



Enabling Grids for E-science

# Data Management Services Practicals

*Riccardo Bruno*

*INFN*

*gLite Tutorial at the First EGEE User Forum*

*CERN, 27-28.02.2006*

[www.eu-egee.org](http://www.eu-egee.org)



Information Society





# Practicals on Data Management LFC and lcg-utils

- Check the content of the environment variable **\$LFC\_HOST**:

```
[localhost] /home/gildauser/globus > echo $LFC_HOST
LFC_HOST=lfc.gilda.ct.infn.it
```

- Ensure that you have your certificates in the **.globus** directory inside your home:

```
[localhost] /home/gildauser/.globus > ll
total 8
-rw-r--r--  1 gildauser gildauser   1644 Jan  4 13:15 usercert.pem
-r-----  1 gildauser gildauser   1925 Jan  4 15:22 userkey.pem
```

- If you don't have a proxy or it has expired, create it with the following command:

```
$ voms-proxy-init --voms gilda
```

- Listing the entries of a LFC directory

*lfc-ls [-cdiLLRTu] [--comment] path*

where *path* specifies the LFC pathname (mandatory)

- Remember that LFC has a directory tree structure
- */grid/<VO\_name>/<you create it>*



- All members of a given VO have read-write permissions under their directory
- *-l* (it is a lowercase “L”) outputs long listing
- *-R* lists the contents of directories recursively (don’t use it so often!)
- You can set *LFC\_HOME* to use relative paths:  
*LFC\_HOME=/grid/gilda/myDir* → */grid/gilda/myDir/myFile*  
 becomes *myFile*

```
$ lfc-ls -l /grid/gilda
```

```
...
-rw-rw-r-- 1 4401 4400 0 Jun 21 2005 user.example
lrwxrwxrwx 1 4401 4400 0 Aug 26 03:32 user.example-link -> user.example
lrwxrwxrwx 1 4467 4400 0 Feb 10 03:25 user.example.link -> user.example
lrwxrwxrwx 1 4407 4400 0 Aug 26 03:31 user.example.tokyo33 /grid/gilda/user.example
-rw-rw-r-- 1 4401 4400 0 Jun 21 2005 user.example2
drwxr-xr-x 2 4466 4400 0 Feb 10 06:07 user04
drwxrwxrwx 7 4473 4400 0 Feb 10 05:59 user23
-rw-rw-r-- 1 4401 4400 115 Jun 21 2005 valencia15.ejemplo
...
```

```
$ export LFC_HOME=/grid/gilda/
```

```
$ lfc-ls -l user.example
```

```
-rw-rw-r-- 1 4401 4400 0 Jun 21 2005 /grid/gilda/user.example
```

- **Creating a symbolic link**

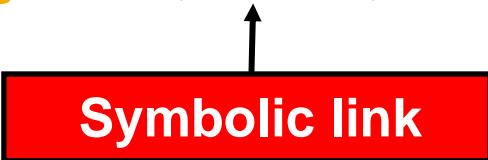
```
lfc-ln -s file linkname
```

```
lfc-ln -s directory linkname
```

Creates a link to the specified *file* or *directory* with *linkname*

```
$ lfc-ln -s /grid/gilda/user.example \  
  /grid/gilda/cern/linkToUser.example
```

**Symbolic link**



**Original file**



Let's check the link using *lfc-ls* with long listing (-l):

```
$ lfc-ls -l /grid/gilda/cern
```

```
lrwxrwxrwx  1 4432  4400          0 Feb 16 16:48 linkToUser.ex ->  
/grid/gilda/user.example
```

- **Creating directories in the LFC**

```
lfc-mkdir [-m mode] [-p] path
```

where *path* specifies the LFC pathname

- Remember that while registering a new file (using **lcg-cr**, for example) the corresponding destination directory must already be created in the catalog.

- Examples:

```
$ lfc-mkdir /grid/gilda/cern/<mydir>
```

- You can just check the directory with:

```
$ lfc-ls -l /grid/gilda/cern
```

- Adding/deleting metadata information

```
lfc-setcomment <path> <comment>
```

```
lfc-delcomment <path>
```

*lfc-setcomment* adds/replaces a *comment* associated with a file/directory in the LFC Catalog.

If the file/directory already has a comment associated it will be replaced by the new one.

*lfc-delcomment* deletes a comment previously added.

If there is no comment associated with the file, this command returns the error “No such file or directory”



- **Example:**

```
lfc-setcomment /grid/gilda/user.example "Hello World"
```

- **Check your job with:**

```
lfc-ls --comment /grid/gilda/user.example
```

```
-rw-rw-r-- 1 4401 4400 0 Jun 21 2005 /grid/gilda/user.example Hello World
```

- **Example:**

```
lfc-delcomment /grid/gilda/user.example
```

- **Check your job with:**

```
lfc-ls -l --comment /grid/gilda/user.example
```

```
-rw-rw-r-- 1 4401 4400 0 Jun 21 2005 /grid/gilda/user.example
```

## Exercise No.1:

- Log onto an UI and initialize your proxy credentials if not already done
- set up properly the environment variables to use `lfc-gilda.ct.infn.it` catalog
- have a look inside the catalog
- create a directory with your surname, under the directory `/grid/gilda/cern/<yourdir>`
- put a link to an existing file into the directory that you created
- add a comment to that file and verify it



## Summary of the LFC Catalog commands

|                       |  |
|-----------------------|--|
| <b>lfc-chmod</b>      | <b>Change access mode of the LFC file/directory</b>          |
| <b>lfc-chown</b>      | <b>Change owner and group of the LFC file/directory</b>      |
| <b>lfc-delcomment</b> | <b>Delete the comment associated with the file/directory</b> |
| <b>lfc-getacl</b>     | <b>Get file/directory access control list</b>                |
| <b>lfc-ln</b>         | <b>Make a symbolic link to a file/directory</b>              |
| <b>lfc-ls</b>         | <b>List file/directory entries in a directory</b>            |
| <b>lfc-mkdir</b>      | <b>Create a directory</b>                                    |
| <b>lfc-rename</b>     | <b>Rename a file/directory</b>                               |
| <b>lfc-rm</b>         | <b>Remove a file/directory</b>                               |
| <b>lfc-setacl</b>     | <b>Set file/directory access control list</b>                |
| <b>lfc-setcomment</b> | <b>Add/replace a comment</b>                                 |

- The LCG Data Management tools (usually called **lcg-utils**) allow users to:
  - copy files between **UI**, **CE**, **WN** and **SE**;
  - register entries in the File Catalog;
  - replicate files between **SEs**.
- Check if **LCG\_GFAL\_INFOSYS** environment variable is correctly set to the local GILDA Information Index (BDII):

```
export LCG_GFAL_INFOSYS=grid004.ct.infn.it:2170
```

- Upload a file to a SE and register it into the catalog

```
lcg-cr -d dest_file | dest_host [-g guid] [-l lfn]
[-v | --verbose] --vo vo src_file
```

where:

- **dest\_host** is the fully qualified hostname of the destination SE
- **dest\_file** is a valid SURL (both sfn:// or srm:// format are valid)
- **guid** specifies the Grid Unique Identifier. If this option is not present, a GUID is generated internally
- **lfn** specifies the Logical File Name associated with the file
- **vo** specifies the Virtual Organization the user belongs to
- **src\_file** specifies the source file name: the protocol can be *file:///* or *gsiftp:///*

- **Example:**

```

$ lcg-cr -v -d grid009.ct.infn.it -l lfn:/grid/gilda/cern/test.txt \
  --vo gilda file:/home/cern/prova
Using grid catalog type: lfc
Source URL: file:/home/cern/prova
File size: 353
Destination specified: grid009.ct.infn.it
Destination URL for copy:
  gsiftp://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006-0...
# streams: 1
Alias registered in Catalog: lfn:/grid/gilda/cern/test.txt
Transfer took 1200 ms
Destination URL registered in Catalog:
  sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006-0...
guid:bf95f82e-de21-4452-a4b5-b9d40a94ee2c

```

- **To discover which SEs the user is allowed to use, remember you can use `lcg-infosites` command:**

`lcg-infosites --vo gilda se`

The output is a list of SEs and related information on available/used space.

- **Adding an alias for a given GUID**

```
lcg-aa --vo vo guid lfn
```

where:

- **vo** specifies the Virtual Organization the user belongs to
- **guid** specifies the Grid Unique Identifier of the file you want to add the alias
- **lfn** specifies the new alias

- Example:

```
$ lcg-aa --vo gilda guid:bf95f82e-de21-4452-a4b5-b9d40a94ee2c lfn:/grid/gilda/cern/alias.txt
```

- To check if the previous command was successful, you can use **lcg-la** command to list the aliases for a given LFN, GUID or SURL:

```
$ lcg-la --vo gilda lfn:/grid/gilda/cern/alias.txt
lfn:/grid/gilda/cern/test.txt
lfn:/grid/gilda/cern/alias.txt
```

## Exercise No.2:

- verify that your **LCG\_GFAL\_INFOSYS** environment variable is correctly set up
- create a dummy file in your home directory
- list the available storage elements
- copy and register the dummy file into the directory you created
- add an alias to the uploaded file
- check if the alias was created correctly





- **Copy a file between SEs and register it in the Catalog**

```
lcg-rep -d dest_file | dest_host [-v | --verbose]
--vo vo src_file
```

where:

- **dest\_host** is the fully qualified hostname of the destination SE
- **dest\_file** is a valid SURL (both sfn:// or srm:// are valid)
- **vo** specifies the Virtual Organization the user belongs to
- **src\_file** specifies the source file name: the protocol can be LFN, GUID or SURL. An SURL scheme can be sfn: for a classical SE or srm:

```
$ lcg-rep -v -d grid005.iucc.ac.il --vo gilda \
lfn:/grid/gilda/cern/test.txt
Using grid catalog type: lfc
Source URL: lfn:/grid/gilda/cern/test.txt
File size: 186
Destination specified: grid005.iucc.ac.il
Source URL for copy: gsiftp://gilda-se-01.pd.infn.it/shared/...
Destination URL for copy: gsiftp://grid005.iucc.ac.il/storage/.../filebe...
# streams: 1
Transfer took 10980 ms
Destination URL registered in LRC:
sfn://grid005.iucc.ac.il/storage/gilda/generated/2006-02-20/filebe...
```

- **Listing replicas for a given LFN, GUID or SURL**

```
lcg-lr --vo vo file
```

where:

- **vo** specifies the Virtual Organization the user belongs to
- **file** specifies the Logical File Name, the Grid Unique Identifier or the Site URL. A SURL scheme can be sfn: or srm:

- **Example:**

```
$ lcg-lr --vo gilda lfn:/grid/gilda/cern/note.txt
sfn://grid-se.bio.dist.unige.it/flatfiles/SE00/gilda/generated...
sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2005-0...
```

we get the same output using the GUID of the file:

```
$ lcg-lr --vo gilda guid:4c10a8e3-2244-4c38-...
```

- **Deleting replicas**

```
lcg-del [ -a ] [ -s se ] [ -v | --verbose ] --vo vo file
```

where:

- **a** is used to delete all replicas of the given file
- **se** specifies the SE from which you want to remove the replica
- **vo** specifies the Virtual Organization the user belongs to
- **file** specifies the Logical File Name, the Grid Unique Identifier or the Site URL. A SURL scheme can be sfn: for a classical SE or srm:.

- **Example:**

- delete one replica

```
$ lcg-del --vo gilda -s grid009.ct.infn.it
lfn:/grid/gilda/cern/alias.txt
```

- delete all the replicas

```
$ lcg-del -a --vo gilda lfn:/grid/gilda/cern/test.txt
```

- let's check if the previous command was successful

```
$ lcg-lr --vo gilda lfn:/grid/gilda/cern/test.txt
lcg_lr: No such file or directory
```

- or by `lfc-ls /grid/gilda/cern` (you will not see anymore test.txt and its alias)

- **Downloading a Grid file in a SE to a local destination**

```
lcg-cp [ -v | --verbose ] --vo vo src_file dest_file
```

where:

- **vo** specifies the Virtual Organization the user belongs to
- **src\_file** specifies the source file name: the protocol can be LFN, GUID, SURL or local file. A SURL scheme can be sfn: for a classical SE or srm:
- **dest\_file** specifies the destination. The protocol can be file:/// or gsiftp://

- **Example:**

```
$ lcg-cp -v --vo gilda lfn:/grid/gilda/cern/test.txt
file:/home/cern/test.txt
```

```
Source URL: lfn:/grid/gilda/cern/test.txt
```

```
File size: 353
```

```
Source URL for copy:
```

```
gsiftp://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006...
```

```
Destination URL: file:/home/cern/test.txt
```

```
# streams: 1
```

```
Transfer took 1360 ms
```

## Exercise No.3:

- Create two replicas of the file you previously uploaded (you could also use the alias to point it out)
- Check if the operation was successful
- Download the file back to your UI
- Delete only one replica and verify that
- Delete all the replicas and verify that
- Verify if the entry is still present in the catalog



- **GOAL:**

Submit a job that does data management: it will retrieve a file previously registered into the catalog.

- **Steps to follow up:**

- Create a new file in your UI and put some data into it
- Choose a SE to upload the file to (hint: use **lcg-infosites**) and use the appropriate command to accomplish at this operation

```
(lcg-cr -v --vo gilda -l
lfn:/grid/gilda/cern/<choose an lfn> -d <an SE
host> file:`pwd`/<your new file>)
```

- create a script.sh file with the following content:

```
#!/bin/sh
/bin/hostname
#Change the LFN_NAME to download from the Catalog.
echo "Start to download.."
lcg-cp --vo gilda lfn:/grid/gilda/cern/<lfn you choose> file:`pwd`/output.dat
echo "Done.."
```

- **Create the JobWithData.jdl:**

```
Type = "job";
JobType = "Normal";

Executable = "/bin/sh";
Arguments = "script.sh";

VirtualOrganisation = "gilda";

StdOutput = "std.out";
StdError = "std.err";

InputSandbox = {"script.sh"};
OutputSandbox = {"std.out", "std.err", "output.dat"};
```

- **Submit it to the grid**
- **Retrieve the output and verify the content of output.dat**

## Replica Management

|                |  |
|----------------|--|
| <b>lcg-cp</b>  | Copies a grid file to a local destination                              |
| <b>lcg-cr</b>  | Copies a file to a SE and registers the file in the catalog            |
| <b>lcg-del</b> | Deletes one file   |
| <b>lcg-rep</b> | Replication between SEs and registration of the replica to the catalog |
| <b>lcg-gt</b>  | Gets the TURL for a given SURL and transfer protocol                   |
| <b>lcg-sd</b>  | Sets file status to "Done" for a given SURL in a SRM request           |

## File Catalog Interaction

|               |  |
|---------------|--|
| <b>lcg-aa</b> | Adds an alias in LFC for a given GUID            |
| <b>lcg-ra</b> | Removes an alias in LFC for a given GUID         |
| <b>lcg-rf</b> | Registers in LFC a file placed in a SE           |
| <b>lcg-uf</b> | Unregisters in LFC a file placed in a SE         |
| <b>lcg-la</b> | Lists the alias for a given SURL, GUID or LFN    |
| <b>lcg-lg</b> | Gets the GUID for a given LFN or SURL            |
| <b>lcg-lr</b> | Lists the replicas for a given GUID, SURL or LFN |





# Practicals on Catalog gLite - FiReMan

- **Browsing the contents of a directory**

```
glite-catalog-ls [-h] [-q] [-s SERVICE] [-V] [-v] [-c]
[-d] [-g] [-l] [-R] [-S] [-t] [-u] [-x] LFN
```

where: **LFN** (Logical File Name) is the absolute path of the file/directory to list.

- **Main Options:**

- l request long output
- v increase the verbosity level
- R request recursive listing
- S sort by size
- t sort by modification time
- g print GUIDs in the long listing
- h print a short help
- s **SERVICE** specifies the service endpoint to use
- d list the directory node itself instead of its contents

- **Example:**

```
$glite-catalog-ls -l /satimages
```

```
-pdrwl-gspdrwl-gspdrwl-gs 9950k 2006-01-21 14:31:14 /satimages/20060719.jpg
-pdrwl-gspdrwl-gspdrwl-gs 3462k 2006-01-23 11:37:48 /satimages/20060720.jpg
-pdrwl-gspdrwl-gspdrwl-gs 5154k 2006-01-23 11:41:25 /satimages/20060721.jpg
-pdrwl-gspdrwl-gspdrwl-gs 5692k 2006-01-23 11:45:01 /satimages/20060722.jpg
-pdrwl-gspdrwl-gspdrwl-gs 5774k 2006-01-23 10:28:04 /satimages/20060723.jpg
-pdrwl-gspdrwl-gspdrwl-gs 5676k 2006-01-23 11:48:14 /satimages/20060724.jpg
-pdrwl-gspdrwl-gspdrwl-gs 6094k 2006-01-23 11:32:31 /satimages/20060725.jpg
```

- **Meaning of the flags:**

The first letter shows the type of the entry: '-' for regular files, 'd' for directories, 'l' for **symbolic links** and 'v' for **virtual directories**.

It follows 3 series of 8 flags, respectively for the **owner**, the **group** and **others**: p indicates the permission to **change attribute**, while d gives rights to **delete** the entry. It follows permissions to **read**, **write**, **list**, **execute**. The last two flags are reserved for **metadata** use, and so are currently unused. They will give the rights to **get** or **set** the metadata.

- **Directory creation**

```
glite-catalog-mkdir [-h] [-q] [-s URL] [-V] [-v] [-p]
[-c] DIR
```

Where **DIR** is the name of the directory to create (in LFN format)

- p create missing parent directory

- c copy the permission of the parent directory

**Example:**

```
glite-catalog-mkdir /cern
```

- **Deleting a directory**

```
glite-catalog-rmdir [-h] [-q] [-s URL] [-V] [-v] DIR
```

The specified directory must be empty or the command will fail

**Example:**

```
glite-catalog-rmdir /cern
```

- Getting file information

```
glite-catalog-stat [-h] [-q] [-s SERVICE] [-V] [-v] [-r] LFN
```

Lists all information about a file or dir in the catalog. It includes LFN, GUID, owner/group with basic permission, list of ACLs (if any), SURL for every replica.

```
$ glite-catalog-stat /cern/new
```

```
LFN:                /cern/new
  Created:           2006-02-17 10:20:34.000
  Modified:          2006-02-17 10:20:34.000
  Size:              0
  Type:              File
  Expires:           Never
GUID:                3dd02562-5131-49ff-95ff-361e501f27bf
  Created:           2006-02-17 10:20:34.000
  Modified:          2006-02-17 10:20:34.000
  Size:              0
  Checksum:          <none>
  Status:            0
User:                /C=IT/O=GILDA/OU=Personal Certificate/L=INFN-Catania/CN=<User Name>...
Group:               egee-group
User rights:         pdrwxgs
Group rights:        p---l-g-
Other rights:        p---l-g-
```

- **Locates files matching a specified pattern in the specified directory**

```
glite-catalog-find [-h] [-q] [-s SERVICE] [-V] [-v]
[-n LIMIT] PATTERN DIR
```

Where:

- **-n LIMIT** return no more than *LIMIT* results. The default is 100
- **PATTERN** is the file name pattern to look for. Recognized wildcards are **?** matching any single character, and **\*** matching any string
- **DIR** Name of the directories where to look for files matching *PATTERN*.

- **Example:**

```
$ glite-catalog-find tes* /
/brunor/test
/pisa/grandi/test
/elmsheus/test
/karlsruhe46/test
```

## Summary of the Fireman Catalog commands

|   |  |
|---|--|
| <b>glite-catalog-ls</b>                                   | <b>Lists file/directory entries in a directory</b>                     |
| <b>glite-catalog-mkdir</b>                                | <b>Creates a directory</b>   |
| <b>glite-catalog-mv</b>                                   | <b>Renames a file/directory</b>  |
| <b>glite-catalog-rm</b><br><b>glite-catalog-rmdir</b>     | <b>Removes a file/directory</b>  |
| <b>glite-catalog-getreplica</b>                           | <b>Gets all replicas associated with a file/GUID</b>                   |
| <b>glite-catalog-touch</b><br><b>glite-catalog-create</b> | <b>Creates a new entry in the catalog/update the modification time</b> |
| <b>glite-catalog-find</b>                                 | <b>Finds entries based on their name pattern</b>                       |
| <b>glite-seindex-list</b>                                 | <b>Lists all SEs having a replica of the given file</b>                |

## Summary of the Fireman Catalog commands

|  |  |
|--|--|
| <p>glite-catalog-chmod<br/>glite-catalog-setacl<br/>glite-catalog-setdefacl<br/>glite-catalog-setdefperm</p> | <p>Change access mode of the Fireman file/directory. Set the ACL, the default ACL and the default permission</p> |
| <p>glite-catalog-stat<br/>glite-catalog-getguid</p>  | <p>List the details of a file – all attributes, replicas. Or just the associated GUID.</p>                       |
| <p>glite-catalog-setattr<br/>glite-catalog-getattr<br/>glite-catalog-setschema</p>                           | <p>Set/get metadata attribute and set the metadata schema of a given directory</p>                               |
| <p>glite-catalog-getacl<br/>glite-catalog-getdefacl</p>  | <p>Get file/directory access control lists and default ACL</p>   |
| <p>glite-catalog-symlink</p>   | <p>Makes a symbolic link to a file. <b>Directory symlinks are not supported by design.</b></p>                   |



## Exercise No.4:

- Create a directory in the catalog named `/cern/<your surname>`
- Create a file in this directory using:  
`glite-catalog-create <filename>`
- Recursively list all the files of the directory `/cern` using the `-R` flag
- Get all the information about the file you created:  
`glite-catalog-stat -v /cern/<your dir>/<filename>`
- Find your file in the catalog using the `glite-catalog-find` command:  
`glite-catalog-find <file> /cern`



```
glite-put <localfilename> <lfn> [-m <mode>] [-c <config>]
```

where:

- **<localfilename>** is the name of the local file you want to upload
- **<lfn>** is the logical file name you want to assign to that file.

**glite-put** is part of the glite-io client CL tools. It will contact the local glite-IO server to accomplish its works. It also interacts with FireMan interfaces to register the new entry into the catalog.

**Example:** `$ glite-put JobWithData.jdl /cern/JobWithData.jdl`

```
1140452058 INFO glite-io-client-put : Start File Transfer
[glite-put] Total 0.00 MB          |=====| 100.00 % [0.0 Mb/s]
1140452061 INFO glite-io-client-put : File Transfer Completed
```

Transfer Completed:

```
LFN                : /cern/JobWithData.jdl
GUID               : 00867928-ead2-13f9-9e7c-c1ced08dbeef
SURL               : srm://egee016.cnaf.infn.it:8443/srm/managerv1?SFN...
Data Written [bytes] : 337
Eff.Transfer Rate[Mb/s] : 0.000044
```

```
glite-get <lfn> <localfilename> [-c <config>]
```

**Copies a file from the local Storage Element to the given local file.**

- **<lfn>** is the logical file name of the file you want to download and **<localfilename>** is the name of the destination file
- **-c <config>** use an alternative config file, overriding the glite-io-client configuration

## Example:

```
$ glite-get /cern/JobWithData.jdl test
1140452921 INFO glite_get : Start File Transfer
[glite-get] Total 0.00 MB      |=====| 100.00 % [0.0 Mb/s]
1140452922 INFO glite_get : File Transfer Completed
```

Transfer Completed:

```
LFN           : /cern/JobWithData.jdl
GUID          : 00867928-ead2-13f9-9e7c-c1ced08dbeef
SURL         : srm://egee016.cnaf.infn.it:8443/srm/managerv1?SFN...
Data Written [bytes] : 337
Eff.Transfer Rate[Mb/s] : 0.000051
```

```
glite-rm <lfn> [-c <config>]
```

Removes the given file from your local Storage Element and delete the entry in the File Catalog

## Example:

```
$ glite-rm /cern/JobWithData.jdl
```

Unlink Completed:

```
File           : /cern/JobWithData.jdl  
Time [s]       : 4.162000
```

## Exercise No.5:

- Create a dummy file in your home directory
- Upload the file you created into the directory `/cern/<your surname>` using `glite-put`
- List the files in this directory using `glite-catalog-ls`
- Retrieve the file you stored in the catalog using the `glite-get` command
- Remove all files in your directory from the catalog
- Delete your directory from the FiReMan catalog



- **LFC, gfal, lcg-utils**

- <http://indico.cern.ch/contributionDisplay.py?contribId=278&sessionId=7&confId=0>

- **gLite homepage**

- <http://www.glite.org>

- **FiReMan catalog user guide**

- <https://edms.cern.ch/file/570780/1/EGEE-TECH-570780-v1.0.pdf>

- **gLite-I/O user guide**

- <https://edms.cern.ch/file/570771/1.1/EGEE-TECH-570771-v1.1.pdf>

