

gLite Data Management System Hands-on



Antonio Calanducci

National Institute of Nuclear Physics

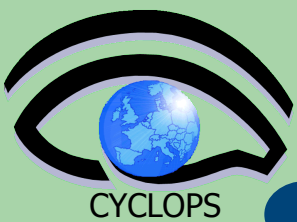
INFN Catania

EGEE NA3 Training & Dissemination

CYCLOPS Second Training Workshop

Chania (Crete), 05th-07th May 2008

Practicals on LFC and lcg-utils



Set up your environment

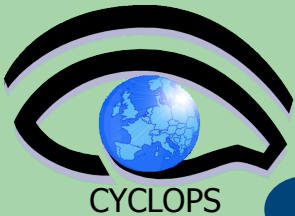
- Check that the following environment variables, used to specify the catalog type and its location, are set up properly:
 - `export LCG_CATALOG_TYPE=lfc`
 - `export LFC_HOST=lfc-gilda.ct.infn.it`
- Ensure you have created a proxy certificate and it is still valid. If not create it by:
 - `voms-proxy-init --voms gilda`





LFC Catalog commands





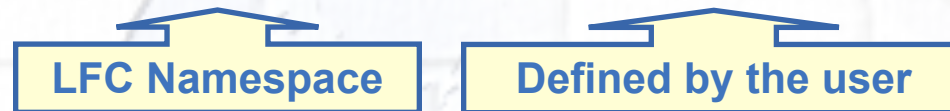
LFC Catalog commands

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*



Ifc-Is examples





lfc-ls examples

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





lfc-ls examples

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

```
...  
drwxrwxr-x 0 117      102          0 Mar 15 16:53 lookee  
drwxrwxr-x 0 122      102          0 Mar 15 16:54 miacartella  
dr--rwxrwx 0 114      102          0 Mar 15 16:50 myfolder  
drwxrwxr-x 0 114      102          0 Mar 15 16:51 myfolder2  
-rwxrwxr-x 1 129      102        193 Mar 15 17:06 nome.txt  
lrwxrwxrwx 1 129      102          0 Mar 15 17:11 nome2.txt -> /  
grid/gilda/tutorial/nome.txt  
drwxrwxr-x 0 129      102          0 Mar 15 16:53 p19  
drwxrwxr-x 3 124      102          0 Mar 15 17:13 pippo  
-rwxrwxr-x 1 124      102        318 Mar 15 17:18 pippopro.txt  
drwxrwxr-x 0 128      102          0 Mar 15 17:00 pollo  
...
```




lfc-ls examples

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

```
...  
drwxrwxr-x 0 117      102          0 Mar 15 16:53 lookee  
drwxrwxr-x 0 122      102          0 Mar 15 16:54 miacartella  
dr--rwxrwx 0 114      102          0 Mar 15 16:50 myfolder  
drwxrwxr-x 0 114      102          0 Mar 15 16:51 myfolder2  
-rwxrwxr-x 1 129      102        193 Mar 15 17:06 nome.txt  
lrwxrwxrwx 1 129      102          0 Mar 15 17:11 nome2.txt -> /  
grid/gilda/tutorial/nome.txt  
drwxrwxr-x 0 129      102          0 Mar 15 16:53 p19  
drwxrwxr-x 3 124      102          0 Mar 15 17:13 pippo  
-rwxrwxr-x 1 124      102        318 Mar 15 17:18 pippopro.txt  
drwxrwxr-x 0 128      102          0 Mar 15 17:00 pollo  
...
```

```
$ export LFC_HOME=/grid/gilda/tutorial  
$ lfc-ls -l nome.txt
```





lfc-ls examples

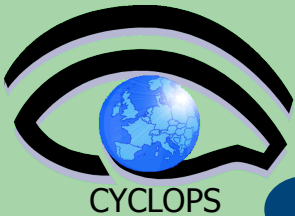
```
$ lfc-ls -l /grid/gilda/tutorial
```

```
...
drwxrwxr-x  0 117      102          0 Mar 15 16:53 lookee
drwxrwxr-x  0 122      102          0 Mar 15 16:54 miacartella
dr--rwxrwx  0 114      102          0 Mar 15 16:50 myfolder
drwxrwxr-x  0 114      102          0 Mar 15 16:51 myfolder2
-rwxrwxr-x  1 129      102        193 Mar 15 17:06 nome.txt
lrwxrwxrwx  1 129      102          0 Mar 15 17:11 nome2.txt -> /
grid/gilda/tutorial/nome.txt
drwxrwxr-x  0 129      102          0 Mar 15 16:53 p19
drwxrwxr-x  3 124      102          0 Mar 15 17:13 pippo
-rwxrwxr-x  1 124      102       318 Mar 15 17:18 pippopro.txt
drwxrwxr-x  0 128      102          0 Mar 15 17:00 pollo
...
```



```
$ export LFC_HOME=/grid/gilda/tutorial
$ lfc-ls -l nome.txt
```

```
-rwxrwxr-x  1 129      102        193 Mar 15 17:06 /grid/gilda/tutorial/nome.txt
```



LFC Catalog commands





LFC Catalog commands

Creating a symbolic link

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

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

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

– *Example:*

```
$ lfc-ln -s /grid/gilda/tutorial/prova.txt /  
grid/gilda/tutorial/linkToProva.txt
```

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

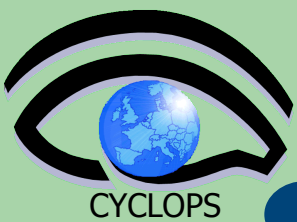
```
$ lfc-ls -l /grid/gilda/tutorial
```

```
lrwxrwxrwx  1 101      102      0 Apr 03 11:42 linkToProva.txt -> /  
grid/gilda/tutorial/prova.txt
```





LFC Catalog commands



LFC Catalog commands

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 be already created in the catalog
- Examples:

```
$ lfc-mkdir /grid/gilda/tutorial/
```

Examples

You can just check the directory with:

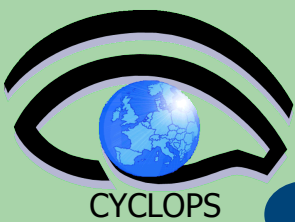
```
$ lfc-ls -l /grid/gilda/tutorial
```





LFC Catalog commands





LFC Catalog commands

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

lfc-delcomment deletes a comment previously added

- Example:

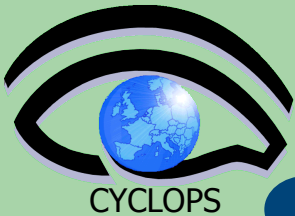
```
lfc-setcomment /grid/gilda/tutorial/hello.txt  
'Hello World!'
```

- Check your job with

```
lfc-ls --comment /grid/gilda/tutorial/hello.txt
```

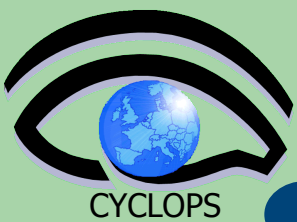
```
/grid/gilda/catania hello.txt Hello World!
```





LFC Catalog commands





LFC Catalog commands

- Example:

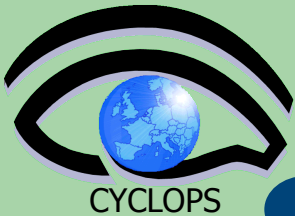
```
lfc-delcomment /grid/gilda/tutorial/  
hello.txt
```

- Check if it was successful with

```
lfc-ls -l --comment /grid/gilda/tutorial/  
hello.txt
```

```
-rw-rw-r-- 1 101 102 13 Apr 03 11:23 /grid/gilda/catania/  
hello.txt
```





Hands-on Session

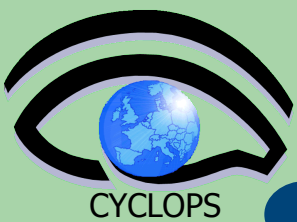




Hands-on Session

Exercise No.1:





Hands-on Session

Exercise No.1:

- Log into the UI and initialize your proxy credentials if not already done

Hands-on Session

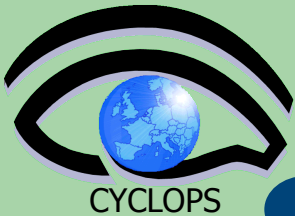
Exercise No.1:

- Log into the UI and initialize your proxy credentials if not already done
- check that your environment variables are properly set up to use the GILDA catalog

Hands-on Session

Exercise No.1:

- Log into the UI and initialize your proxy credentials if not already done
- check that your environment variables are properly set up to use the GILDA catalog
- have a look inside the catalog



Hands-on Session

Exercise No.1:

- Log into the UI and initialize your proxy credentials if not already done
- check that your environment variables are properly set up to use the GILDA catalog
- have a look inside the catalog
- create a sub-directory, under `/grid/gilda/chania`, with your name



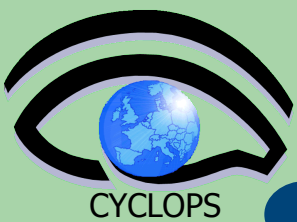


Hands-on Session

Exercise No.1:

- Log into the UI and initialize your proxy credentials if not already done
- check that your environment variables are properly set up to use the GILDA catalog
- have a look inside the catalog
- create a sub-directory, under `/grid/gilda/chania`, with your name
- put inside the just created dir a link to an existing file (from files in other directories)



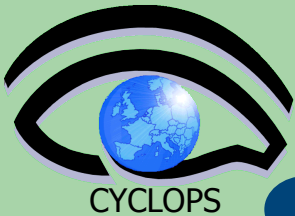


Hands-on Session

Exercise No.1:

- Log into the UI and initialize your proxy credentials if not already done
- check that your environment variables are properly set up to use the GILDA catalog
- have a look inside the catalog
- create a sub-directory, under `/grid/gilda/chania`, with your name
- put inside the just created dir a link to an existing file (from files in other directories)
- add a comment to that file and verify it





LFC Catalog commands

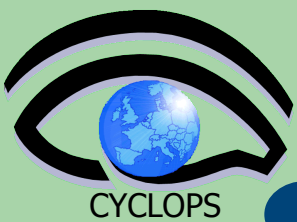
Summary of the LFC Catalog commands

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



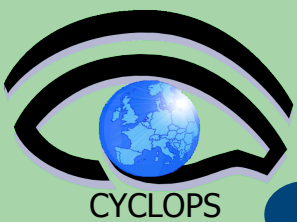
lcg-utils





lcg-utils

- The LCG Data Management tools (usually called ***lcg-utils***) allow users to copy files between UI, CE, WN and a SE, to register entries in the File Catalogs and replicate files between SEs.



lcg-utils

- The LCG Data Management tools (usually called ***lcg-utils***) allow users to copy files between UI, CE, WN and a SE, to register entries in the File Catalogs and replicate files between SEs.



lcg-utils

- The LCG Data Management tools (usually called ***lcg-utils***) allow users to copy files between UI, CE, WN and a SE, to register entries in the File Catalogs and 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=glite-rb.ct.infn.it:2170`





lcg-utils: lcg-cr



lcg-utils: lcg-cr

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

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



lcg-utils: lcg-cr



CYCLOPS

lcg-utils: lcg-cr

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

```
lcg-infosites --vo gilda se
```

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

- lcg-cr usage example:



CYCLOPS

lcg-utils: lcg-cr

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

```
lcg-infosites --vo gilda se
```

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

- lcg-cr usage example:

```
$ lcg-cr -v -d trigriden01.unime.it -l lfn:/grid/gilda/tutorial/note.txt --vo
gilda file:///home/tcaland/note.txt
Using grid catalog type: lfc
Using grid catalog : lfc.trigrid.it
Source URL: file:///home/tcaland/note.txt
File size: 279
VO name: gilda
Destination specified: life006.dft.unipa.it
Destination URL for copy: gsiftp://trigriden01.unime.it/flatfiles/SE00/gilda/generated/2006-04-03/file2cc930b1-
c61f-4747-b55a-bc521c93622b
# streams: 1
# set timeout to 0 seconds
Alias registered in Catalog: lfn:/grid/gilda/tutorial/note.txt
      279 bytes      0.12 KB/sec avg      0.12 KB/sec inst
Transfer took 29520 ms
Destination URL registered in Catalog: sfn://trigriden01.unime.it/flatfiles/SE00/gilda/generated/2006-04-03/
file2cc930b1-c61f-4747-b55a-bc521c93622b
guid:aa5952f7-27b8-4acc-b6ee-0031af0a0f19
```



lcg-utils: lcg-aa and lcg-la



lcg-utils: lcg-aa and lcg-la

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 to
- **lfn** specifies the new alias

- *Example:*

```
$ lcg-aa --vo gilda guid:aa5952f7-27b8-4acc-  
b6ee-0031af0a0f19 lfn:/grid/gilda/tutorial/  
aliasToNote.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/tutorial/  
aliasToNote.txt
```

```
lfn:/grid/gilda/tutorial/note.txt
```

```
lfn:/grid/gilda/tutorial/aliasToNote.txt
```




Hands-on session





Hands-on session

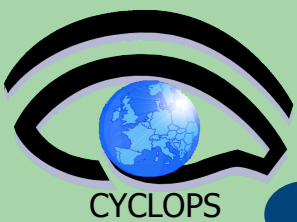
Exercise No.2:



Hands-on session

Exercise No.2:

- verify that your **LCG_GFAL_INFOSYS** is correctly set up



Hands-on session

Exercise No.2:

- verify that your **LCG_GFAL_INFOSYS** is correctly set up
- create a dummy file

Hands-on session

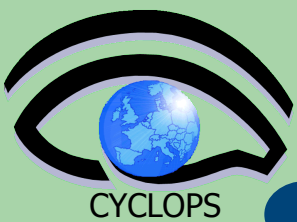
Exercise No.2:

- verify that your **LCG_GFAL_INFOSYS** is correctly set up
- create a dummy file
- check the available storage elements

Hands-on session

Exercise No.2:

- verify that your **LCG_GFAL_INFOSYS** is correctly set up
- create a dummy file
- check the available storage elements
- copy and register the previous created file into your previously created dir



Hands-on session

Exercise No.2:

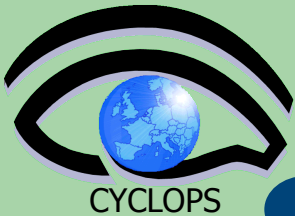
- verify that your **LCG_GFAL_INFOSYS** is correctly set up
- create a dummy file
- check the available storage elements
- copy and register the previous created file into your previously created dir
- add an alias to the just uploaded file



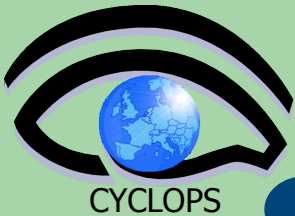
Hands-on session

Exercise No.2:

- verify that your **LCG_GFAL_INFOSYS** is correctly set up
- create a dummy file
- check the available storage elements
- copy and register the previous created file into your previously created dir
- add an alias to the just uploaded file
- check if the alias was assigned correctly



lcg-utils commands for replicas (I)



lcg-utils commands for replicas (I)

Copying a file from one SE to another one and register its replica 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 sfm:// 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 sfm: for a classical SE or srm:



lcg-utils commands for replicas (I)

Copying a file from one SE to another one and register its replica 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 grid009.ct.infn.it --vo gilda lfn:/grid/gilda/tutorial/note.txt
```

```
Using grid catalog type: lfc
```

```
Using grid catalog : lfc.trigrid.it
```

```
Source URL: lfn:/grid/gilda/tutorial/note.txt
```

```
File size: 279
```

```
VO name: gilda
```

```
Destination specified: grid009.ct.infn.it
```

```
Source URL for copy: gsiftp://gildase.oact.inaf.it/flatfiles/SE00/gilda/generated/2006-04-03/  
file02c844ad-1299-44db-a347-75ed3003c11e
```

```
Destination URL for copy: gsiftp://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006-04-03/  
file1e0415e0-3486-42bc-947a-824451f69dc4
```

```
# streams: 1
```

```
# set timeout to 0
```

```
0 bytes      0.00 KB/sec avg      0.00 KB/sec inst
```

```
Transfer took 2020 ms
```

```
Destination URL registered in LRC: sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006-04-03/  
file1e0415e0-3486-42bc-947a-824451f69dc4
```



lcg-utils commands for replicas (II)



lcg-utils commands for replicas (II)

Listing of replicas for a given LFN, GUID or SURL

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

where

- **vo_name** specifies the Virtual Organization the user belongs to
- **file** specifies the Logical File Name, the Grid Unique Identifier or the Site URL. An SURL scheme can be sfn: for a classical SE or srm:

- Example:

```
$ lcg-lr --vo gilda lfn:/grid/gilda/tutorial/  
note.txt
```

```
sfn://gildase.oact.inaf.it/flatfiles/SE00/gilda/generated/2006-04-03/file02c844ad-1299-44db-  
a347-75ed3003c11e
```

```
sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006-04-03/  
file1e0415e0-3486-42bc-947a-824451f69dc4
```

```
sfn://life006.dft.unipa.it/flatfiles/SE00/gilda/generated/2006-04-03/file2cc930b1-c61f-4747-  
b55a-bc521c93622b
```

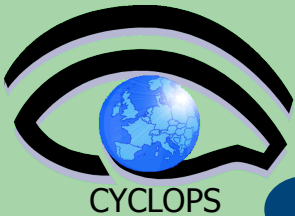
we can have the same output using the GUID:

```
$ lcg-lr --vo gilda guid:aa5952f7-27b8-4acc-  
b6ee-0031af0a0f19
```





lcg-utils commands for replicas (III)



lcg-utils commands for replicas (III)

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. An 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/tutorial/note.txt
```

- delete all the replicas

```
lcg-del -a --vo gilda lfn:/grid/gilda/tutorial/  
note.txt
```

- let's check if the previous command was successful

```
lcg-lr --vo gilda lfn:/grid/gilda/tutorial/note.txt
```

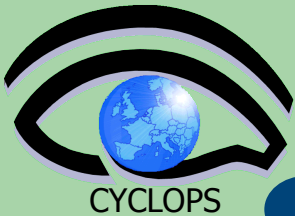
lcg_lr: No such file or directory

- or by `lfs-ls /grid/gilda/tutorial` (you will not see anymore note.txt and its alias)





lcg-utils: lcg-cp



lcg-utils: lcg-cp

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. An 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 --vo gilda lfn:/grid/gilda/tutorial/note.txt file:/home/  
claudio/note2.txt
```

```
Using grid catalog type: lfc
```

```
Using grid catalog : lfc.trigrid.it
```

```
Source URL: lfn:/grid/gilda/tutorial/note.txt
```

```
File size: 279
```

```
VO name: gilda
```

```
Source URL for copy: gsiftp://gildase.oact.inaf.it/flatfiles/SE00/  
gilda/generated/2006-04-03/file02c844ad-1299-44db-  
a347-75ed3003c11e
```

```
Destination URL: file:/home/claudio/note2.txt
```

```
# streams: 1
```

```
# set timeout to 0 (seconds)
```

```
0 bytes
```

```
0.00 KB/sec avg
```

```
0.00 KB/sec inst
```

```
Transfer took 1030 ms
```



Hands-on session

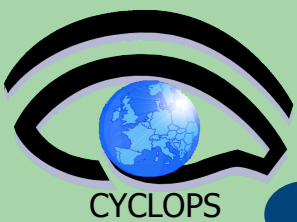




Hands-on session

Exercise No.3:



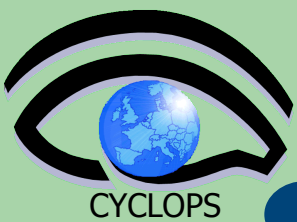


Hands-on session

Exercise No.3:

- Create two replicas of the file you previously uploaded (you could also use the alias to point it out)



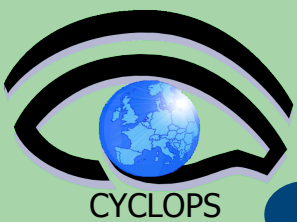


Hands-on session

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



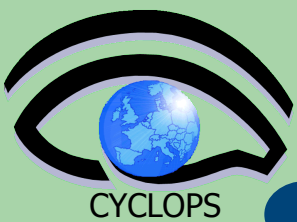


Hands-on session

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 in your UI





Hands-on session

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 in your UI
- Delete just one replica and verify that

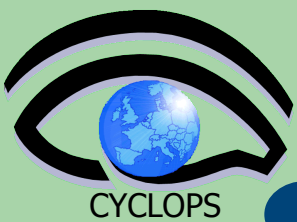


Hands-on session

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 in your UI
- Delete just one replica and verify that
- Delete all the replicas and verify that



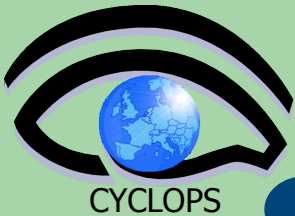


Hands-on session

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 in your UI
- Delete just one replica and verify that
- Delete all the replicas and verify that
- Verify if the entry is still into the catalog





Handling entries permissions (I)





Handling entries permissions (I)

Get LFC entries (file/directories) permissions and ACLs





Handling entries permissions (I)

Get LFC entries (file/directories) permissions and ACLs

`lfc-getacl path...`





Handling entries permissions (I)

Get LFC entries (file/directories) permissions and ACLs

`lfc-getacl path...`





Handling entries permissions (I)

Get LFC entries (file/directories) permissions and ACLs

```
lfc-getacl path...
```

where path is a LFC file or directory



Handling entries permissions (I)

Get LFC entries (file/directories) permissions and ACLs

```
lfc-getacl path...
```

where path is a LFC file or directory

Example:

```
$ lcg-getacl /grid/gilda/tutorial
```

```
# file: /grid/gilda/tutorial
```

```
# owner: /C=IT/O=GILDA/OU=Personal Certificate/L=CATANIA/  
CN=CATANIA01/Email=roberto.barbera@ct.infn.it
```

```
# group: gilda
```

```
user::rwx
```

```
group::rwx
```

```
#effective:rwx
```

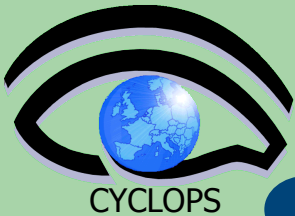
```
other::r-x
```

```
default:user::rwx
```

```
default:group::rwx
```

```
default:other::r-x
```





Handling entries permissions (II)





Handling entries permissions (II)

Change the permission of a file/directory

```
lfc-chmod absolute_mode path...
```

where

- **absolute_path** is the permission pattern expressed in octal mode
- **path** is a LFC file or directory
- Example:

```
$ lfc-chmod 750 /grid/gilda/tutorial/note.txt
$ lfc-getacl /grid/gilda/tutorial/note.txt
```

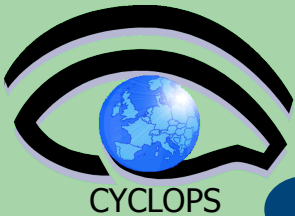
```
# file: /grid/gilda/tutorial/note.txt
# owner: /C=IT/O=GILDA/OU=Personal Certificate/
L=INFN      Catania/CN=Tony Calanducci/
Email=tony.calanducci@ct.infn.it
# group: gilda
user::rwx
group::r-x      #effective:r-x
other::---
```





Setting ACLs (I)





Setting ACLs (I)

Set LFC directory/file access control list

```
lfc-setacl [-d] [-m] [-s] acl_entries path...
```

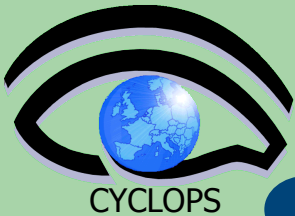
where

- *acl_entries* is a comma separated list of entries. Each entry has colon separated fields: ACL type, id (uid or gid), permission.

The entries look like:

```
user::perm  
user:uid:perm  
group::perm  
group:gid:perm  
mask:perm  
other:perm  
default:user::perm  
default:user:uid:perm  
default:group::perm  
default:group:gid:perm  
default:mask:perm  
default:other:perm
```

The ACL type can be abbreviated with the first letter



Setting ACLs (I)

Set LFC directory/file access control list

```
lfc-setacl [-d] [-m] [-s] acl_entries path...
```

where

- *acl_entries* is a comma separated list of entries. Each entry has colon separated fields: ACL type, id (uid or gid), permission.

The entries look like:

user::perm

user:uid:perm

group::perm

group:gid:perm

mask:perm

other:perm

default:user::perm

default:user:uid:perm

default:group::perm

default:group:gid:perm

default:mask:perm

default:other:perm

•user and group -> owner
permission



The ACL type can be abbreviated with the first letter



Setting ACLs (I)

Set LFC directory/file access control list

```
lfc-setacl [-d] [-m] [-s] acl_entries path...
```

where

- *acl_entries* is a comma separated list of entries. Each entry has colon separated fields: ACL type, id (uid or gid), permission.

The entries look like:

user::perm

user:uid:perm

group::perm

group:gid:perm

mask:perm

other:perm

default:user::perm

default:user:uid:perm

default:group::perm

default:group:gid:perm

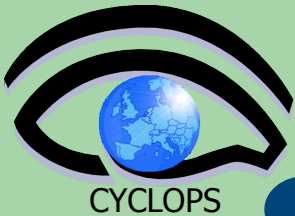
default:mask:perm

default:other:perm

•user and group -> owner
permission

•user:gid and group:gid -> specific
users and groups

The ACL type can be abbreviated with the first letter



Setting ACLs (I)

Set LFC directory/file access control list

```
lfc-setacl [-d] [-m] [-s] acl_entries path...
```

where

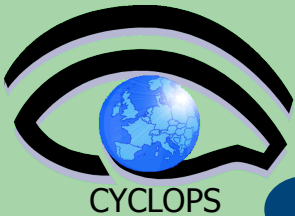
- *acl_entries* is a comma separated list of entries. Each entry has colon separated fields: ACL type, id (uid or gid), permission.

The entries look like:

```
user::perm  
user:uid:perm  
group::perm  
group:gid:perm  
mask:perm  
other:perm  
default:user::perm  
default:user:uid:perm  
default:group::perm  
default:group:gid:perm  
default:mask:perm  
default:other:perm
```

- **user** and **group** -> owner permission
- **user:gid** and **group:gid** -> specific users and groups
- **mask** -> maximum permissions to specific users/groups

The ACL type can be abbreviated with the first letter



Setting ACLs (I)

Set LFC directory/file access control list

```
lfc-setacl [-d] [-m] [-s] acl_entries path...
```

where

- *acl_entries* is a comma separated list of entries. Each entry has colon separated fields: ACL type, id (uid or gid), permission.

The entries look like:

```
user::perm
user:uid:perm
group::perm
group:gid:perm
mask:perm
other:perm
default:user::perm
default:user:uid:perm
default:group::perm
default:group:gid:perm
default:mask:perm
default:other:perm
```

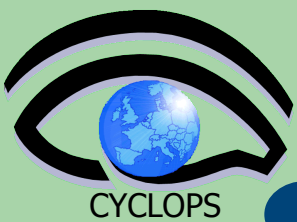
- **user** and **group** -> owner permission
- **user:gid** and **group:gid** -> specific users and groups
- **mask** -> maximum permissions to specific users/groups
- **default** permissions (only for a directory) are inherited by the files or sub-directories of that dir

The ACL type can be abbreviated with the first letter



Setting ACLs (II)





Setting ACLs (II)

- **path** specifies the LFC pathname
- **uid** can be given as the DN or the corresponding numeric id.
- **gid** can be given as the VO name or the corresponding numeric id.
- **perm** can be expressed as a combination of characters rwx- or as a value between 0 and 7.

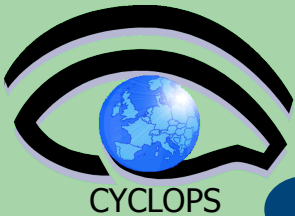
OPTIONS

- d** remove ACL entries. The "perm" field is ignored.
- m** modify existing ACL entries or add new entries.
- s** set the ACL entries. The complete set of ACL entries is replaced.





ACLs example



ACLs example

Example:

```
$ lfc-setacl -m u:'/C=IT/O=GILDA/OU=Personal  
Certificate/L=INFN/CN=Diego Scardaci/  
Email=diego.scardaci@ct.infn.it':rwx,g:tri  
grid:r-x,m:rwx /grid/gilda/tutorial/  
note.txt
```

```
$ lfc-getacl /grid/gilda/tutorial/note.txt
```

```
# file: /grid/gilda/tutorial/note.txt
```

```
# owner: /C=IT/O=GILDA/OU=Personal Certificate/L=INFN  
Catania/CN=Tony Calanducci/  
Email=tony.calanducci@ct.infn.it
```

```
# group: gilda
```

```
user::rwx
```

```
user:/C=IT/O=GILDA/OU=Personal Certificate/L=INFN/  
CN=Diego Scardaci/Email=diego.scardaci@ct.infn.it:rwx  
#effective:rwx
```

```
group::r-x #effective:r-x
```

```
group:trigrid:r-x #effective:r-x
```

```
mask::rwx
```

```
other::---
```

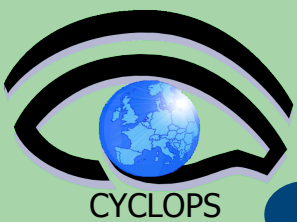


Exercise No. 4



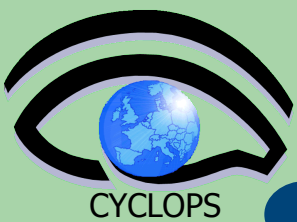
Exercise No. 4

- Create an entry in your working FC directory. Take a look to the default permission of it.



Exercise No. 4

- Create an entry in your working FC directory. Take a look to the default permission of it.
- Change the permission in a way that only you (the owner) can delete the entry



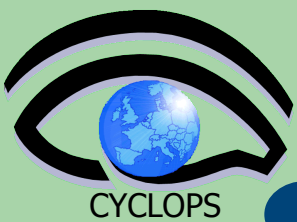
Exercise No. 4

- Create an entry in your working FC directory. Take a look to the default permission of it.
- Change the permission in a way that only you (the owner) can delete the entry
- Ask your neighbor to delete that entry (he should not be able)



Exercise No. 4

- Create an entry in your working FC directory. Take a look to the default permission of it.
- Change the permission in a way that only you (the owner) can delete the entry
- Ask your neighbor to delete that entry (he should not be able)
- Ask again your neighbor his DN and add it as an ACL entry granting full permission.



Exercise No. 4

- Create an entry in your working FC directory. Take a look to the default permission of it.
- Change the permission in a way that only you (the owner) can delete the entry
- Ask your neighbor to delete that entry (he should not be able)
- Ask again your neighbor his DN and add it as an ACL entry granting full permission.
- Ask now to delete it (it should be able to do that now) – delete with `lcg-del -s STORAGEHOST`