CMIP6-DICAD – FU Berlin

Portal for the provision of results with **FREVA**
as part of the **CMIP6-DICAD** project

SPONSORED BY THE
Federal Ministry
of Education
and Research

Freie Universität Berlin

Table of Content

Version	Date	Editor	Changes
0.1	28.11.2016	Thomas Schartner	First version - draft
1.0	02.12.2016	Thomas Schartner	First iteration
1.1	23.12.2016	Thomas Schartner	Format: table of content, headlines etc.
1.2	06.01.2016	Thomas Schartner	Typos, logos

Basis functionality of Freva and the integration in CMIP6-DICAD

1. Introduction

Freva is an all-in-one solution run on high performance computers to handle evaluation systems of research projects, institutes or universities. It combines different software technologies into one common hybrid infrastructure, including all features present in the shell and web environment. The database interface satisfies the international standards provided by the Earth System Grid Federation. Freva indexes different data projects into one common search environment by storing the meta data information of the self-describing model, reanalysis and observational data sets in a database. This implemented meta data system with its advanced but easy-to-handle search tool supports users, developers and their plugins to retrieve the required information. A generic application programming interface allows scientific developers to connect their analysis tools with the evaluation system independently of the programming language used. Users of the evaluation techniques benefit from the common interface of the evaluation system without any need to understand the different scripting languages. Facilitation of the provision and usage of tools and climate data automatically increases the number of scientists working with the data sets and identifying discrepancies. Additionally, the history and configuration sub-system stores every analysis performed with the evaluation system in a database. Configurations and results of the tools can be shared among scientists via shell or web system. Therefore, plugged-in tools benefit from transparency and reproducibility. Furthermore, if configurations match while starting an evaluation plugin, the system suggests to use results already produced by other users – saving CPU/h, I/O, disk space and time. The efficient interaction between different technologies improves the Earth system modeling science framed by Freva.

2. Software requirement for Freva

2.1 Installation

For a complete installation of the Freva framework, the following software stack is needed:

2.1.1 Prerequisite:

- MySQL Server
- Slurm Scheduler

2.1.2 Software Stack (system software):

- bash (4.2.37)
- mysql (14.14)
- git (1.7.10.4)
- python-dev (2.7.10)
- java (1.6, 1.7, 1.8)
- libmysqlclient (5.5.47)
- libffi (3.0.10)
- libssl (1.0.1e)
- apache2 (2.2.22)
- netcdf (4.1.3)
- hdf5 (1.8.8)
- cdo (1.6.4)
- nco (4.4.4)
- wget (1.13.4)
- curl (7.26.0)

2.1.3 pip packages (needed by Freva):

- basemap (1.0.7)
- cdo (1.3.2)
- cffi (1.5.0)
- cryptography (1.2.2)
- cycler (0.10.0)
- Django (1.8.2)
- enum34 (1.1.2)
- fusepy (2.0.2)
- idna (2.0)
- ipaddress (1.0.16)
- matplotlib (1.5.3)
- MyProxyClient (1.4.3)
- MySQL-python (1.2.5)
- nco (0.0.2)
- netCDF4 (1.1.1)
- numpy (1.9.2)
- pam (0.1.4)
- pip (7.1.0)
- plot-engine (1.1.17)
- pyasn1 (0.1.9)
- pycparser (2.14)
- pyOpenSSL (0.15.1)
- pyparsing (2.1.10)
- pyPdf (1.13)
- python-dateutil (2.5.3)
- python-pam (1.8.2)
- pytz (2016.7)
- ScientificPython (2.8.1)
- scipy (0.15.1)
- setuptools (18.0.1)
- six (1.10.0)
- wget (2.2)
- wheel (0.24.0)

The installation is described in detail in the Freva Basic Admin Guide (BAG). The Admin Guide is available at the following url: https://freva.met.fu-berlin.de/guides/bag/

2.2 System core (short description)

2.2.1 Plugin section: The plugin framework of Freva handles the connectivity of stand-alone tools to the evaluation system of the research group through an application programming interface (API).

2.2.2 History section: In Freva all information about performed analyses is saved in a database. Each analysis have an unique ID, which is then combined with user-id, plugin name, caption, time stamp and status. The configuration parameters with possible data retrieval options (e.g. Solr fields) are saved.

2.2.3 Databrowser: The databrowser is the search engine of Freva. Freva indexes the model output directory structure of the research group and saves the meta data information in a Solr database. Solr has a faceting component. This is part of the standard request handler which allows a faceted navigation. Therefore, Freva applies the Solr faceted search on the setup data directories and datasets using standards like DRS or CMOR. All files of a chosen directory get crawled and where after all model datasets and their locations get ingested into the Solr server.

2.2.4 Help section: In the help section you can find the documentation for the Freva system itself and for the individual plugins. For easy access, there is a web page tour.

2.2.4.1 web page tour

For a better understanding of freva, a web page tour is available under the tab “help”. The web page tour gives an introduction in the databrowser, history and plugin section. New users are requested to start the web page tour to get to know the system.

2.2.4.2 Guides & documentation

There are a lot of guides in the help section for different use of the freva framework: the Freva Basic User Guide (BUG) for users, the Freva Basic Developer Guide (BDG) for developers, the Freva Basic Admin (BAG) for administrators. In addition, almost every plugin is described in the plugin documentation section.

2.2.4.3 available use cases

In the BUG section there are some use cases, e.g.:

- defining a search with freva --databrowser
- bash auto completion
- ESGF search and download options
- the use of history
- the plugin mechanism

3. Modification in CMIP6-DICAD

3.1 Plugin section:

In CMIP6-DICAD a new plugin will be written, called ESMVal collector (EVC). The main purpose of the EVC is to collect pictures and index meta data respectively tags. Tags are ingested in a database to make ESMVal output searchable, retrievable and visible. Meta data and tags are defined by the ESMVal developers. For a useful search, facets of the input file, information about regridding and reformating as well as the used diagnostic is needed. All informations are stored in an json object which can save various types of information in different levels.

Requirements for the plugin:

- copy ESMVal output (pictures, meta data and description) to a predefined folder
- read meta data from picture (EXIF header, if available), namelists and description
- rewrite meta data to an json object
- store json object in a database
- json object is stored under one plugin user

3.2 History section:

For CMIP6-DICAD the history section is used to make the ESMVal output searchable, retrievable and visible. An annotation and a discussion function will be implemented to discuss the results with the ESMVal community.

Requirements for the history section

- the main history will be the one of the result producer, not of each user
- Well defined captions to be traced
- sort the list by plugin, timestamp, meta data
- develop hidden function (similar to delete) to have the chance to order main results
- filter the list by any words, e.g. plugin name, meta data
- make the history accessible by guest user (later stage)
- each entry in the history can be commented or discussed by any guest user (later stage)

3.3 Databrowser:

In CMIP6-DICAD the databrowser is provided to the CMIP6 Community at the DKRZ and the ESMVal Developer to simplify locating and integrating local data in their diagnostic. There are no modification. It is used as it is. The databrowser can be filled up with CMIP5, OBS4MIPS, ANA4MIPS, etc. data. When CMIP6 data sets are available, they will be part of the databrowser.

3.4 Help section:

3.4.1 internal use cases

The internal use cases gives an overview of the usage of freva for the CMIP6-DICAD project.

3.4.1.1 create an ESMValTool analysis

With the ESMVal Collector plugin (see 3.1) the output of the ESMValTool will be prepared for the ingest process. This is a one click tool. After calling the EVC plugin, a python script searches for new results and copies one result from ESMVal to a predefined folder. Then the meta data are read from the EXIF, namelist and the description of the picture. This informations are rewritten to an json object and stored in a database. The json object can store various types of information in different levels. .

3.4.1.2 use the freva --databrowser for ESMVal

The use of freva --databrowser makes finding data easier. The output of freva --databrowser return the complete path with CMOR facets which can be used in the namelist of ESMVal. Detailed information of the usage of freva --databrowser gives the BUG (see 3.4.3).

3.4.1.3 show result of the ESMValTool

The history contains a list of all carried out ESMVal diagnostics with ID, user, plugin name, caption, time stamp and status as well as an info panel with results button and a tool tip info button. To display a particular outcome the user should use the search box. The user can search for different information, e.g. information about the available diagnostic, the available input data, the reformatting routines, the regridding routines, the available nameslist etc.