

Once the metalink XML-file has been created different approaches and applications can be used to download the data. In this section an example of how to use the Aria2 utility (see [?https://aria2.github.io/manual/en/html/index.html](https://aria2.github.io/manual/en/html/index.html) for more details) to download the files is briefly described. First of all, the ESGF credentials should be obtained in order to obtain the permission to download the data. To this aim we could use the API getESGFCredentials. In command line:

```
$ ESGF_HOME=.esg
$ OPENID=https://esgf-node.example/esgf-idp/openid/userID
$ OPENID_PASS=userPASSWORD
$ java -jar getESGFCredentials-0.1.4.jar --openid $OPENID --password $OPENID_PASS --writeall --output $ESGF_HOME
```

Note that the user openID, password and local repository of the ESGF credentials is defined as argument of the application. The auxiliary variables ESGF\_HOME, OPENID and OPENID\_PASS have been defined to simplify and clarify the use of the application getESGFCredentials.

Once the credentials have been obtained we can use the Aria2 utility to download the files included in our metalink XML-file. As in the previous case some auxiliary variables with the arguments of the Aria2 application are defined before call it.

```
$ METALINK_FILE=example.metalink
$ ARIA2C_SEC_OPTNS="--private-key=$ESGF_HOME/credentials.pem --certificate=$ESGF_HOME/credentials.pem --check-certificate=true"
$ ARIA2C_LOG_OPTNS="--log-level=info --summary-interval=0 --console-log-level=warn"
$ ARIA2C_FIL_OPTNS="--continue --file-allocation=none --remote-time=true --auto-file-renaming=false --conditional-get=true"
$ aria2c --enable-rpc --dir=./download/ $ARIA2C_FIL_OPTNS --log=log/aria2c_$(date +%Y%m%dT%H%M%S).log $ARIA2C_LOG_OPTNS
```

According to the auxiliary variables, this call download the files contained in the METALINK\_FILE (example.metalink) in the directory ./download (path relative to the current location) using the ESGF-credentials, contained in the ESGF\_HOME directory, and generating a log-file with the result of the download (ARIA2C\_LOG\_OPTNS).

One of the advantage of this downloading process, at least for the local downloads, is that it can be easily managed in the following web-page:

[?http://ziahmza.github.io/webui-aria2/#](http://ziahmza.github.io/webui-aria2/#)

When the download is done through one working node of the cluster it is more difficult to manage but it is possible (ask Antonio).

**NOTE: A known problem of the ESGFToolsUI application is that the results obtained include all the versions of a specific dataset, not only the latest one. To filter the different versions in the exported metalink we have created the matlab function metalinkFilter.m which has been attached along with the metalink file corresponding to the HadGEM2 Earth System Model included in the CMIP5 project. Also the following example can be used to filter the older versions in the metalink file <https://meteo.unican.es/trac/blog/FilterVersionsMetalink>**

Other older examples:

With ca certificates pem file

```
$ aria2c --private-key=credentials.pem --certificate=credentials.pem --check-certificate=true --ca-certificate=ca-certificatio
```

Without ca certificates pem file not recommended

```
$ aria2c --private-key=.credentials.pem --certificate=credentials.pem --check-certificate=false esgf_metalink_file.metalink
```