

## Table of Contents

<b>What is esgf-getCredentials?</b>	<b>2</b>
<b>Getting started</b>	<b>2</b>
Pre-requisites	2
Download	2
Run it	2
<b>Command line UI Guide</b>	<b>2</b>
<b>Graphic UI Guide</b>	<b>3</b>
Setting user	3
Setting output files	3
Retrieve credentials	3
Advanced options	4
<b>Specific use cases</b>	<b>5</b>
Aria2	5
cURL	5
WGET (OPENSSL)	6
WGET (GNU TLS)	6
ToolsUI	6
ESGF WGET Script (Linux)	6
ESGF WGET Script (cygwin)	6
NETCDF DAP	6
<b>Developers Guide</b>	<b>7</b>
Github	7
Architecture	7
<b>See Also</b>	<b>7</b>

## What is esgf-getCredentials?

A tool to retrieve user credentials from ESGF. It have one graphic interface and another command line interface.

## Getting started

### Pre-requisites

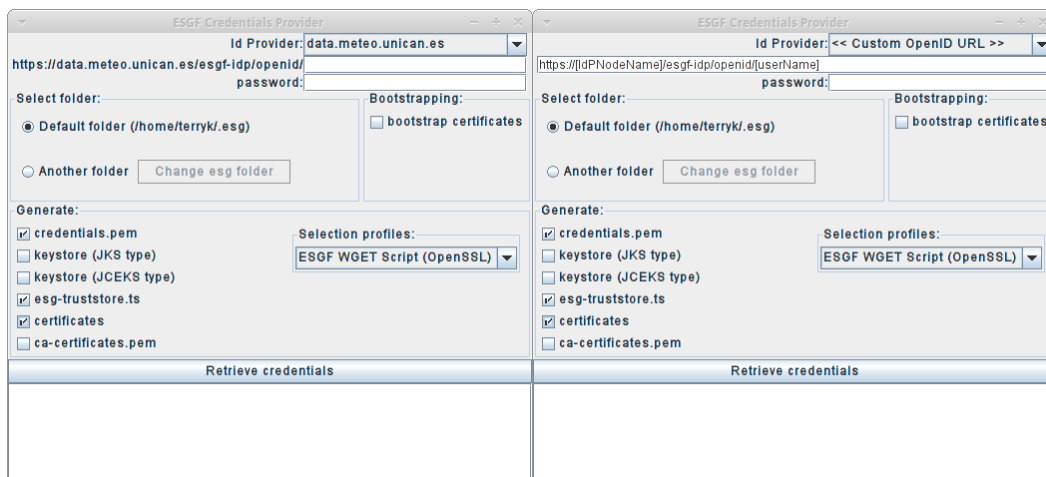
JDK or OpenJDK 6 and upper versions

### Download

Download the jar -> [getESGFCredentials-0.1.jar](#) 476.4 KB new

[Other versions..](#)

### Run it



Go to download folder:

- In Windows:
  - Open ESGFToolsUI-v0.8.jar
- Command-line interpreter:

```
java -jar ESGFToolsUI-v0.8.jar
```

## Command line UI Guide

Command line help

```
$ java -jar getESGFCredentials-0.1.jar --help
```

Basic usage

```
esgf-getcredentials --openid <openid> [other options]
```

Summary of **options**

Option	Description
--help	Display this help message
--openid <openid>	OpenID URL
--folder <folder>	Folder to save generated files
--bootstrap-certificates	Generate bootstrap certificates
--generate <options>	Generate files
--selection-profiles <profile>	Selection profile
--retrieve-credentials	Retrieve credentials

}}}

To view specific use cases -->

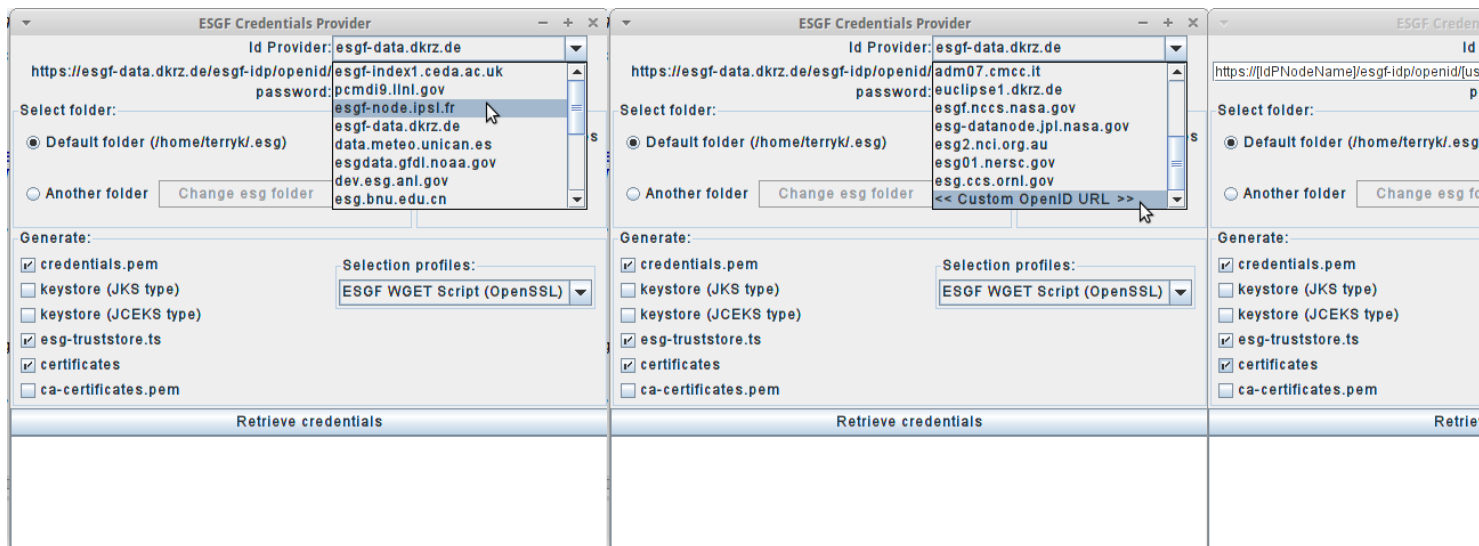
## Graphic UI Guide

- In Windows:
  - Open ESGFToolsUI-v0.8.jar
- Command-line interpreter:

```
java -jar ESGFToolsUI-v0.8.jar
```

## Setting user

You can select your IdP provider in the top drop-down list. If your IdP provider isn't in the list of providers. Select "Custom OpenID URL", with this option the GUI interface change to be able write OpenID URL's



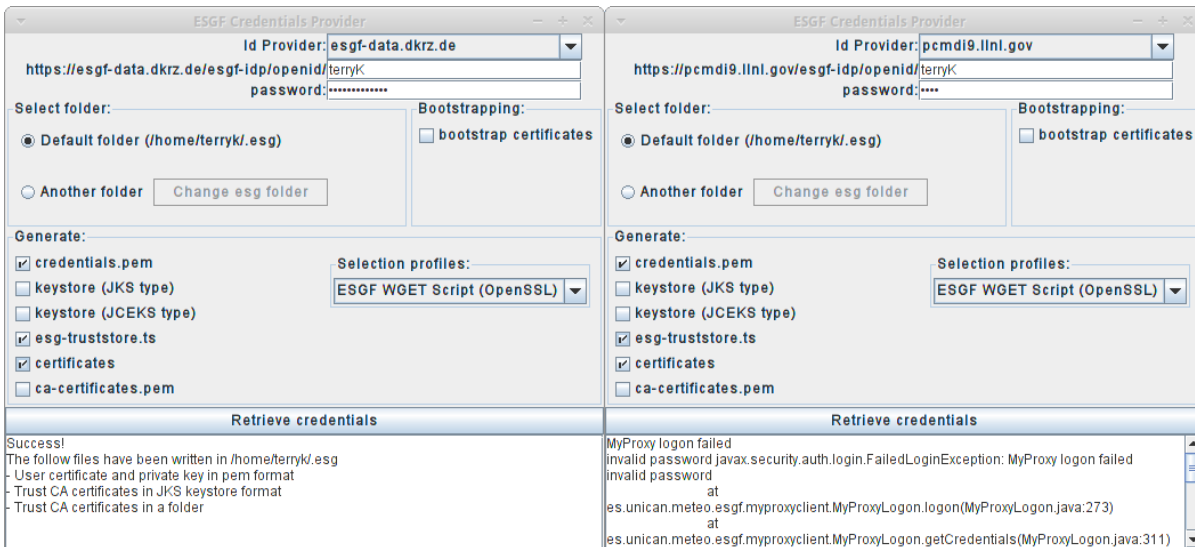
## Setting output files

You can select in "Generate" section what output files will be generated in the output folder.

credentials.pem	It's a pem file that contains the x509 user certificate and the RSA private key
keystore (JKS type)	It's a keystore in format JKS which is build with user cert, cert chain and private key
keystore (JCEKS type)	It's a keystore in format JCEKS which is build with user cert, cert chain and private key
esgf-truststore.ts	CA's certificates in keystore in format JKS
certificates	CA's certificate files and policy files in a folder
ca-certificates.pem	CA's certificates in pem format

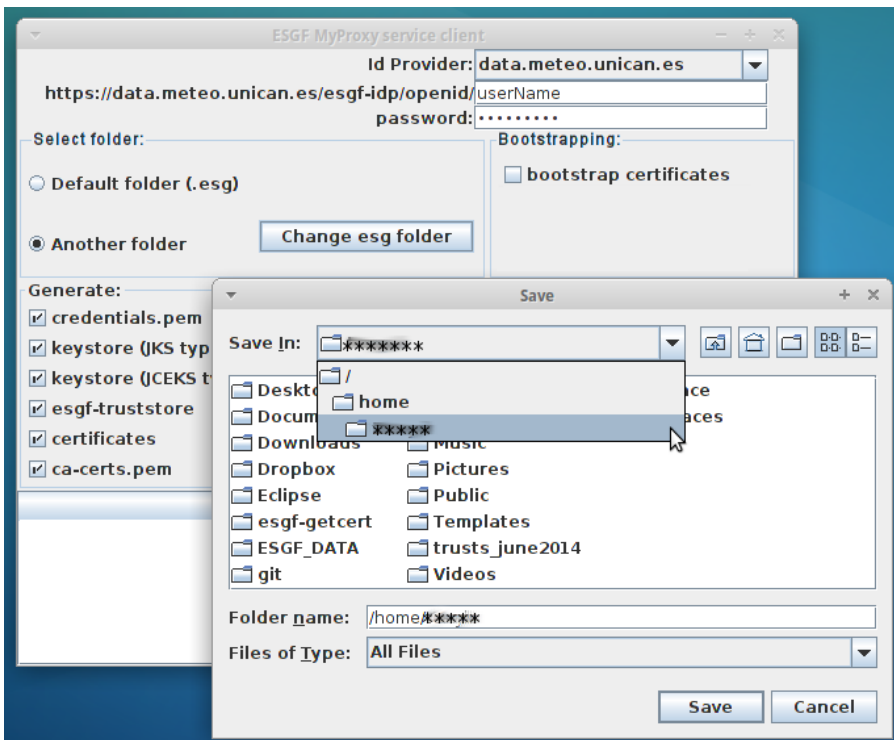
## Retrieve credentials

Click on "retrieve credentials" button. If all goes well a success message is shown. However, if some error happens then the Exception is showed



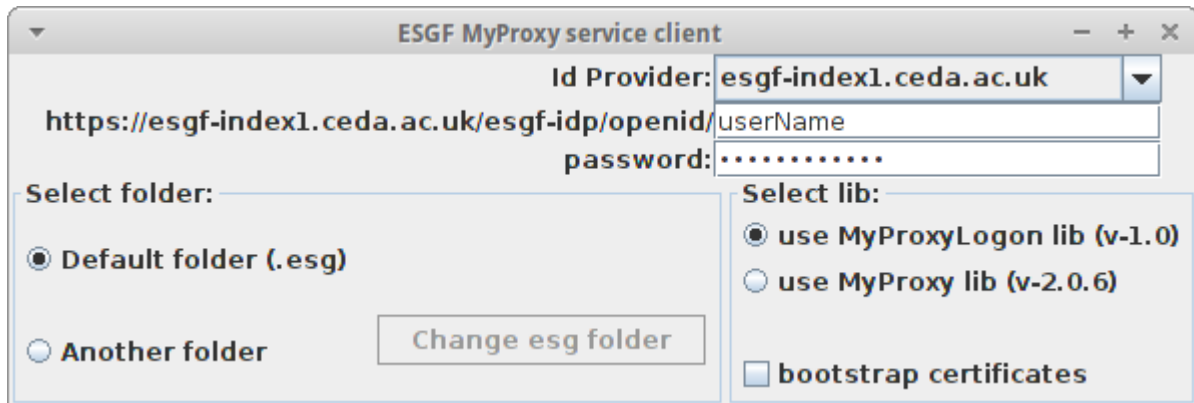
**Advanced options**

- 1. You can bootstrap the certificates. For that, select the check box "bootstrap certificates" in "Select Lib" section
- 1. You can change the output folder. The default is \$USER\_HOME/.esg



- 1. You can download a multilib myproxy version to select it in the "Select Lib" section
  - MyProxyLogon lib v1.0
  - MyProxy lib v2.0.6

MultiLib jar -> [getESGFCredentialsMultLib-0.1.jar](#) 2.3 MB new



## Specific use cases

### Aria2

#### 1. Get a metalink of ESGF Files

- Download this metalink file -> [example\\_metalink](#)

For more info, ESGFToolsUI generates metalinks of ESGF files: <https://meteo.unican.es/trac/wiki/ESGFToolsUI#ExporttoMetalink>

#### 1. Retrieve ESGF credentials in \$HOME/.esg

```
java -jar getESGFCredentials-0.1.jar --openid <openid> --password <password> --credentials --cacertspem
```

Run aria2c with credentials and **example\_metalink**

```
aria2c --private-key=$USER_HOME/.esg/credentials.pem --certificate=$HOME/.esg/credentials.pem --check-certificate=true
```

### cURL

#### 1. Retrieve ESGF credentials in current directory

```
java -jar getESGFCredentials-0.1.jar --openid <your_openid> --password <your_password> --credentials --cacertspem --outp
```

Use curl to download the file

example\_file:

[http://wdcc-esgf.dkrz.de:8080/ESGF/fileServer/cmip5/output1/IPSL/IPSL-CM5A-LR/esmrcp85/6hr/atmos/6hrPlev/r1i1p1/v20120114/ta/ta\\_6hrPlev\\_IPSL-CM5A-LR](http://wdcc-esgf.dkrz.de:8080/ESGF/fileServer/cmip5/output1/IPSL/IPSL-CM5A-LR/esmrcp85/6hr/atmos/6hrPlev/r1i1p1/v20120114/ta/ta_6hrPlev_IPSL-CM5A-LR)

```
curl -L -C - --cookie-jar curl-cookie --cookie curl-cookie --cert credentials.pem --cacert ca-certificates.pem -O http:
```

#### • Explanation of cURL options:

- L** (**L**/--location) If the server reports that the requested page has a different location let curl attempt to reattempt the get on the new place
- C <offset>** (**C**/--continue-at) to continue/Resume a previous file transfer at the given offset. "-C -" is used to tell curl to automatically find out where/how to resume the transfer.
- cookie-jar <cookie-name>** (**c**/--cookie-jar) to write cookies (cookies are generated after the handshake)
- cookie <cookie-name>** (**b**/--cookie) to load cookies from file
- cert <certfile>** (**E**/--cert) to use the specified certificate file when getting a file with HTTPS. The certificate must be in PEM format. Certificate file must content user certificate and private key.
- cacert <cacertfile>** to use the specified certificate file to verify the peer. The file may contain multiple CA certificates. The certificate(s) must be in PEM format.
- O** (**O**/--remote-name) to write output to a local file named like the remote file we get. You can use (**o**/--output <file-name> option) to specify the name of the file.

If this option is used twice, the second will again disable location following.

## WGET (OPENSSL)

1. Retrieve ESGF credentials in \$HOME/.esg

```
java -jar getESGFCredentials-0.1.jar --openid <openid> --password <password> --credentials --cacerts
```

Wget file

example\_file:

[http://wdcc-esgf.dkrz.de:8080/ESGF/fileServer/cmip5/output1/IPSL/IPSL-CM5A-LR/esmrcp85/6hr/atmos/6hrPlev/r1i1p1/v20120114/ta/ta\\_6hrPlev\\_IPSL-CM5A-LR](http://wdcc-esgf.dkrz.de:8080/ESGF/fileServer/cmip5/output1/IPSL/IPSL-CM5A-LR/esmrcp85/6hr/atmos/6hrPlev/r1i1p1/v20120114/ta/ta_6hrPlev_IPSL-CM5A-LR)

```
wget -c --ca-directory=$HOME/.esg/certificates --certificate=$HOME/.esg/credentials.pem --private-key=$HOME/.esg/creden
```

## WGET (GNU TLS)

1. Retrieve ESGF credentials in \$HOME/.esg

```
java -jar getESGFCredentials-0.1.jar --openid <openid> --password <password> --credentials --cacertspem
```

Wget file

example\_file:

[http://wdcc-esgf.dkrz.de:8080/ESGF/fileServer/cmip5/output1/IPSL/IPSL-CM5A-LR/esmrcp85/6hr/atmos/6hrPlev/r1i1p1/v20120114/ta/ta\\_6hrPlev\\_IPSL-CM5A-LR](http://wdcc-esgf.dkrz.de:8080/ESGF/fileServer/cmip5/output1/IPSL/IPSL-CM5A-LR/esmrcp85/6hr/atmos/6hrPlev/r1i1p1/v20120114/ta/ta_6hrPlev_IPSL-CM5A-LR)

```
wget -c --certificate=$HOME/.esg/credentials.pem --private-key=$HOME/.esg/credentials.pem --save-cookies=$HOME/.esg/cool
```

## ToolsUI

1. Retrieve ESGF credentials in \$HOME/.esg

```
java -jar getESGFCredentials-0.1.jar --openid <openid> --password <password> --keystorejks --cacertsjks
```

1. Start ToolsUI

```
java -Dkeystore=$HOME/keystore_jks.ks -Dkeystorepassword=changeit -Dtruststore=$HOME/.esg/esg-truststore.ts -Dtruststore
```

1. This link is a ESGF OPENDAP resource

[http://data.meteo.unican.es/thredds/dodsC/esg\\_cordexnoncommercial/cordex/output/EUR-22/UCAN/ECMWF-ERAINT/evaluation/r1i1p1/UCAN-WRF331G/v02/3/](http://data.meteo.unican.es/thredds/dodsC/esg_cordexnoncommercial/cordex/output/EUR-22/UCAN/ECMWF-ERAINT/evaluation/r1i1p1/UCAN-WRF331G/v02/3/)

For more: <http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/reference/Session.html>

## ESGF WGET Script (Linux)

```
java -jar getESGFCredentials-0.0.2.jar -o <openid> -p <password> --credentials --cacerts --cacertsjks
```

## ESGF WGET Script (cygwin)

```
java -jar getESGFCredentials-0.0.2.jar -o <openid> -p <password> --credentials --cacertspem --cacertsjks
```

## NETCDF DAP

<https://www.unidata.ucar.edu/software/netcdf/docs/netcdf/DAP-Support.html>

```
java -jar getESGFCredentials-0.0.2.jar -o <openid> -p <password> --credentials --cacerts --cacertspem --output <path>
```

## Developers Guide

### Github

[?https://github.com/SantanderMetGroup/esgf-getcredentials](https://github.com/SantanderMetGroup/esgf-getcredentials)

### Architecture

### See Also

- [ESGFToolsUI - a desktop client for ?ESGF services](#)