

# Data transfers with SRM

All the SRM specifications are available here [13].

In this case, a voms-proxy is needed (see in the previous sections for details on proxy generation).

```
-bash-4.2$ voms-proxy-init --voms virgo:/virgo/virgo
Enter GRID pass phrase for this identity:
Contacting voms-01.pd.infn.it:15009 [/DC=org/DC=terena/DC=tcs/C=IT/L=Frascati/O=Istituto Nazionale di Fisica
Nucleare/CN=voms-01.pd.infn.it] "virgo"...
Remote VOMS server contacted succesfully.

Created proxy in /tmp/x509up_u10162.

Your proxy is valid until Tue Aug 18 22:45:07 CEST 2020
-bash-4.2$ voms-proxy-info --all
subject : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina arendina@infn.
it/CN=1185352064
issuer  : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina arendina@infn.
it
identity : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina
arendina@infn.it
type    : RFC3820 compliant impersonation proxy
strength : 1024
path    : /tmp/x509up_u10162
timeleft : 11:57:53
key usage : Digital Signature, Key Encipherment
=== VO virgo extension information ===
VO      : virgo
subject : /DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Andrea Rendina arendina@infn.
it
issuer  : /DC=org/DC=terena/DC=tcs/C=IT/L=Frascati/O=Istituto Nazionale di Fisica Nucleare/CN=voms-01.pd.infn.it
attribute : /virgo/virgo/Role=NULL/Capability=NULL
attribute : /virgo/Role=NULL/Capability=NULL
timeleft : 11:57:53
uri     : voms-01.pd.infn.it:15009
```

In contrast to the *gridFTP* protocol, we have to contact a StoRM frontend on the port 8444. Then the frontend communicates the request to a backend [14].

The Virtual Organizations use dedicated storm endpoint for data management and data transfer:

<https://www.cnaf.infn.it/~usersupport/StorageAreas.html>.

For example, AMS uses storm-fe-ams for the disc area (/storage/gpfs\_ams), but storm-archive to write on the buffer tape (/storage/gpfs\_archive/ams). Also, the path to read or write is not the real path on the filesystem, but all the experiments use a singular access path to the *storage area*.

Now we consider two tools for the SRM protocol: Gfal and ClientSRM.

## Gfal utils

Documentation is available here [15]. However the user can get the full list using the command `man gfal-copy`.

Most used commands are:

- **gfal-ls**: List information about the file
- **gfal-xattr**: Display attributes of a file or set them to a new value
- **gfal-cat**: Concatenate file to standard output
- **gfal-mkdir**: Create the DIRECTORY(ies), if they do not already exist
- **gfal-stat**: Display extended information about a file or directory
- **gfal-chmod**: Change the permissions of a file
- **gfal-rename**: Renames SOURCE to DESTINATION
- **gfal-sum**: Calculates the checksum of the specified file, using a specified checksum algorithm
- **gfal-rm**: Removes each specified file or directory (using the option -r, -R, --recursive: remove directories and their contents recursively)
- **gfal-save**: Reads from stdin and writes to a file until it finds EOF

These are the steps to install Gfal assuming the machine is CentOS7:

1. Enable *epel* repo:

```
curl http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm >/tmp/epel-release-latest-7.noarch.rpm
sudo rpm -ivh /tmp/epel-release-latest-7.noarch.rpm
```

## 2. Enable *egi* repo:

```
echo '\[EGI-trustanchors\]name=EGI-trustanchorsbaseurl=http://repository.egi.eu/sw/production/cas/1/current/gpgkey=http://repository.egi.eu/sw/production/cas/1/GPG-KEY-EUGridPMA-RPM-3gpgcheck=1enabled=1' | sudo tee /etc/yum.repos.d/egi.repo
```

## 3. Install several tools:

```
sudo yum install -y gfal2-util gfal2-all fetch-crl ca-policy-egi-core globus-proxy-utils
```

Install personal certificate on the machine:

```
cd $HOME
mkdir -p .globus
cd .globus
openssl pkcs12 -clcerts -nokeys -in cert.p12 -out usercert.pem
openssl pkcs12 -nocerts -in cert.p12 -out userkey.pem
chmod 600 usercert.pem
chmod 400 userkey.pem
```

To check all is correctly working:

```
grid-proxy-init -valid 168:00
gfal-copy --version
```

The last command should produce a list of the available protocols. The list should include *gridftp*. If this is not the case, try to do: `yum update`.

Some examples of gfal utils below.

```

-bash-4.2$ gfal-ls srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4
generated
virgo
RomePSS
virgoIn
virgoout_O2LH_01_0127_067_-81.305_-90.000_.mat
prova_lucia2
prova_lucia3
vedovato
virgojob_summary_O2LH_01_0127_067_-81.305_-90.000_.mat
virgoRomePSS
testalbe
vdbroeck
prova_lucia
prova_andrea

-bash-4.2$ gfal-rm srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea
srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea DELETED

-bash-4.2$ gfal-rm -R srm://storm-fe-archive.cr.cnaf.infn.it:8444/ams/folder_test
srm://storm-fe-archive.cr.cnaf.infn.it:8444/ams/folder_test/dlattanzio2ams.txt DELETED
srm://storm-fe-archive.cr.cnaf.infn.it:8444/ams/folder_test RMDIR

-bash-4.2$ gfal-copy /home/USER-SUPPORT/arendina/sleep.sub srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4
/prova_andrea
Copying file:///home/USER-SUPPORT/arendina/sleep.sub [DONE] after 2s

-bash-4.2$ gfal-sum srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea ADLER32
srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea 2bca5372

```

For some experiments it is also possible to use the https or davs protocol contacting directly the StoRM frontend in the same way, but some configuration changes of the gfal tool are needed on the client. In fact, the SRM plugin specifics into the `/etc/gfal2.d/srm_plugin.conf` file can be ordered as in the following example:

```

# ordered list of turls protocols for remote I/O
# the top priority protocol is the first one
TURL_PROTOCOLS=davs;gsiftp;rfio;gsidcap;dcap;kdcap

# ordered list of turls protocols for third party transfer
# only protocol supporting third party copy should be here
# the top priority protocol is the first one
TURL_3RD_PARTY_PROTOCOLS=https;gsiftp;root

```

For all the gfal commands see [\[16\]](#).

## ClientSRM utils

In case of local to remote transfer, you have to request the storage space in the destination filesystem and this is done with the command `clientSRM PtP`, where PtP stands for *Prepare to Put*. For example:

```

$ clientSRM PtP -e http://storm-fe-archive.cr.cnaf.infn.it:8444 -s srm://storm-fe-archive.cr.cnaf.infn.it:8444
/virgo4/prova_andrea

```

where:

- `-e` is used to specify the endpoint.
- `-s` is used to specify the destination url, which is composed by a space token (virgo4 in the example) and the file path. The space token will be communicated by the Experiment Support group.
- `-v` is the verbose level, further.

and the complete list of the options is listed by the command `clientSRM PtP -help` or in [\[17\]](#). The output should be something like this:

```

-bash-4.2$ clientSRM PtP -e httpg://storm-fe-archive.cr.cnaf.infn.it:8444 -s srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea
=====
Sending PtP request to: httpg://storm-fe-archive.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 17
Poll Flag 0
=====
Request status:
statusCode="SRM_REQUEST_QUEUED" (17)
=====
SRM Response:
requestToken="51e58c63-afdd-4ccb-8a6d-8551b4261c33"
arrayOfFileStatuses (size=1)
[0] SURL="srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea"
[0] status: statusCode="SRM_REQUEST_QUEUED" (17)
explanation=""
=====

```

It is important to pay attention to the request token (in this case 51e58c63-afdd-4ccb-8a6d-8551b4261c33), which will be used later. Then it is necessary to check the status of the request with **clientSRM SPTP** (*Status of Prepare to Put*) :

```

clientSRM SPTP -v -e httpg://storm-fe-archive.cr.cnaf.infn.it:8444 -t 51e58c63-afdd-4ccb-8a6d-8551b4261c33

```

where with **-t** you provide the token shown in the output of the **clientSRM PtP** command. The output will show you whether the request is successful in the status field.

```

-bash-4.2$ clientSRM SPTP -v -e httpg://storm-fe-archive.cr.cnaf.infn.it:8444 -t 51e58c63-afdd-4ccb-8a6d-8551b4261c33
=====
Sending StatusPtP request to: httpg://storm-fe-archive.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 0
Poll Flag 0
=====
Request status:
statusCode="SRM_SUCCESS" (0)
explanation="All chunks successfully handled!"
=====
SRM Response:
arrayOfFileStatuses (size=1)
[0] SURL="srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea"
[0] status: statusCode="SRM_SPACE_AVAILABLE" (24)
explanation="srmPrepareToPut successfully handled!"
[0] TURL="gsiftp://ds-815.cr.cnaf.infn.it:2811//storage/gpfs_virgo4/virgo4/prova_andrea"
=====

```

It is important to remember the TURL which will be used in transfer command with the **globus-url-copy** command. After that, we can perform the file transfer:

```

bash-4.2$ globus-url-copy /home/USER-SUPPORT/arendina/sleep.sub gsiftp://ds-815.cr.cnaf.infn.it:2811//storage/gpfs_virgo4/virgo4/prova_andrea

```

Actually, with this command we overwrite the file "prova\_andrea", prepared with the **clientSRM PtP** command, with the local file "sleep.sub".

At the end, in order to avoid the open request in the StoRM database, we have to finish with the **clientSRM Pd** command where Pd stands for *Put done*:

```
-bash-4.2$ clientSRM Pd -e httpg://storm-fe-archive.cr.cnaf.infn.it:8444 -t 51e58c63-afdd-4ccb-8a6d-8551b4261c33 -s srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea
=====
Sending PutDone request to: httpg://storm-fe-archive.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 0
Poll Flag 0
=====
Request status:
statusCode="SRM_SUCCESS"(0)
explanation="All file requests are successfully completed"
=====
SRM Response:
arrayOfFileStatuses (size=1)
[0] SURL="srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea"
[0] status: statusCode="SRM_SUCCESS"(0)
explanation="Success"
=====
```

Similarly, in case of remote to local transfer the command is **clientSRM PtG**, where PtG stands for *Prepare to Get*, and to check the preparation status is **clientSRM SPtG**:

```
-bash-4.2$ clientSRM PTG -e httpg://storm-fe-archive.cr.cnaf.infn.it:8444 -s srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea
=====
Sending PtG request to: httpg://storm-fe-archive.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 17
Poll Flag 0
=====
Request status:
statusCode="SRM_REQUEST_QUEUED"(17)
=====
SRM Response:
requestToken="fe633fd3-de07-4a3e-a388-3cc2adflfd3a"
arrayOfFileStatuses (size=1)
[0] status: statusCode="SRM_REQUEST_QUEUED"(17)
explanation=""
[0] sourceSURL="srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea"
=====

-bash-4.2$ clientSRM SPtG -e httpg://storm-fe-archive.cr.cnaf.infn.it:8444 -t fe633fd3-de07-4a3e-a388-3cc2adflfd3a
=====
Sending StatusPtG request to: httpg://storm-fe-archive.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 0
Poll Flag 0
=====
Request status:
statusCode="SRM_SUCCESS"(0)
explanation="All chunks successfully handled!"
=====
SRM Response:
arrayOfFileStatuses (size=1)
[0] status: statusCode="SRM_FILE_PINNED"(22)
explanation="srmPrepareToGet successfully handled!"
[0] sourceSURL="srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/prova_andrea"
[0] fileSize=280
[0] transferURL="gsiftp://ds-915.cr.cnaf.infn.it:2811//storage/gpfs_virgo4/virgo4/prova_andrea"
=====
```

So we can perform the transfer with a globus-url-copy command:

```
-bash-4.2$ globus-url-copy gsiftp://ds-915.cr.cnaf.infn.it:2811//storage/gpfs_virgo4/virgo4/prova_andrea copia
-bash-4.2$ ls
ce_testp308.sub copia pass sleep.sh sleep.sub test.sub
```

Finally, to list the file in a directory you can use the command **clientSRM Ls**:

```
-bash-4.2$ clientSRM Ls -e http://storm-fe-archive.cr.cnaf.infn.it:8444 -s srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4
```

A user is also able to delete a file using the SRM clients. An easy example follows below:

```
[arendina@ui-tier1 ~]$ clientSRM Rm -e http://storm-fe-archive.cr.cnaf.infn.it:8444 -s srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/test2402
=====
Sending Rm request to: http://storm-fe-archive.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 0
Poll Flag 0
=====
Request status:
  statusCode="SRM_SUCCESS"(0)
  explanation="All files removed"
=====
SRM Response:
  arrayOfFileStatuses (size=1)
    [0] SURL="srm://storm-fe-archive.cr.cnaf.infn.it:8444/virgo4/test2402"
    [0] status: statusCode="SRM_SUCCESS"(0)
           explanation="File removed"
=====
```

More information on using SRM clients can be found here [\[18\]](#).

**NB:** For some VOs there is also the possibility to retrieve an https TURL using the ClientSRM utils. Indeed, this could be done adding the **-T -P https** option, for instance:

```

[arendina@ui-tier1 ~]$ clientSRM PtP -T -P https -e httpg://storm-fe-ams.cr.cnaf.infn.it:8444 -s srm://storm-fe-ams.cr.cnaf.infn.it:8444/juno/storm/prova_andrea
=====
Sending PtP request to: httpg://storm-fe-ams.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 17
Poll Flag 0
=====
Request status:
  statusCode="SRM_REQUEST_QUEUED"(17)
=====
SRM Response:
  requestToken="de8af950-9869-4e50-891f-a8510c36cfdd"
  arrayOfFileStatuses (size=1)
    [0] SURL="srm://storm-fe-ams.cr.cnaf.infn.it:8444/juno/storm/prova_andrea"
    [0] status: statusCode="SRM_REQUEST_QUEUED"(17)
      explanation=" "
=====

[arendina@ui-tier1 ~]$ clientSRM SPtP -e httpg://storm-fe-ams.cr.cnaf.infn.it:8444 -s srm://storm-fe-ams.cr.cnaf.infn.it:8444/juno/storm/prova_andrea -t "de8af950-9869-4e50-891f-a8510c36cfdd"
=====
Sending StatusPtP request to: httpg://storm-fe-ams.cr.cnaf.infn.it:8444
Before execute:
Afer execute:
Request Status Code 0
Poll Flag 0
=====
Request status:
  statusCode="SRM_SUCCESS"(0)
  explanation="All chunks successfully handled!"
=====
SRM Response:
  arrayOfFileStatuses (size=1)
    [0] SURL="srm://storm-fe-ams.cr.cnaf.infn.it:8444/juno/storm/prova_andrea"
    [0] status: statusCode="SRM_SPACE_AVAILABLE"(24)
      explanation="srmPrepareToPut successfully handled!"
    [0] TURL="https://ds-510.cr.cnaf.infn.it:8443/juno/storm/prova_andrea"
=====

```