



03/2022

# AGATA data download from GRID

*AGATA Data distribution and re-processing Team*

---

---

This document provides a guide to help the users to download AGATA data from the grid.

---

---

The last version of this document, provided by the AGATA Data distribution and re-processing team, can be found on ATRIUM (<https://atrium.in2p3.fr/>). Just click on WORKSPACE (ARBORESCENCE) and follow the tree on the left side (see the following picture) to reach the DataProcessing entry point.

ATRIUM HOME ARBORESCENCE RECHERCHE TICKET SUPPORT ME CONNECTER Guest Document, utilisateur, groupe

Atrium Zone de Diffusion Grand Public **Projet AGATA** DataProcessing

## DataProcessing

Documents related to online/offline processing of the Data

Contenu Résumé Historique

Documents / page: 30

Type	Titre	Atrium ID	Créé le	Auteur	Modifié le	Modifié par	Vrs. Obs.	Etat
	WatcherUsersGuide	ATRIUM-3982	5 mars 2015	Olivier STEZOWSKI	21 mai 2015	stezow@ipnl.in2p3.fr	3.0	Validé
	GwUserGuide.odt	ATRIUM-3884	27 févr. 2015	Olivier STEZOWSKI	27 févr. 2015	stezow@ipnl.in2p3.fr	1.0	Validé
	Cookbook.odt	ATRIUM-3883	27 févr. 2015	Olivier STEZOWSKI	27 févr. 2015	stezow@ipnl.in2p3.fr	1.0	Validé

<b>Preparation</b> .....	3
1) Get a User Interface account and its prerequisites.....	3
2) Get a Grid certificate.....	3
3) Join the AGATA Virtual Organization .....	4
4) Some basic information on the data management on the grid .....	4
<b>Script user guide</b> .....	5
5) Download the python script.....	5
6) Presentation of the different options on the script .....	5
--help:.....	6
--new_proxy:.....	6
--proxy_status: .....	6
--from_LYON: .....	6
--from_CNAF: .....	6
--show_conf: .....	6
--ls_dir:.....	7
--input_dir: .....	7
--output_dir: .....	7
--exc: .....	8
--inc:.....	8
--build_list:.....	8
--bring_online: .....	8
This command will launch the download. ....	9
By default, a checksum is done after each downloaded file. To remove this operation, using the --nochecksum option. ....	9
To overwrite the files already downloaded, use the --overwrite option.....	9
7) Example: download a folder on a local computer.....	9

***People involved in this document:***

J. Dudouet

NOTE: Please for any comment/question/suggestion, use the forum: <http://www.agata.org/forum>

# Preparation

The Grid is a cooperation of many different clusters and research organizations, and as such, there is no centralized user management. Yet, there must be a way for the system to identify you and your work. This is why **Grid certificates** and **Virtual Organizations** (VOs) are introduced.

Your digital identity starts with a private key. **Only you** are allowed to know the contents of this key. Next, you need a Grid certificate, which is issued by a Certificate Authority (CA). The Grid certificate contains your name and your organization, and it says that the person who owns the private key is really the person mentioned, and that this is certified by the Certificate Authority.

Now this is your identity. Big international collaborations do not want to deal with every user individually. Instead, users become part of Virtual Organizations. To give an analogy, the Grid certificate provides authentication (identity, e.g., like a passport) and the VO provides authorization (approval, e.g., like a visa).

In order to access to the Grid, you have to make three essential steps:

1. **Get a User Interface (UI)** account, so that you can interact with the Grid.
2. **Get a Grid certificate**, so that you can be identified on the Grid.
3. **Join the AGATA Virtual Organization (VO)**, so that you can access to the Grid.

The UI account will provide you with the proper environment to interact with the Grid. The Grid certificate is required to authorize you for using the Grid. Finally, the membership is based on your research domain and determines which resources you can use.

## 1) Get a User Interface account and its prerequisites

The User Interface (UI) account will provide you with the environment to interact with the Grid. It is your access point to the Grid. You log in to a UI machine via SSH. On this computer you can, amongst others, do the following things: access Grid resources, create proxies, submit jobs, compile programs or prototype your application.

To request for a UI account, please refer to your local computer service. In the context of this script based on **python3** (version 3.6 or newer), the UI needs to include the **gfal2 library and the gfal2-python module** (<https://dmc-docs.web.cern.ch/dmc-docs/gfal2/gfal2.html>).

If you are still working on an old version of the UI, the old script using python2 and gfal is still available (see part 5)

Please note that the UI is simply a Linux machine and working on it requires some familiarity with Linux commands and remote access methods.

## 2) Get a Grid certificate

Grid certificates are supplied by a Certificate Authority (CA). The procedure for acquiring a grid certificate is different at each university, laboratory etc. If you have some active grid users or specialists at your home institute, then ask them for advice on how to get a grid certificate. Note that **AGATA is NOT** issuing any grid certificates.

Once you get your grid certificate, you need first to load it in your browser, this will allow you to request to join the AGATA Virtual Organisation (VO). You can then download from your browser your certificate file. On Firefox: Preferences menu, Advanced settings, Certificates, View Certificates, Your Certificates, Select the grid certificate, Backup. You will obtain a certificate file in the PKCS12 format, with a .p12 extension. It then needs to be converted in the PEM format using the openssl command. Converting PCSK12 to PEM is done as

follows: (assuming your certificate name is myCertif.p12)

```
ls -l
total 8
-rw-rw-r-- 1 dudouet dudouet 6113 mars 12 13:32 myCertif.p12

openssl pkcs12 -nocerts -in myCertif2012.p12 -out userkey.pem
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
openssl pkcs12 -clcerts -nokeys -in myCertif.p12 -out usercert.pem
Enter Import Password:
MAC verified OK

ls -l
total 16
-rw-rw-r-- 1 dudouet dudouet 6113 mars 12 13:32 myCertif.p12
-rw-rw-r-- 1 dudouet dudouet 1959 juil. 25 22:32 usercert.pem
-rw-rw-r-- 1 dudouet dudouet 2006 juil. 25 22:31 userkey.pem
```

In your home directory (\$HOME) create the directory .globus/ and copy there your certificate in PEM format:

```
mkdir $HOME/.globus
cp usercert.pem userkey.pem $HOME/.globus/
```

Change then the permissions:

```
cd $HOME/.globus
chmod 444 usercert.pem
chmod 400 userkey.pem
ls -l
total 8
-r--r--r-- 1 dudouet dudouet 1959 juil. 25 22:37 usercert.pem
-r----- 1 dudouet dudouet 2006 juil. 25 22:37 userkey.pem
```

### 3) Join the AGATA Virtual Organization

A Virtual Organization (VO) is a group of geographically distributed people that have common objectives and that are using shared Grid resources to achieve them. Every Grid user is a member of one or more VOs.

In practice your VO membership determines to which resources (compute and storage) you have access to. You are eligible to register for a VO **only once you get a valid certificate**. To join the AGATA VO, you must have the grid certificate successfully installed in your browser. Then click on this link: AGATA VOMS Server (<https://cclcgvomsli01.in2p3.fr:8443/voms/vo.agata.org>)

For any further information and help about the AVO, please contact Johan Nyberg ([johan.nyberg@physics.uu.se](mailto:johan.nyberg@physics.uu.se))

### 4) Some basic information on the data management on the grid

The dCache storage consists of magnetic tape storage and hard disk storage. Before it can be used, the data stored on magnetic tape has to be copied to a hard drive. This action is called ‘Staging’ files or ‘bringing a file online’. When a file status is “NEARLINE”, this means that the file is on **tape only**. On the contrary, the status ‘ONLINE’ means that the file is on **disk only**.

Before downloading data, a ‘bring\_online’ operation is necessary to copy all the files from the tapes to the disks. The disk pool where your files are staged has limited capacity and is only meant for data that a user wants to process. When you stage a file, you set a pin lifetime. The pin lifetime here is set to 1 week, but note that this counts from the moment you submit the request independent to the actual time that the files are on disk. The file will not be purged until the pin lifetime has expired. Then the data may be purged from disk, as soon as the space is required for new stage requests. When the disk copy has been purged, it has to be staged again in order to be processed on a Worker Node.

When a pool group is full with pinned files, staging is paused. Stage requests will just wait until pin lifetimes for other files expire. dCache will then use the released space to stage more files until the pool group is full again. **When you are done with your processing, we recommend you release (or unpin) all the files that you don't need any more. It is an optional action, but helps a lot with the effective system usage.**

The script automatically unpins the files after being downloaded, but if a list of file has been brought online but not downloaded, please use the release command to unpin the files.

## Script user guide

### 5) Download the python script

The python script dedicated to download AGATA data is accessible at the following link:  
[https://gitlab.in2p3.fr/IPNL\\_GAMMA/scripts/raw/master/Grid/GridDataSync.py](https://gitlab.in2p3.fr/IPNL_GAMMA/scripts/raw/master/Grid/GridDataSync.py)

The auto-completion script (using TAB), is accessible at the following link:  
[https://gitlab.in2p3.fr/IPNL\\_GAMMA/scripts/raw/master/Grid/do\\_grid\\_completion.sh](https://gitlab.in2p3.fr/IPNL_GAMMA/scripts/raw/master/Grid/do_grid_completion.sh)

Using a Linux computer, it can be do as following:

```
wget https://gitlab.in2p3.fr/IPNL_GAMMA/scripts/raw/master/Grid/python3-gfal2/GridDataSync.py
wget https://gitlab.in2p3.fr/IPNL_GAMMA/scripts/raw/master/Grid/python3-gfal2/do_grid_completion.sh
```

In case you are using the old version of the UI, not compatible with gfal2, use the following:

```
wget https://gitlab.in2p3.fr/IPNL_GAMMA/scripts/raw/master/Grid/python2-gfal/GridDataSync.py
wget https://gitlab.in2p3.fr/IPNL_GAMMA/scripts/raw/master/Grid/python2-gfal/do_grid_completion.sh
```

The auto-completion script needs to be moved in your home (\$HOME) folder, and its execution added to your `“.bashrc”` file:

```
mv do_grid_completion.sh ${HOME}/
echo "source ${HOME}/do_grid_completion.sh" >> ${HOME}/.bashrc
source ${HOME}/.bashrc
```

The auto-completion script allows, by pressing the TAB command, to auto-complete the possible options of the script

### 6) Presentation of the different options on the script

To show on the terminal the available options, use the `--help` option:

```
./GridDataSync.py --help
Usage: GridDataSync.py [options]

Browse and download AGATA data from the grid

Options:
  -h, --help            show this help message and exit
  --new_proxy           create a new proxy
  --proxy_status       print the proxy status
  --from_LYON          download data from CC Lyon (default)
  --from_CNAF         download data from Bologna
  --show_conf         show the current configuration (paths, patterns)
  --ls_dir            list the content of the given folder
  --input_dir=path    copy grid data from distant path
  --output_dir=path   copy grid data into local path
  --exc=pattern       exclude patterns, separated by ":", will skip all files containing
                    exc patterns (none to reset)
  --inc=pattern       include patterns, separated by ":", will skip all files not
                    containing inc patterns (none to reset)
                    (check https://regexone.com/references/python regex format)
  --build_list        build the list of files to be downloaded (mandatory before start)
```

```

--check_status    check the status of the files to be downloaded (locality, downloaded...)
--verbose        increase the verbosity
--bring_online    move files from tape to disks (make the copy of files faster)
--start          launches the download of the files from the grid
--force          force the download of offline files (much slower)
--nochecksum     remove the checksum on each downloaded file
--overwrite      Overwrite the already downloaded files
--release        release all files from disk

```

In the following, the different options will be presented:

### **--help:**

Print the above message. It will be also printed if no option is given

### **--new\_proxy:**

This command will create a new proxy, valid during 72 hours (this require your certificate password)

```

./GridDataSync.py --new_proxy
Enter GRID pass phrase for this identity:
Contacting cclcgvomsl01.in2p3.fr:15007 [/O=GRID-FR/C=FR/O=CNRS/OU=CC-
IN2P3/CN=cclcgvomsl01.in2p3.fr] "vo.agata.org"...
Remote VOMS server contacted successfully.

Created proxy in /tmp/x509up_u2471.

Your proxy is valid until Sat Mar 19 14:59:40 CET 2022

```

To verify that your proxy is valid, use again the **--proxy\_status** command, the time left should now be around 72 hours.

### **--proxy\_status:**

This command will show the status of your proxy, including the remaining time to use it:

```

./GridDataSync.py --proxy_status
subject   : /O=GRID-FR/C=FR/O=CNRS/OU=IPNL/CN=Jeremie Dudouet/CN=1858711231
issuer    : /O=GRID-FR/C=FR/O=CNRS/OU=IPNL/CN=Jeremie Dudouet
identity  : /O=GRID-FR/C=FR/O=CNRS/OU=IPNL/CN=Jeremie Dudouet
type      : RFC3820 compliant impersonation proxy
strength  : 2048
path      : /tmp/x509up_u2471
timeleft  : 71:57:52
key usage : Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment, Key Agreement
=== VO vo.agata.org extension information ===
VO       : vo.agata.org
subject   : /O=GRID-FR/C=FR/O=CNRS/OU=IPNL/CN=Jeremie Dudouet
issuer    : /O=GRID-FR/C=FR/O=CNRS/OU=CC-IN2P3/CN=cclcgvomsl01.in2p3.fr
attribute : /vo.agata.org/Role=NULL/Capability=NULL
timeleft  : 71:57:52
uri       : cclcgvomsl01.in2p3.fr:15007

```

### **--from\_LYON:**

To download the data from the Lyon Tier1 site (this is the default value)

### **--from\_CNAF:**

To download the data from the Bologna Tier1 sit

### **--show\_conf:**

To store the different parameters, a configuration file is automatically created at the first execution of the program and updated as a function of the parametrization. To see the status of the configuration file, use this

command, or check the file “Grid.conf”:

```
./GridDataSync.py --show_conf
*****
** GridDataSync configuration **
*****
SERVER          : srm://storm-fe-archive.cr.cnaf.infn.it:8444/srm/managerv2?SFN=
BASE_DIR_ON_GRID : /agata/
INPUTDIR        : e806/e806/run_0204.dat.05-07-21_00h12m13s
OUTPUTDIR       : /gridgroup/matnuc/scratch/
```

### **--ls\_dir:**

This allows to show the arborescence of a folder from the grid. Different cases are possible:

- without argument: this will print the arborescence of the base directory of the server on which you are connected:

```
./GridDataSync.py --ls_dir
-- List content of folder: srm://ccsrm02.in2p3.fr:8443/srm/man-
agerv2?SFN=/pnfs/in2p3.fr/data/agata/
Folder 02 Sep 2010 lschwarz
Folder 08 Sep 2010 pietro_test
Folder 21 Sep 2010 2010_week19
Folder 22 Oct 2010 Milano
Folder 25 Oct 2010 2010_week28
Folder 24 Mar 2011 2010_week48
 80.6 MB 31 Mar 2011 test_small_file_johan.tar
Folder 01 Apr 2011 252Cf
Folder 22 Feb 2013 generated
Folder 26 Apr 2013 kaci-test
Folder 07 Oct 2013 ycalas
```

- By giving as argument a sub folder:

```
./GridDataSync.py --ls_dir e680/e680
-- List content of folder: srm://ccsrm02.in2p3.fr:8443/srm/man-
agerv2?SFN=/pnfs/in2p3.fr/data/agata/e680/e680
Folder 09 Jun 2015 NarvalAGATAsolo
Folder 09 Jun 2015 Replay
Folder 09 Jun 2015 ReplayFromAnalysisServer
Folder 09 Jun 2015 run_0001.dat.07-05-15_14h17m05s
Folder 09 Jun 2015 run_0002.dat.07-05-15_14h22m27s
Folder 09 Jun 2015 run_0003.dat.11-05-15_09h55m42s
Folder 09 Jun 2015 run_0004.dat.11-05-15_10h06m11s
Folder 09 Jun 2015 run_0005.dat.11-05-15_11h05m02s
Folder 09 Jun 2015 run_0006.dat.11-05-15_11h58m20s
Folder 09 Jun 2015 run_0007.dat.11-05-15_12h16m13s
Folder 09 Jun 2015 run_0008.dat.11-05-15_12h26m07s
Folder 09 Jun 2015 run_0009.dat.11-05-15_12h34m52s
Folder 09 Jun 2015 run_0010.dat.11-05-15_13h04m12s
Folder 09 Jun 2015 run_0011.dat.11-05-15_13h55m12s
Folder 09 Jun 2015 run_0012.dat.11-05-15_14h20m10s
Folder 09 Jun 2015 run_0013.dat.11-05-15_14h41m16s
Folder 09 Jun 2015 run_0015.dat.11-05-15_15h04m29s
```

This **--ls\_dir** command will help you to move through the grid arborescence in order to find the folder you want to download.

### **--input\_dir:**

Once you have found the folder you want to download, you will set it as `input_dir` with this option.

### **--output\_dir:**

This command is used to give the local folder where you want to download the data. Data from the grid will be stored in the same arborescence in your local folder “`output_dir`” than on the grid.

### **--exc:**

This option allows to define exclude patterns to limit the type of data you want to download. The exclude patterns are using the regular expression norm of python. You can find more information on this link: <https://regexone.com/references/python>.

When you define exclude patterns, this will exclude, when building the list of files (see below) all file names that contains this (or these) patterns. If more than one pattern is asked, it needs to be separated by “.”

#### Example:

```
./GridDataSync.py --exc *.*tar:*.cdat.*
```

This command will exclude all .tar and cdat files.

To remove the exclude patterns, you need to use this **--exc** option with the “**none**” argument

### **--inc:**

Similarly, this **--inc** option allows to specify include patterns, which means patterns that are necessarily present in the file name to be added to the list of files. The same procedure is applied than for the **--exc** option

#### Example:

```
./GridDataSync.py --inc */03./.*
```

This command will only save files for the crystals starting with 03 (03A, 03B, 03C)

Here also, to remove the include patterns, you need to use this **--inc** option with the “**none**” argument

Include and exclude patterns are recalled in the **--show\_conf** option.

### **--build\_list:**

Once the `input_dir` has been set, it is necessary to build the list of all the files of this folder. This is done using this option. This operation will build a catalog “*grid\_summary.csv*”. This file allows you to follow the status of the files (urls, tape or disk locality, download status)

### **--bring\_online:**

This mandatory operation consists in copying the files from tapes to disks (see part 4). The **--bring\_online** command will launch a background task on the grid server to copy all your files on the disks. You can wait during this process to see its evolution, or closing it and letting the staging processing in background. For large number of files, the staging needs to be done by batch of 1000 files with 1 minute interval time to prevent overloading the staging namespace server.

### **--check\_status:**

This command allows to check the status process. It will print the locality of the files (see part 4) and the download status. The catalog will be updated to take into account the advancement of the `bring_online` operation

To have more information, the options **--verbose** can be added to this command.

#### Example:

```
./GridDataSync.py --check_status
... updating the catalog
-- 117 Files from e806/e806/run_0204.dat.05-07-21_00h12m13s in the list
==> 3 downloaded (114 remaining)
```



```
==> 114 brought online for non downloaded files (0 remaining)
```

Here we can see that all the files have been copied on disks, and 3 are already downloaded.

### --start:

This command will launch the download.

By default, a checksum is done after each downloaded file. To remove this operation, using the **--nochecksum** option.

To overwrite the files already downloaded, use the **--overwrite** option.

### --release:

This command will release all the files that have been pinned using the **--bring\_online** option. In case the files will finally not be downloaded for example.

## 7) Example: download a folder on a local computer

In this example, we will download the configuration files and the traces (cdat files) for crystals 03ABC in the run 003 of the e680 experiment from the Lyon grid server:

1. Create a new proxy if necessary

```
./GridDataSync.py --new_proxy
```

2. Move in the arborescence of the grid up to the good folder

```
./GridDataSync.py --ls_dir e680
-- List content of folder: srm://ccsrm02.in2p3.fr:8443/srm/man-
agerv2?SFN=/pnfs/in2p3.fr/data/agata/e680
Folder    09 Jun 2015 e680
Folder    09 Jun 2015 ReadATCA
Folder    15 Jun 2015 e680_NoTraces
...

./GridDataSync.py --ls_dir e680/e680
-- List content of folder: srm://ccsrm02.in2p3.fr:8443/srm/man-
agerv2?SFN=/pnfs/in2p3.fr/data/agata/e680/e680
Folder    09 Jun 2015 NarvalAGATAsolo
Folder    09 Jun 2015 Replay
Folder    09 Jun 2015 ReplayFromAnalysisServer
Folder    09 Jun 2015 run_0001.dat.07-05-15_14h17m05s
Folder    09 Jun 2015 run_0002.dat.07-05-15_14h22m27s
Folder    09 Jun 2015 run_0003.dat.11-05-15_09h55m42s
...
```

3. Define the folder as input

```
./GridDataSync.py --input_dir e680/e680/run_0003.dat.11-05-15_09h55m42s
```

4. Define the output folder

```
./GridDataSync.py --output /gridgroup/matnuc/scratch
```

5. Define the include and exclude patterns

```
./GridDataSync.py --exc ^.*.adf$
./GridDataSync.py --inc ^.*03./.*$
```

Here: “. \*.adf” will exclude adf files, and “. \*/03./.\*” will include 03ABC crystals

## 6. Build the list of files

```
./GridDataSync.py --build_list
==> All the files have been released from disks
=> adding: 80.0 kB e680/e680/run_0003.dat.11-05-15_09h55m42s/Conf/03A/SRM_AGATA_small_files.tar
=> adding: 80.0 kB e680/e680/run_0003.dat.11-05-15_09h55m42s/Conf/03B/SRM_AGATA_small_files.tar
=> adding: 80.0 kB e680/e680/run_0003.dat.11-05-15_09h55m42s/Conf/03C/SRM_AGATA_small_files.tar
=> adding: 1020.0 MB e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03A/SRM_AGATA_event_mez-
zdata.cdat.0000
=> adding: 64.5 MB e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03A/SRM_AGATA_small_files.tar
=> adding: 895.0 MB e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03B/SRM_AGATA_event_mez-
zdata.cdat.0000
=> adding: 68.8 MB e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03B/SRM_AGATA_small_files.tar
=> adding: 890.0 MB e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03C/SRM_AGATA_event_mez-
zdata.cdat.0000
=> adding: 40.4 MB e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03C/SRM_AGATA_small_files.tar
-- 9 Files from e680/e680/run_0003.dat.11-05-15_09h55m42s added to the list of files to be down-
loaded
--> Total files size: 2.9 GB
```

## 7. Copy the files from tapes to disks

```
./GridDataSync.py --bring_online
... Start the copy of files from tapes to disks ...
-> press CTRL+C to skip the display (the staging operation will keep working in background
** Staging launched for 9/9 files **
[|] ... 4 files staged over 9, 5 remaining ... sopping the dispay, the staging continues in
background...
```

## 8. Check the locality of the files

Once the staging has been launched, you need to wait few minutes and test with the **--check\_status** command until the first files are in the **ONLINE\_AND\_NEARLINE** status

```
./GridDataSync.py --check_status --verbose
... updating the catalog
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Conf/03A/SRM_AGATA_small_files.tar
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Conf/03B/SRM_AGATA_small_files.tar
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Conf/03C/SRM_AGATA_small_files.tar
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03A/SRM_AGATA_event_mez-
zdata.cdat.0000
NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03A/SRM_AGATA_small_files.tar
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03B/SRM_AGATA_event_mez-
zdata.cdat.0000
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03B/SRM_AGATA_small_files.tar
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03C/SRM_AGATA_event_mez-
zdata.cdat.0000
ONLINE_AND_NEARLINE : e680/e680/run_0003.dat.11-05-15_09h55m42s/Data/03C/SRM_AGATA_small_files.tar
-- 9 Files from e680/e680/run_0003.dat.11-05-15_09h55m42s in the list
==> 0 downloaded (9 remaining)
==> 8 brought online for non downloaded files (1 remaining)
```

Here, we can see that one file is still offline. This is not mandatory to wait that all the files are ONLINE to start the download. The missing files will be downloaded in a secondary step.

## 9. Download the files

Once its ready, start the download with the **--start** command:

```
./GridDataSync.py --start
... updating the catalog
-- 9 Files from e680/e680/run_0003.dat.11-05-15_09h55m42s in the list
==> 0 downloaded (9 remaining)
```

```
==> 9 brought online for non downloaded files (0 remaining)
...starting to download the 9 requested files...
Copied files: 9/9, current: 40.4 MB, total: 2.9 GB/ 2.9 GB, rate= 36.8 MB/s
-- 9 Files have been to be downloaded in 1min, 0s,
... updating the catalog
-- 9 Files from e680/e680/run_0003.dat.11-05-15_09h55m42s in the list
==> All the files have been downloaded
```