

How to use Grid resources in batch jobs

To use remote grid resources in a batch job, the batch job has to be 'attached' to the users [grid ID](#).

As the passport of the grid world is the grid proxy based on the X509 scheme, every job (that is supposed to access resources at remote grid sites) has to be equipped with the proxy.

Environment Preparation

It is assumed here, that users have grid knowledge and know how to obtain a grid proxy, so we will be a bit short here.

Please setup the necessary Grid tools in your environment. Most experiments provide the tools in their setups.

ATLAS setup

For example, **ATLAS** you can setup the grid tools managed in the ATALS repository with

```
lsetup emi
```

which makes voms-proxy-init etc. available in your current session.

Band Aid Option

We provide a script to setup a generic grid environment with

```
source /cvmfs/grid.desy.de/etc/profile.d/grid-ui-env.sh
```

While it should setup the generic tools, this might be not tailored to/by your experiments. So ideally inquire with your group, how they setup their environments for an optimal experience.

Getting a proxy

After setting up your environment for grid jobs, you can obtain a proxy with tools like

```
voms-proxy-init --rfc --voms FOOVO {-out /NON/STANDARD/PATH/YOURPROXYFILE.PEM}
```

or

```
arcproxy --cert=/PATH/TO/usercert.pem --key=/PATH/TO/userkey.pem --voms=FOOVO
```

Most sites only support proxies with a limited lifetime (assume ~24h to avoid unlucky surprises). So you will not be able to create a proxy with a month lifetime, submit 100k jobs and go for vacation assuming that everything will have run fine in the meantime.

Grid Identity

This proxy identifies you and all your permissions in the grid!

Be sure what you do! When accessing grid resources locally as well as remote you might have a lot of power!

Keep it safe! Anyone getting a hand on your proxy can impersonate you!

Where to find the proxy

When requesting a new proxy with the aforementioned grid tools, the proxy is put as a X509 certificate file into the file system - by default it is placed under `/tmp/x509up_u{YOURUSERID}`.

An environment variable is assumed to point to the path of the proxy - if it is not set your environment after creating a proxy, set it to the corresponding path of your proxy (you might need to check your shell's syntax)

```
export X509_USER_PROXY=/PATH/TO/YOUR/FRESH/YOURPROXYFILE.PEM
```

Reading ntuples from remote storage end points in ROOT

If you have files on a remote storage end point as EOS, you can point ROOT point to the files over the network via xrootd as protocol with a valid proxy around.

Setup your environment to include all necessary tools as ROOT and xrootd and you should be able to access remote tuples like local files.

E.g. for ATLAS

```
| lsetup xrootd rucio root
```

(note that in ATLAS case root has to be setup as last package as else there seem to be a clash of some shared libraries).

```
| root [X] TFile *mytuple =TFile::Open("root://eosatlas.cern.ch/eos/atlas/PATH/TO/MYtuple.root")
```

Take care, that your read patterns and workflows are reasonable!

If you or your jobs put to much stress on remote end points (e.g., thousands of jobs in parallel, randomly jumping from event to event in a tuple exceeding the TTree cache,...), be prepared that some admin will try to hunt you down! see also: [WAN Reads](#)

Using xrootd to copy files directly from Grid storage end points

If one knows the storage endpoint and path of a file, one can also copy a file manually.

For any 'namespace operations' one could use 'xrdfs MYSEADDRESS COMMAND PATHONMYSE, e.g., to list everything in one directory, that lives on eosatlas.cern.ch

```
| xrdfs root://eosatlas.cern.ch ls /eos/atlas/atlascerngroupdisk/phys-exotics/lpx/LQPP2018/ttauttau/v1/mc16a/DIRNAME/
```

To copy a file from there, one can use xrdcp or gfal-copy

```
| gfal-copy root://eosatlas.cern.ch/eos/atlas/atlascerngroupdisk/phys-exotics/lpx/LQPP2018/ttauttau/v1/mc16a/DIRNAME  
/FILENAME.root /tmp/
```

```
| xrdcp root://eosatlas.cern.ch/eos/atlas/atlascerngroupdisk/phys-exotics/lpx/LQPP2018/ttauttau/v1/mc16a/DIRNAME  
/FILENAME.root /tmp/
```

Watch out, what you do on the remote storage end point, check for any destructive behaviour and do not overload the network!

Getting the proxy into the jobs

A job will need a copy of the proxy file locally to access under your identity grid resources.

To do so stage the the proxy file during the job submission (in principle, you can also put the proxy file on a shared path that can be read by all jobs):

```
| > condor_job.description  
  
executable = YOURJOB.EXEC  
...  
transfer_input_files = YOURPROXYFILE.PEM, OTHER.FILE1, ANOTHER.FILE2  
...
```

this will tell Condor to transfer the proxy file during job submission and put it into the job's home directory on the batch node

Making the proxy known within the job

So that in your job any grid tools know where to find the proxy file, you have to export the path first into the jobs' environment before running anything grid related

If you have staged the proxy file to the job, the file should be available in the job's HOME directory, so that you can either set the path in the job wrapper script with

```
| export X509_USER_PROXY=${HOME}/YOURPROXYFILE.PEM
```

or tell Condor in the job description to put the environment variable already in the job's environment

```
| > condor_job.description  
...  
transfer_input_files = ...  
...  
environment = "X509_USER_PROXY=${HOME}/YOURPROXYFILE.PEM"
```

If you have placed the proxy file somewhere else, e.g., a shared path, you have to adapt the environment export accordingly.