

Basic data management

Last review date	Reviewer
2009-09-15	Marco Bencivenni Enrico Fattibene

Table of Contents

[Basic Data Management](#)

[Data Management Terminology](#)

[The LFC](#)

[Examples](#)

Basic Data Management

How to use the data management command-line tools to store, copy and delete files.

The following examples assume that you have already created a proxy, that you are using a User Interface (UI) which has been configured in a standard way, that you are using a Bourne-type shell and that your VO name is myvo. The commands will also work inside a running job, since the environment there satisfies the same requirements.

Useful references:

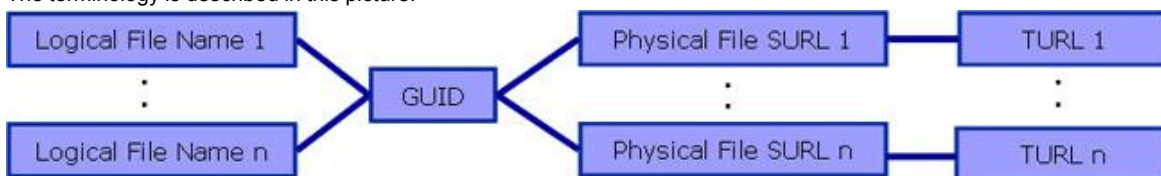
- [lcg-utils command line and API reference](#)
- [LFC command line and API reference](#)

Data Management Terminology

In the Grid, a file is stored in a *Storage Element* (SE). Files cannot be modified once written, only deleted. One logical file may have several identical replicas in different SEs. Files are identified by a *Logical File Name* (LFN), and a *file catalogue* stores the connection between the LFN and pointers to any replicas. Such a pointer is known as a *Site URL* (SURL). The SURL may be partly specified by the user, but it can be generated automatically so for simple cases there is no need to worry about it.

Files are also identified by a *Globally Unique Identifier* (GUID), which is a fixed-format string generated by the middleware and guaranteed to be absolutely unique. However, this is not very human-friendly, and for most purposes you can ignore it and just use the LFN.

The terminology is described in this picture:



The LFC

The file catalogue technology currently used in the EGEE Grid is called the *LCG File Catalogue* (LFC).

For some purposes you need to know the host name of the LFC for your VO. This information can be obtained with the `lcg-infosites` command, e.g.:

```
lcg-infosites --vo myvo lfc
```

```
lfc-myvo.example.org
```

If it is not defined by default, it should be stored in the environment variable `LFC_HOST`, e.g.:

```
export LFC_HOST=`lcg-infosites --vo myvo lfc`
```

LFNs follow a Unix- style naming system. The namespace can be explored with the `lfc-ls` command, which works in a similar way to the standard `ls`, although you should bear in mind that the underlying technology is quite different from a Unix file system. In particular, **recursive listings from high up in the namespace hierarchy should be avoided**, as this can place a large load on the server.

The top of the LFN namespace is normally `/grid/myvo`. The organisation of the namespace is defined by each VO, so you may need to consult VO- specific documentation to see if users are expected to create files in a particular area. You may also be able to see how the hierarchy is structured by browsing the directory tree with `lfc-ls`, e.g. user files might typically be created under a directory called `/grid/myvo/users` in subdirectories named for the user in question. It is also possible that your UI configuration may have predefined the `LFC_HOME` environment variable with the path of your home directory; when using the commands described below the content of this variable is prefixed to LFNs which do not start with a `/`.

Alternatively, for testing purposes, a temporary directory with a distinctive name can be created and deleted afterwards. This can be done using the `lfc-mkdir` and `lfc-rm` commands respectively, but **don't use `lfc-rm` to delete files**, use `lcg-del` as described below, as this deletes the file itself as well as the catalogue entry.

Examples

The following examples assume that `LFC_HOME` points to a suitable directory in which to create test files. Directories are not created automatically, so this needs to be done first if necessary, e.g.:

```
lfc-mkdir -p /grid/myvo/user/`id -nu`/test
export LFC_HOME=/grid/myvo/user/`id -nu`/test
```

which creates a directory path named for the current Unix user ID and sets `LFC_HOME` to point to it.

The following examples illustrate simple cases for storing, replicating, retrieving and deleting Grid files. A `-v` option can be given to the `lcg-*` commands to get a more verbose description of what the command is doing.

The commands all need to know the name of the your VO. Recent versions of the tools can take this from a VOMS proxy, or it can be specified explicitly with a `--vo` option. It can also be set as a default via the `LCG_GFAL_VO` environment variable, e.g.:

```
export LCG_GFAL_VO=myvo
```

The examples assume that a default VO is available in one way or another.

Write a file to the Grid

To begin with, create a test file called `hw`, containing the string "Hello World":

```
echo "Hello World" | cat > hw
```

Store the file on the Grid with the `lcg-cr` command (`cr` = copy and register), using an LFN of `test1` relative to the home directory:

```
lcg-cr file:`pwd`/hw -l lfn:test1
guid:edfce915-69e8-4b51-ad80-aaefbf2de7fb
```

where the response from the command shows the allocated GUID. Note that local files must be referred to as `file:` URLs using an absolute path; the format is also picky about the number of leading `/` characters.

Technically, the LFN is optional and the file can be referred to using the GUID, but normally you should use an LFN. You can check that the LFN has been created using the `lfc-ls` command as before:

```
lfc-ls -l
```

```
-rw-rw-r-- 1 19277 2688 12 Sep 07 16:15 test1
```

This form of the `lcg-cr` command will use a standard default SE, usually at your own site, to store the file. To get a list of all SEs available to your VO you can again use the `lcg-infosites` command:

```
lcg-infosites --vo myvo -v 1 se
```

```
se1.example.org  
se2.example.org  
se3.example.org
```

You can store a file on a specific SE by adding a `-d` option to `lcg-cr` followed by the name of the SE.

Replicating a Grid file

Alternatively, an existing file can be replicated to another SE:

```
lcg-rep lfn:test1 -d se2.example.org
```

You can see the SURLs of all replicas registered under a given LFN with the `lcg-lr` command (`lr` = list replicas):

```
lcg-lr lfn:test1
```

```
srm://se1.example.org/pnfs/example.org/data/myvo/generated/2007-09-12/  
fileceab3763-b674-4311-be75-c01c69d41034  
srm://se2.example.org/dpm/example.org/home/myvo/generated/2007-09-12/  
file1e93b10b-31b9-4665-aa4a-55fb0e93b
```

The response shows one line per replica, with a URL containing the hostname of the SE and an internal file name. In this example the part of the name after `/myvo/` has been generated automatically; it is possible for the user to specify this part of the name but this is not needed for simple applications. These SURLs can be used in place of the LFN or GUID if you want to refer to a specific replica, otherwise the tools will choose a replica for you.

Reading a file from the Grid

To retrieve a local copy of a file, use the `lcg-cp` command:

```
lcg-cp lfn:test1 file:`pwd`/hw2
```

Check the content of `hw2`:

```
cat hw2
```

```
Hello World
```

Deleting a file from the Grid

To delete files there are two variants of the `lcg-del` command, depending on whether you want to delete just a single replica, or every instance plus the LFN. To delete an individual SURL (as obtained from `lcg-lr`):

```
lcg-del srm://se1.example.org/pnfs/example.org/data/dteam/generated/2007-09-12/  
fileceab3763-b674-4311-be75-c01c69d41034
```

You can verify the deletion by listing the replicas again:

```
lcg-lr lfn:test1
```

```
srm://se2.example.org/dpm/example.org/home/dteam/generated/2007-09-12/\  
file1e93b10b-31b9-4665-aa4a-55fb0e93bbb1
```

Alternatively, with the `-a` option you can delete all replicas and the LFN itself, i.e. the file is completely removed from the Grid:

```
lcg-del -a lfn:test1
```

This can again be verified with `lcg-lr` and `lfc-ls`:

```
lcg-lr lfn:test1
```

```
lfc-myvo.example.org: /grid/myvo/user/johndoe/test/test1: No such file or directory  
lcg_lr: No such file or directory
```

```
lfc-ls test1
```

```
test1: No such file or directory
```

Finally, if necessary clean up by deleting the LFC directory in which the test files were created (only if the directory is empty):

```
lfc-rm -r /grid/myvo/user/`id -nu`/test
```