misc4freva/db4freva/solr/server
```

Set 'makeSOLRSERVER' to False.

4.2.8 MySQL tables via install-script

Set 'makeMYSQLtables' to True.

Start the script. The install-script will write the needed tables into the mySQL database. As a security check, the mySQL will ask you to type in the password, because:

WARNING: When the given mySQL account is not empty, this procedure will overwrite the whole table! All saved history entries could get lost. So when you already have such tables and you just want to re-new the rest of Freva, don't use this switch. If this a total new installation, using this script is totally fine.

Set 'makeMYSQLtables' to False.

4.3 Step 3: The Basic Setup and Testing

Congrats, you should have a running Freva instance now. However, you should test some basics.

4.3.1 Load Freva

If you have a 'module' system use the

```
/path/2/project-ces/misc4frev/loadscripts/loadfrev.modules
```

to load Freva. Otherwise just 'source'

```
/path/2/project-ces/misc4frev/loadscripts/loadfrev.source.
```

If no errors occur, type

```
frev
```

The help should appear, similar to this:

```
Freva
Available commands:
  --plugin          : Applies some analysis to the given data.
  --history         : provides access to the configuration
                     history (use --help for more help)
  --databrowser     : Find data in the system
  --esgf            : Browse ESGF data and create wget script
  --crawl_my_data   : Use this command to update your projectdata
  .

This is the main tool for the evaluation system.
Usage: frev --COMMAND [OPTIONS]
To get help for the individual commands use
  frev --COMMAND --help

Please test if the main options: --plugin, the --history, and
the --databrowser command don't produce erros:
frev --plugins
frev --history
frev --databrowser
```

The output should be nothing with these options, if no plugins are plugged-in, no plugin was ever started, and the database is empty.

4.3.2 Testing Freva Plugin

There is an example plugin in the

```
/path/2/project-ces/plugins4frev
```

directory - called example.py. You can plug this into frev by:

```
export EVALUATION_SYSTEM_PLUGINS=/path/2/project-ces/  
plugins4freva , example  
freva --plugin
```

should print out:

```
'ExamplePlugin: An example plugin incl CMOR facets '
```

This plugin just tells you which standardized files are linked into freva's databrowser. As there is no data in Freva yet, we fake the search by:

```
freva --plugin ExamplePlugin  
ERROR: Error found when parsing parameters. Missing mandatory  
parameters: project , product , institute , model , experiment ,  
time_frequency , variable  
  
freva --plugin ExamplePlugin project=bla product=bla institute=  
bla model=bla experiment=bla time_frequency=bla variable=bla  
WARNING: Could not read git version  
STARTING THE DATABROWSER  
freva --databrowser project=bla product=bla model=bla institute  
=bla experiment=bla time_frequency=bla variable=bla ensemble  
=None  
WARNING - No files found
```

The plugin does not has its own git repository, usually plugins should have to track its status. Now put the example plugin into freva, that everyone can use it.

```
/path/2/project-ces/misc4freva/conf4freva/evaluation_system.  
conf
```

Add these lines at the bottom of the file:

```
[ plugin : ExamplePlugin ]  
python_path=$EVALUATION_SYSTEM_HOME/ ../ plugins4freva / example  
module=example_plugin.py
```

Now the example plugin is part of Freva.

4.3.3 Testing Freva History

The step before was the first start of a plugin by Freva. We can test now the -history option:

```
freva --history  
1) exampleplugin [YYYY-MM-DD hh:mm:ss] {"product": "bla", "  
dryrun": true, "institute": "bla", "cache": "/n...
```


4.3.4 Testing Freva Databrowser

Now we will put some test data into the databrowser, to see if it works. In the db4freva you find some basic structure in the CMOR standard.

```
/path/2/project-ces/misc4freva/db4freva/cmor4freva/  
example_structure4solr/project/product/institute/model/  
experiment/time_frequency/realm/ensemble/  
variable_cmortable_model_experiment_ensemble_startdate-  
enddate.nc
```

Within the file.py of the configurations, we set some different possibilities/examples to put in data. (Warning: There are differences between CMOR, DRS (CMIP5), CORDEX, OBS4MIPS, etc).

```
/path/2/project-ces/misc4freva/conf4freva/file.py
```

Let's keep it easy for a moment and put in CMOR data.

```
ln -s /path/2/project-ces/misc4freva/db4freva/cmor4freva/  
example_structure4solr/project /path/2/project-ces/  
data4freva/crawl_my_data/project
```

The admin command to ingest this data into the databrowser is:

```
/path/2/project-ces/freva/sbin/solr_server path2ingest /path/2/  
project-ces/data4freva/crawl_my_data/project
```

Now, test the databrowser

```
freva --databrowser ENTER  
PATH/variable_cmortable_model_experiment_ensemble_startdate-  
enddate.nc
```

Now, test the tabbing function

```
freva --databrowser project= TAB TAB  
project
```

4.3.5 Testing Freva crawl_my_data

The crawl_my_data path has its name, because the users are allowed to put in their own data next to the official. Let's pretend, username smith has data:

```
mkdir /path/2/project-ces/data4freva/crawl_my_data/user-smith
```

user-smith is now a project within the CMOR standard! Let's link the test data into his project:

