



# GFAL and lcg\_util

Enabling Grids for E-science

**lcg\_util** is a set of tools to copy/replicate/delete files among storage elements and register them in a Grid File Catalog. Written in C, lcg\_util provides a Python binding and a Command Line Interface (CLI)

**GFAL**, or Grid File Access Library, is a C library developed by LCG to give a uniform POSIX interface to local and remote Storage Elements (SE) on the Grid. It also provides Python interface.

## simple tools to hide the complexity

**NEW**

**Want to know where are your files are?**  
**lcg-ls srm://bigse.example.org /home/myvo/data**

- contact BDII to get the SE type (srmv1, srmv2, classic se) and endpoint
- contact the SE to get the directory content

**Want to upload and register a file?**

**lcg-cr file:/tmp/input -d bigse.example.org**

- contact BDII to get the SE type (srmv1, srmv2, classic se)
- contact the SE to get the corresponding TURL
- copy the file from local to SURL with gsiftp
- register the SURL into LFC

### Generalized Storage Operations

High level abstractions over SRMV1 and SRMV2 functions.

```
gfal_pin (gfal_internal req, char *errbuf,...)
gfal_prestage (gfal_internal req, char *errbuf,...)
gfal_prestagestatus (gfal_internal req, char *errbuf,...)
gfal_setfilchg (int fd, const void *buf, size_t size)
gfal_release (gfal_internal req, char *errbuf,...)
gfal_set_xfer_done (gfal_internal req, char *errbuf,...)
gfal_set_xfer_running (gfal_internal req, char *errbuf,...)
```

```
gfal_ls (gfal_internal req, char *errbuf,...)
gfal_turlfromsurls (gfal_internal req, char *errbuf,...)
gfal_deletesurls (gfal_internal req, char *errbuf,...)
gfal_get (gfal_internal req, char *errbuf,...)
gfal_get_results (gfal_internal req, gfal_filestatus **)
gfal_getstatus (gfal_internal req, char *errbuf,...)
```

## lcg\_util

### New commands:

- lcg-ls** lists the directory contents (i.e. srm-ls)
- lcg-getturl** get the TURL for a given SURL and transfer protocol
- lcg-aa** add an alias in RMC for a given GUID
- lcg-cp** copy a Grid file to a local destination, or copy a local file to a SE
- lcg-cr** copy and register a file
- lcg-del** delete one file (either one replica or all replicas)
- lcg-gt** get the TURL for a given SURL and transfer protocol
- lcg-la** lists the aliases for a given LFN, GUID or SURL
- lcg-lg** get the GUID for a given LFN or SURL
- lcg-lr** lists the replicas for a given LFN, GUID or SURL
- lcg-ra** remove an alias in RMC for a given GUID
- lcg-rep** copy a file from one SE to another SE and registers it in the LRC
- lcg-rf** register in the LRC (and optionally in the RMC) a file residing on an SE
- lcg-sd** set file status to "Done" for a given SURL in a specified request
- lcg-uf** unregister in the LRC a file residing on an SE

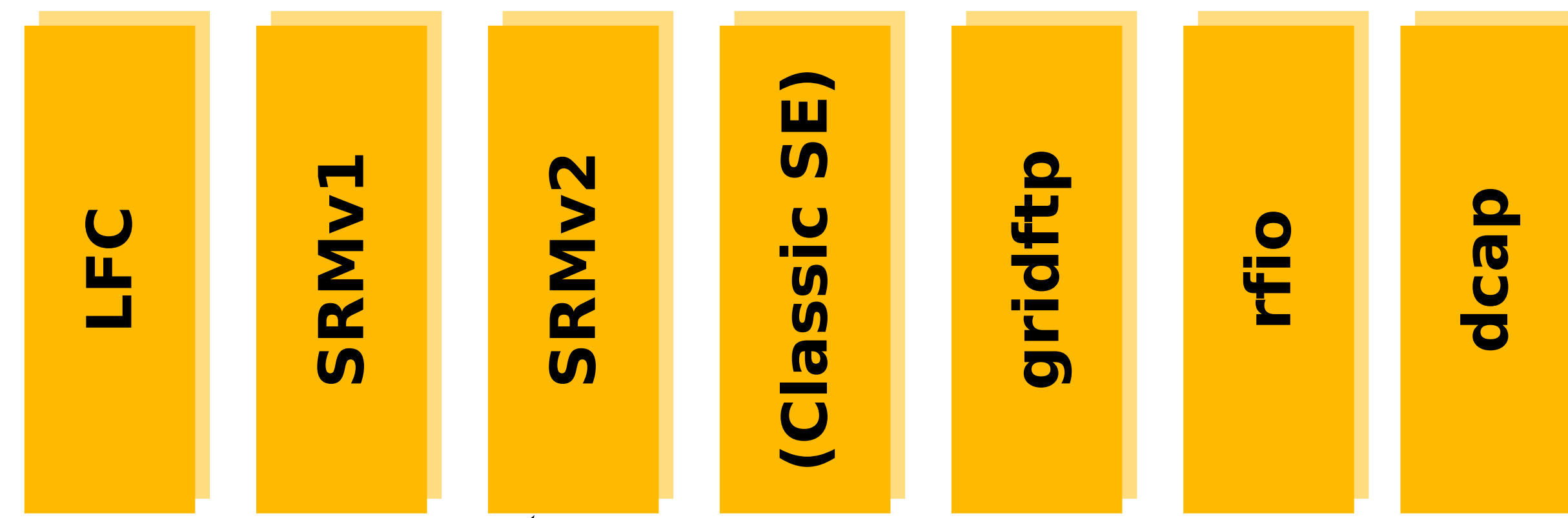
### Also available in C:

```
lcg_cp(char *src_file, char *dest_file, char *vo,
int nbstreams, char *conf_file, int insecure,
int verbose);
lcg_del(char *file, int aflag, char *se, char *vo,
char *conf_file, int insecure, int verbose);
...
and Python:
import lcg_util
lcg_util.lcg_cr(src,dest,guid,lfm,vo,...)
lcg_util.lcg_rep(src_file,dest_file,vo,relativepath,...)
```

## Data Management



## Information System / Environmental Variables



## GFAL

### POSIX style file and I/O operations

These functions try to follow the behavior and semantics of their POSIX equivalents, even in the error codes.

```
gfal_open(char *filename, int flags, mode_t mode)
gfal_creat(char *filename, mode_t mode)
gfal_read(int fd, void *buf, size_t size)
gfal_write(int fd, const void *buf, size_t size)
gfal_lseek(int fd, off_t offset, int whence)
gfal_close(int fd)
```

```
gfal_access(char *path, int amode)
gfal_lstat(char *filename, struct stat *)
gfal_stat(char *filename, struct stat *)
gfal_chmod(char *path, mode_t mode)
gfal_rename(char *old_name, char *new_name)
gfal_unlink(char *filename)
gfal_mkdir(char *dirname, mode_t mode)
gfal_rmdir(char *dirname)
gfal_opendir(char *dirname)
gfal_readdir(DIR *dir)
gfal_closedir(DIR *dir)
```

**Want to download any replicate of a file?**  
**lcg-cp lfn://home/myvo/input file://tmp/input**

- contact BDII to get the SE type (srmv1, srmv2, classic se)
- contact the SE to get the corresponding TURL
- copy the file from the SE to local with gsiftp

