



**GRID'05
EGEE SUMMER SCHOOL
BUDAPEST**



Practice of GRID DATA HANDLING Part 2
Gfal API handling Exercise

Gabor Hermann

Goal: Use the grid files directly within a C (C++) program.

Temporary restriction: gfal calls for the time being rfio. Consequence: Only grid files on the same site (local file system) can be used because rfio not Grid safe.

An other unpleasant consequence due to rfio: sfm (storage location defined) files must be used in the special form
sfm://<hostname>//<path>

The example c++ program `gfal_example.cpp` uses an arbitrary existing grid file:

Opens it for writing, fills it with a simple content and reopens it for control reading.

Practice:

1. **Let us copy the files** `"gfal_example.cpp"` `"gfal_examlpe.jdl"` `"gfal_exampleComp.sh"` from
`n40.hpcc.sztaki.hu:/home/hermann/NyariIskola/Gfal_example`
in our directory:

1.a
`mkdir <workdir>`

1.b
`cd <workdir>`

1.c
`scp /home/hermann/NyariIskola/Gfal_example/gfal_example.cpp .`
`scp /home/hermann/NyariIskola/Gfal_example/gfal_example.jdl .`
`scp /home/hermann/NyariIskola/Gfal_example/gfal_exampleComp.sh .`

2. **Let us compile the C++ program:**

2.a
`chmod +x gfal_exampleComp.sh`

2.b
`./gfal_exampleComp.sh`

3. **Testing it in the IUF machine:**

3.a Controlling the valid proxy:

```
grid-proxy-info
If its lifetime is too short
```

```
grid-proxy-destroy  
grid-proxy-init
```

3.b make a test file:

```
history > dum
```

3.c Executing the test. **In the argument absolute path is needed!**

```
./gfal_example file:<path_of_workdir>/dum
```

For example:

```
./gfal_example file:/home/speci13/workdir/dum
```

4. Test in the grid

4.1 Lets make a arbitrary GRID file:

4.1.1 Let us find a Storage Element

```
lcg-infosites --vo hunggrid se
```

4.1.2 Let us create a grid file from "dum" using a storage found in the listing of 4.1.1

```
lcg-cr --vo hunggrid -d <SE> file:<path_of_workdir>/dum
```

Please use the KFKI SE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

For example:

```
lcg-cr --vo hunggrid -d grid100.kfki.hu \  
file:/home/speci13/workdir/dum
```

4.1.3 Let us determine the storage address of the grid file upon guid returned by 4.1.2
This value will be used in the "gfal_example.jdl" file.

```
lcg-lr --vo hunggrid <guid_resulted_from_4.1.2>
```

For example:

```
lcg-lr --vo hunggrid guid:ff569696-881e-4e92-bcf0-fae579c49cb1
```

The result may look like the following:

```
sfn://grid100.kfki.hu/storage0/hunggrid/generated/2005-07-06/filefd782043-6d59-4862-8ba7-fb6b3c6c47ec
```

4.2 Let us edit the "gfal_example.jdl" and put the storage address as the value of "Arguments".

!!!!!! There is one fallacy !!!!

Because the implied use of "rfio" the form sfn://<hostname>//<path> must be used

i.e.

For example:

```
>>>-----+-----+  
          |           |  
          v           v
```

```
Arguments="sfn://grid100.kfki.hu//storage0/hungrid/generated/2005-07-06/filefd782043-6d59-4862-8ba7-fb6b3c6c47ec
```

4.3 Let us run the program

4.3.1 Let us investigate where our program may run. The list of CE-s will be stored for example in "res.list"

```
edg-job-list-match -o res.list --vo hungrid gfal_example.jdl
```

4.3.2 Let us submit our program and store the job identifier in file for example in file "job.list01"

```
edg-job-submit -i res.list -o job.list01 gfal_example.jdl
```

4.3.3 Let us observe the state of the job

```
watch "edg-job-status -i job.list01"
```

If the displayed state reaches "done" or "abort" we can interrupt the watch process by "CTRL c"

4.3.4 lets define a directory for the results for example the directory "output":

```
mkdir output
```

4.3.5 let us gather the result:

```
edg-job-get-output --dir ./output -i job.list01
```

The output will be generated in under a generated directory composed by the user name and the long string returned by this command.

4.3.6 Let us see what has happened (listings of standard output, and of standard error files)

```
cd output/<username>_<output of 4.3.5>
```

For example:

```
cd output/speci13_ooDg0_2-B2Sc4WfmV7IFJg/  
ls -al  
cat std.out  
cat std.err
```