

# About ELFI

Detailed description of ELFI and its features.

Contents

- ELFI for the application programmer
  - ◆ Easier porting and debugging
  - ◆ Independent of language
- ELFI for the command line user
  - ◆ ELFI usage vs lcg-utils usage
- Run-time requirements
  - ◆ GSI-enabled RFIO
  - ◆ FUSE

## ELFI for the application programmer

ELFI provides *transparent* POSIX access to Grid stored files. You can use your standard file manipulation interface — no need to learn new APIs.

Also, since ELFI is a standard Linux filesystem, you can use it from any language – not just those that already have a binding for the EGEE data management libraries.

### Easier porting and debugging

ELFI can ease the process of porting and debugging of a application: you can perform debug runs on your local machine, then submit a job to the Grid using ELFI with no need to recompile or change the code:

```
# local run
myapp -i local/input -o local/output

# Grid job
myapp -i /elfi/input -o /elfi/output
```

### Independent of language

Be it bash, C, FORTRAN or your language of choice, ELFI makes accessing Grid-stored files as normal as accessing local files on your Linux box.

FORTRAN 77:

```
c  Open a Grid file and read from it
    program elfi_file_access
    integer n
c  Open the data file
    open (10,file='elfi/LFC/grid/egrid/myfile.dat',
status='old',form='formatted')

c  Read an integer from the Grid file
    read(10,*) n
c  Write it to stdout
    write(*,*) n
c  Close the file
    close (10)
end
```

C:

```
main() {
    int n;

    /* open Grid-stored data file */
    FILE *input = fopen("elfi/LFC/grid/egrid/myfile.dat", "r");

    /* read data */
    fscanf(input, "%d", &n);

    /* write it to stdout */
    printf("read %d\n", n);

    exit(0);
}
```

bash:

```
#!/bin/bash

# read a number from a Grid-stored data file
read num < elfi/LFC/grid/egrid/myfile.dat

# write it to stdout
echo $num
```

## **ELFI for the command line user**

With ELFI you can perform all common Grid data management operations, as if all files in the Grid were on your local computer. No need to learn new and specific data management commands.

- ◆ transparently access a Grid file contents, either by logical file name:

```
emacs /elfi/LFC/logical/file/name
```

or by replica name:

```
emacs /elfi/<SE-name>/logical/file/name
```

- ◆ create and register a file in the catalog:

```
cp /path/to/local/file /elfi/LFC/logical/file/name
```

- ◆ delete a file from the catalog, and all of its replicas:

```
rm /elfi/LFC/logical/file/name
```

- ◆ delete a single replica from the catalog:

```
rm /elfi/<SE-name>/logical/file/name
```

- ◆ replicate a file across SEs:

```
ln /elfi/LFC/logical/file /elfi/<SE-name>/logical/file
```

- ◆ list LFC catalog contents:

```
ls -l /elfi/LFC/path
```

- ◆ list replicas residing on a given SE:

```
ls -l /elfi/<SE-name>/path
```

- ◆ list LFC ACL of a LFN:

```
getfacl /elfi/LFC/logical/file
```

- ◆ set LFC ACL on a LFN:

```
setfacl -m u:123:rw /elfi/LFC/logical/file
```

## **ELFI usage vs lcg-utils usage**

A detailed comparison of ELFI and lcg-utils usage can be found in the [elfi vs lcg-utils](#) document.

## **Run-time requirements**

ELFI is built on top of the [FUSE](#) (Filesystem is USEr space) system. The current implementation of ELFI can only access the [LFC](#) logical file name catalog and operate on SEs that support the GSI-enabled RFIO.

### **GSI-enabled RFIO**

ELFI does only work with the GSI-enabled version of the RFIO protocol.

A GSI-enabled RFIO server is provided in the EGEE software distribution, in the `DPM-rfio-server` package.

The GSI-enabled RFIO was chosen over the non-secure version because the secure daemon performs Grid to local user mapping via the [LCMAPS](#) library, thus allowing non-site-local access to the files stored in the SE.

*It should be noted that data is anyway transmitted in cleartext over the wire, so no sensitive data should be transmitted with RFIO across local network boundaries.*

## **FUSE**

The *FUSE* (*F*-ilesystem in *USE*-r space) suite provides an interface layer by which a user-space program may respond to Linux kernel VFS requests. Any filesystem-like content may thus be seen as a "real" filesystem on Linux.

FUSE consists of a kernel module and a user-space interface library.

- ◆ FUSE is in mainstream Linux kernel  $\geq 2.6.14$
- ◆ available as an add-on module on 2.4.x and 2.6.x kernels

The FUSE interface has proven itself to be very stable, and is now considered production-quality.

For more information, see <http://fuse.sf.net/>

## **Docutils System Messages**

System Message: ERROR/3 (<string>, line 144); [backlink](#)

Unknown target name: "lfc".

System Message: ERROR/3 (<string>, line 158); [backlink](#)

Unknown target name: "lcmaps".