



Programming API(s) for gLite services

Giuseppe LA ROCCA

INFN Catania

giuseppe.larocca@ct.infn.it

Joint EGEE and SEE-GRID Summer School
on Grid Application Support
Budapest – Hungary, June 25–30, 2007



dkfz.



cilea

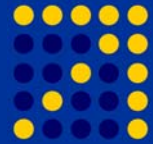


Information Society
and Media





- **Overview of the WMPProxy Java API**
- **Overview of the SEE-GRID File Management Java API**
- **Overview of the GFAL Java API**



WMPProxy Java API ver 3.1.0

Overview of the WMPProxy Java API
Software Requirements
Class `org.glite.wms.wmproxy.WMPProxyAPI`
An use case



- **The WMPProxy is the service responsible to provide access to the WMS functionality through a Web Service Interface**
- **The gLite WMPProxy Server can be either accessed directly through the published WSDL, the C++ command line interface, or the API**
- **In these two links you can find a guide about how to interact with the main services exposed by the WMPProxy through the provided API Java and C++ API**

<https://grid.ct.infn.it/twiki/bin/view/GILDA/ApiJavaWMPProxy>

<https://grid.ct.infn.it/twiki/bin/view/GILDA/WMPProxyCPPAPI>



Installation guide API JAVA /1

- **0.) Start from a fresh install of Scientific Linux CERN**
 - other platforms are also compatible but yet not properly tested
- **1.) Install needed rpm(s) following versions:**
 - take the rpm(s) as usual from <http://glite.web.cern.ch/glite/packages/externals/bin/rhel30/RPMS>



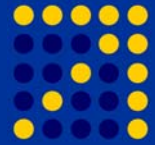
Installation guide API JAVA /2

- Axis (<http://ws.apache.org/axis/>)

where the needed jars available are:

axis-ant.jar
axis.jar
jaxrpc.jar
commons-discovery-0.2.jar
commons-logging-1.0.4.jar
log4j-1.2.8.jar
log4j.properties
saaj.jar
wsdl4j-1.5.1.jar

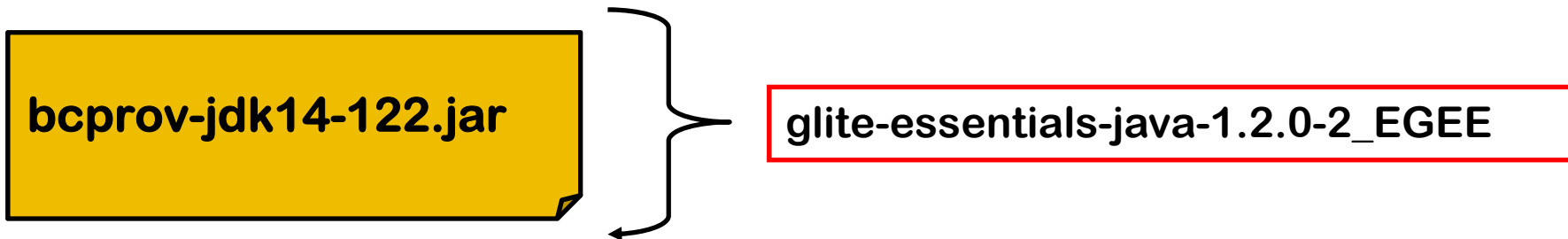
dcache-client-1.7.0-31.i586.rpm



Installation guide API JAVA /3

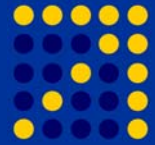
- BouncyCastle (<http://www.bouncycastle.org/>)

where the needed jar is:



W A R N I N G

Make sure all the listed jar(s) are included in your CLASSPATH environment variable !!



Installation guide API JAVA /4

- 2.) Install the following component(s) from the latest gLite release (<http://glite.web.cern.ch/glite/packages>):

glite-security-test-utils
glite-security-util-java
glite-security-delegation-java
glite-security-trustmanager
glite-wms-wmproxy-api-java



- **Verify that your User Interface (UI) or workstation contains the following rpm package(s):**

**glite-wms-wmproxy-api-java-3.1.1-1_N20061202.noarch.rpm
glite-wms-ui-api-java-3.1.1-1_N20061202.i386.rpm
glite-security-util-java-1.3.8-1_N20061202.noarch.rpm
glite-security-trustmanager-1.8.8-1_N20061202.noarch.rpm
glite-security-delegation-java-1.4.1-1_N20061202.noarch.rpm**

- **If these rpm package(s) are not installed you can downloaded them from this web page**

<http://lxb2071.cern.ch:8080/etics/index.jsp>



- **Download and extract, from here**
<http://glite.web.cern.ch/glite/packages/R3.1/N20060723/src/>

the two tarball :

glite-jdl-api-java-3.1.1_bin.tar.gz

glite-wms-wmproxy-api-java-3.1.0_src.tar.gz



- **The main Java class of this API is:**
org.glite.wms.wmproxy.WMProxyAPI;
- **A client object can be created using one of these following constructor(s):**

```
public WMProxyAPI(String url, String proxyFile)
```

```
public WMProxyAPI(String url, String proxyFile, String  
CAcerts)
```

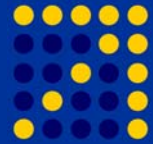
```
public WMProxyAPI(String url, InputStream  
proxyStream)
```

```
public WMProxyAPI(String url, InputStream  
proxyStream, String CAcerts)
```



where:

- **url**: the WMPProxy server URL to be contacted
(e.g. `https://<host>:<port>/glite_wms_wmproxy_server`);
- **proxyFile**: the pathname to a valid user proxy;
for the default value (`/tmp/x509up_u<UID>`)
- **CAcerts**: the path of a local CA directory
- **proxyStream**: a valid proxy passed as input stream;



Therefore, using one of the four constructors

```
WMPProxyAPI Client = new WMPProxyAPI( ... );
```

a new instance is created. This instance will be created to invoke any other WMPProxy services.

For example:

```
WMPProxyAPI Client = new WMPProxyAPI(  
    “https://trinity.datamat.it:7443/glite_wms_wmproxy_server”,  
    “/x509up_u504”);
```

```
jobIDs = Client.jobSubmit(jdlString, delegationID);
```



- In case of failure, one of the following exceptions is thrown:

org.glite.wms.wmproxy.AuthenticationFaultType

org.glite.wms.wmproxy.JobUnknownFaultType

org.glite.wms.wmproxy.InvalidArgumentFaultType

org.glite.wms.wmproxy.NoSuitableResourcesFaultType

org.glite.wms.wmproxy.GetQuotaManagementFaultType

org.glite.wms.wmproxy.ProxyFileException

org.glite.wms.wmproxy.GrstDelegationException

org.glite.wms.wmproxy.GenericFaultType

org.gridsite.www.namespaces.delegation_1.DelegationExceptionType



- **The JDL to be matched must contains some mandatory Requirements and Rank attributes;**
 - when using command line interface these are added by default by the UI, but in our case user has to take care to check if his/her JDL file contains them:

Requirements = (other.GlueCEStateStatus == "Production");

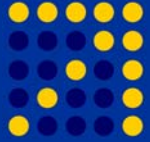
Rank = (-other.GlueCEStateEstimatedResponseTime);



- The client must be properly authorized when interacts with the WMPProxy service.
 - This means that either the FQAN or the DN (in case of globus-style proxies) of the client must be properly listed and authorized in the *glite_wms_wmproxy.gacl* file on the WMPProxy machine.

```
[root@glite-rb2 etc]# cat glite_wms_wmproxy.gacl

<gacl version='0.0.1'>
  <entry>
    <voms><fqan>bio/Role=NULL</fqan></voms>
    <allow><exec/></allow>
  </entry>
</gacl>
```

Delegation of user credential

- Before calling submission or list match services, *client proxy credentials* are requested to be transferred from the client to the server through a *delegation process*.
- The following Java code performs this delegation process :

```
String delegationId = "larocca";
```

```
WMPProxyAPI client = new WMPProxyAPI  
wmp_proxy_url,user_proxy,CACertsPath);
```

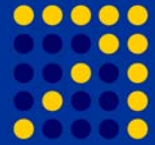
```
/*Get a proxy identified by the delegationID String*/
```

```
proxy = client.grstGetProxyReq (delegationId);
```

```
/* Allows delegating user's credential to the WMPProxy */
```

```
client.grstPutProxy(delegationId, proxy);
```

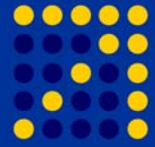
The obtained *delegation identifier* can be reused for several calls to the WMPProxy server;



Difference between submission and registration /1

- Submission of simple jobs, DAGs, collections and parametric jobs to the WMPProxy service requires as input a job description file in which job characteristics and requirements are expressed by means of Condor class-ad-like expressions.
- In this description the users can specify some files that are needed by the submitted jobs during the execution on the remote Computing Elements. These files are listed in the JDL InputSandbox attribute specifying for each of them the complete URI of its location.
- If the job does NOT have any file in the InputSandbox to be transferred from the submitting machine to the WMPProxy node, the submission can be performed also calling the **jobSubmit** service:

```
jobIds = client.jobSubmit(jdlString, delegationId);
```



Otherwise, these following steps are needed:

- a preliminary server registration:

```
jobId = client.jobRegister(jdlString, delegationId);
```

- transfer of files in the InputSandbox from the client machine to the WMS node;
- call the jobStart service to trigger the submission:

```
client.jobStart(jobId);
```



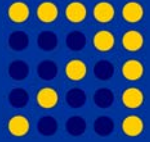
Software Requirements /3

- In order to allow grid user to invoke, within Java code, the **UrlCopy** class to copy files from the WMS Server to the UI and vice versa the following Java package is requested

cog-jglobus-1.4-bin.tar.gz

The above package, with all the needed documentation, can be downloaded from this web page http://dev.globus.org/wiki/CoG_JGlobus_1.4





```
Executable = "/bin/sh";  
Arguments = "start_hostname.sh";  
StdOutput = "hostname.out";  
StdError = "hostname.err";  
InputSandbox = {"start_hostname.sh"};  
OutputSandbox = {"hostname.err", "hostname.out"};  
Requirements = (other.GlueCEStateStatus == "Production");  
Rank = (-other.GlueCEStateEstimatedResponseTime);  
RetryCount = 3;
```

hostname.jdl

```
#!/bin/sh  
sleep 5  
hostname -f
```

start_hostname.sh



Submit a new request to WMPProxy /1

BioinfoGRID

```
[larocca@glite-tutor:~/API]$ java WMPProxyGetProxyAndSubmit
```

```
WMPProxyGetProxyAndSubmit.java
```

```
"A simple client to interact with the WMPProxy Server."
```

```
Author: Giuseppe La Rocca (giuseppe.larocca@ct.infn.it)
```

```
I.N.F.N. - Sez. of Catania - ITALY
```

```
Via S.Sofia, 64 - 95123 Catania
```

```
Phone: +39.095.378.53.74
```

Usage :

```
java WMPProxyGetProxyAndSubmit -h[elp]
```

```
java WMPProxyGetProxyAndSubmit <user_proxy> <delegation_id> <wmpproxy_server> <InputSandboxFiles>  
                                <jdl_file> <CAcertsPath> [CAs paths (optional)]
```

where:

```
<user_proxy>          ... the file containing the user's credentials
```

```
<delegation_id>      ... the string used to save the user's delegation
```

```
<wmpproxy_server>    ... the entry point of the WMPProxy Server to contact  
                    (e.g. : https://glite-rb3.ct.infn.it:7443/glite_wms_wmpproxy_server)
```

```
<InputSandboxFiles> ... The list of file(s) to transfer to the WMPProxy Server
```

```
<jdl_file>           ... the jdl file to submit to the grid
```

```
<CAcertsPath>       ... the path location of the directory containing all the Certificate  
                    Authorities files
```

```
Contacting... https://glite-rb2.ct.infn.it:7443/glite_wms_wmpproxy_server with the proxy..  
/tmp/x509up_u512
```

```
Your job has been successfully submitted.
```

```
jobID = [ https://glite-rb2.ct.infn.it:9000/XAoY7FZg LJjgCp4U9grsBw ]
```



Submit a new request to WMPProxy /2

BioinfoGRID

```
for (int index = 0; index < InputSandboxFiles.length; index++)
{
    String toURL = front + "2811" + rear;
    toURL = toURL + "/" + InputSandboxFiles[index];
    fromURL = "file:/// " + InputSandboxFiles[index];

    try {
        GlobusURL from = new GlobusURL(fromURL);
        GlobusURL to = new GlobusURL(toURL);

        UrlCopy uCopy = new UrlCopy();
        uCopy.setDestinationUrl(to);
        uCopy.setSourceUrl(from);
        uCopy.setUseThirdPartyCopy(true);

        uCopy.copy();
    } catch (Exception e) {System.err.println(e.getMessage());}
}
```

**Specify the Destination
and Source URL(s)**

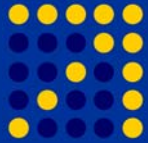
**Copy file(s) from the UI to
the Resource Broker**

The script, thanks to the **UrlCopy** Class, performs the copy of the InputSandbox files to the reduced path of the WMS as you can see:

```
[root@glite-rb2 root]# cd /var/glite/SandboxDir/XA/
```

```
[root@glite-rb2 root]# ll https_3a_2f_2fglite-  
rb2.ct.infn.it_3a9000_2fXAoY7FZgJgC4U9grsBw/input/
```

```
-rwxrwxr-x 1 gilda001 glite 30 Jan 11 09:05 start_hostname.sh
```



Retrieve output file(s)

With the job finishes you can retrieve the output file(s) as follow:

```
[larocca@glite-tutor:~/API]$ java WMPProxyGetOutputAndPurge
Usage :
  java WMPProxyGetOutputAndPurge -h[elp]
  java WMPProxyGetOutputAndPurge <user_proxy> <wmproxy_server> <jobId>
                                     <dirPath> <CAcertsPath> [CAs paths (optional)]
+-----+
WMPProxy URL = [https://glite-rb2.ct.infn.it:7443/glite_wms_wmproxy_server]
proxyFile    = [/tmp/x509up_u512]
JobID        = [https://glite-rb2.ct.infn.it:9000/XAoY7FZgLJjgCp4U9grsBw]
dirPath      = [/home/larocca/API/]
CA certs     = [/etc/grid-security/certificates/]
+-----+
```

```
Start downloading output file(s)..
file n. 1
```

```
-----
name = [gsiftp://glite-rb2.ct.infn.it:2811/var/glite/SandboxDir/XA/https_3a_2f_2fglite-
rb2.ct.infn.it_3a9000_2fXAoY7FZgLJjgCp4U9grsBw/output/hostname.err]
size = [0]
```

```
file n. 2
```

```
-----
name = [gsiftp://glite-rb2.ct.infn.it:2811/var/glite/SandboxDir/XA/https_3a_2f_2fglite-
rb2.ct.infn.it_3a9000_2fXAoY7FZgLJjgCp4U9grsBw/output/hostname.out]
size = [28]
```

**List of file(s) retrieved from to the
Resource Broker to the user's
account**



API Documentation

<http://trinity.datamat.it/projects/EGEE/wiki/apidoc/3.1/htmljava/index.html>



Datamat – WMPProxy quickstart

<http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=WMPProxyClient.QuickStart>



JDL Attributes guide for WMPProxy

<https://edms.cern.ch/document/590869/1>



WMPProxy user guide

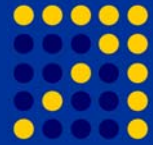
<https://edms.cern.ch/document/674643/1>



- Overview of the WMPProxy Java API

- Overview of the SEE-GRID File Management Java API

- Overview of the GFAL Java API

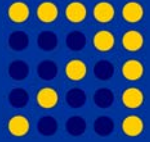


SEE-GRID File Management Java API ver 1.2

Overview of the SEE-GRID File Management Java API
Installation & Configuration
The Classes : LFCDataStorage, LFCDirectoryItem and LFCFileItem
Some different examples



- **SEE-GRID File Management Java API supports most of the data management operations offered by LFC and LCG_UTILS C APIs.**
- **These Java API are compatible with LCG 2.7.x and gLite grid middleware.**



Available feature(s)

- **Uploading / Downloading files from UI to grid**
- **Reading file information: permissions, GUID, comment**
- **Listing file replicas**
- **File management operations:**
 - replicating files, deleting files and file replicas, unregistering files and file replicas
- **Reading directory information**
 - Permissions, comment
- **Listing directory contents**
- **Reading user and group ids for files and directories**
- **Reading user and group names for files and directories**
- **Reading filemode information as described in LFC C/C++ API manual**
- **Listing file/directory aliases**
- **Alias management operations (create/rename/delete)**
- **Reading file/directory date information**
- **Additional file management operations (rename/move)**
- **Directory management (create/rename/move)**
- **File/directory comment modification**
- **Listing of SE from BDII with the ability to ignore some SEs set in a properties file**

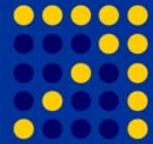


Installation and Configuration /1

- **0.) First of all you have to install the following APIs:**
 - **GFAL C/C++ API**
 - **lcg_util C/C++ API**
 - **LFC C/C++ API**
 - **gLite FTS Java API**

These APIs come with the last release of gLite middleware and should be already installed and configured by your sys-admin.

- **1.) Then you have to download the SEE-GRID File Management API (see references)**



Installation and Configuration /2

- 2.) Set the **CLASSPATH** variable to contain the path of the SEE-GRID File Management API.
 - Set the value of the VO variable with the name of your Virtual Organization.
 - Set the value of **bdii.host.name** and **bdii.host.port**.
 - Optionally, set the value of property **SE.ignore.list** with the list of SEs to be ignored when obtaining availables SEs from the BDII.



```
$ cat repmgr.properties
e.o, LFest Java API properties. This file's location needs
  to be in classpath.
#
# Name of Virtual Organisation
VO      =      gilda

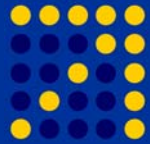
# List of available Storage Elements separated by space
  character

SEList  =      trigrid-ce01.unime.it
  gildase01.roma3.infn.it iceage-se-01.ct.infn.it
  gildase.oact.inaf.it  aliserv6.ct.infn.it
  testbed005.cnaf.infn.it egee016.cnaf.infn.it
  grid005.iucc.ac.il  grid-se.bio.dist.unige.it
  gilda02.lcca.usp.br

bdii.host.name = grid004.ct.infn.it
bdii.host.port = 2170
```




- 3.) Set the **LCG_GFAL_INFOSYS** and **LFC_HOST** variables as follow:
 - `export LFC_HOST=lfc-gilda.ct.infn.it`
 - `export LCG_GFAL_INFOSYS=grid004.ct.infn.it`



- **LFCDataStorage** class is an implementation of **DataStorageInterface** which provides information and access to data storage resources of a grid.

Method Summary	
java.lang.String	getCatalogType() Returns type of used grid file catalogue.
Item	getItem(java.lang.String pathName) Returns the Item of the corresponding type for a given pathname.
DirectoryItem	getRoot() Returns root directory of grid file catalogue.
SEList	getSEList() Returns list of available storage elements.
java.lang.String	getSeparator() Returns default pathname-separator character for used grid file catalogue.
java.lang.String	getVO() Returns name of Virtual Organisation.



- LFCDirectoryItem class provides methods to retrieve directory information and data management.

Method Summary	
boolean	canExecute() Test for execute permission.
boolean	canRead() Test for read permission.
boolean	canWrite() Test for write permission.
boolean	copyAndRegister (java.lang.String sourceFilePath, java.lang.String destinationSE) Copies and registers file in grid catalogue directory.
boolean	copyAndRegister (java.lang.String sourceFilePath, java.lang.String destinationFileName, java.lang.String destinationSE) Copies and registers file in grid catalogue directory.
boolean	createNewAlias (java.lang.String newAliasPathname) Creates the Item's alias with a given pathname.
boolean	exists() Test if the item denoted by pathname exists.
java.lang.String[]	getAliases() Returns the list of Item's aliases.
java.lang.String	getComment() Returns associated comment.
int	getFileMode() Returns the filemode value describing item's type and permissions.
int	getGID() Returns the Group ID (GID) of the group owning the Item.
java.lang.String	getGroup() Returns the name of the group owning the Item.

Item	getParent() Returns parent item.
java.lang.String	getParentPathName()
long	getSize() Returns size in bytes.
int	getUID() Returns the User ID (UID) of the user owning the Item.
java.lang.String	getUser() Returns the name of the user owning the Item.
boolean	mkdir (java.lang.String name) Creates subdirectory with the given name.
boolean	mkdir (java.lang.String name, LFCFileMode lfcFileMode) Creates subdirectory with the given name and permissions.
protected void	populateChildren() Fetches the items contained by the directory.
void	refresh() Refreshes the cached information about the directory.
boolean	renameTo (java.lang.String newPathName) Renames/moves the item to a given pathname.
void	setComment (java.lang.String comment) Assigns a new comment to the item.



- LFCFileItem class provides methods to retrieve file information and data management.

Method Summary	
boolean	canExecute() Test for execute permission.
boolean	canRead() Test for read permission.
boolean	canWrite() Test for write permission.
boolean	createNewAlias(java.lang.String newAliasPathname) Creates the Item's alias with a given pathname.
boolean	delete() Deletes file.
boolean	deleteReplicaFromSE(java.lang.String se) Deletes replica of a file from specified Storage element
boolean	deleteReplicaFromSurl(java.lang.String surl) Deletes replica specified by surl.
boolean	download(java.lang.String destinationFile) Downloads file to local filesystem.
boolean	download(java.lang.String surl, java.lang.String destinationFile) Downloads file to local filesystem.
boolean	exists() Test if the item denoted by pathname exists.
java.lang.String[]	getAliases() Returns the list of Item's aliases.
java.lang.String	getComment() Returns comment associated with file.

int	getFileMode() Returns the filemode value describing item's type and permissions.
int	getGID() Returns the Group ID (GID) of the group owning the Item.
java.lang.String	getGroup() Returns the name of the group owning the Item.
java.lang.String	getGUID() Returns guid of a file.
Item	getParent() Returns parent item.
java.lang.String	getParentPathName()
java.lang.String[]	getReplicas() Returns list of file's replicas.
long	getSize() Returns size in bytes.
int	getUID() Returns the User ID (UID) of the user owning the Item.
java.lang.String	getUser() Returns the name of the user owning the Item.
void	refresh() Refreshes the cached information about the file.
boolean	renameTo(java.lang.String newPathName) Renames/moves the item to a given pathname.
boolean	replicate(java.lang.String se) Replicates file.



Listing directory content of a LFC

BioinfoGRID

```
import yu.ac.bg.rcub.grid.dataManagement.*;

/**
 * List files for a given directory. SEE-GRID File Management API test
 * @author Dragan Okiljevia
 * @version 1.2 06/10/15
 */

public class LfcLs {
    public static void main(String [] args){
        if(args.length!=1){
            System.out.println("Usage: java LfcLs grid_directory");
            System.exit(-1);
        }
        DataSourceInterface dsi = new LFCDataSource();
        DirectoryItem di = new LFCDirectoryItem(args[0],null,dsi);
        ItemIterator iter = di.itemIterator();

        while(iter.moreChildren()){
            Item nextItem = iter.next();
            System.out.println((new LFCFileMode(nextItem.getFileMode()))+"\t"
                +nextItem.getUID()+"\t"+nextItem.getGID()+"\t"+nextItem.getSize()+"\t"
                +nextItem.getName());
        }
    }
}
```

provides information and access to data storage resources of a grid Infrastructure

provides methods for retrieving directory information and data management



Retrieve list of SE(s) from the BDII

BioinfoGRID

```
import yu.ac.bg.rcub.grid.dataManagement.*;

/**
 * Returns list of SE from BDII.
 * @author Dragan Okiljević
 * @version 1.2 06/10/15
 * @since 1.2
 */

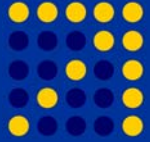
public class listSE {

    public static void main(String [] args) {

        if(args.length!=0){
            System.out.println("Returns list of SE from BDII.");
            System.out.println("Usage: java listSE");
            System.exit(-1);
        }

        DataStorageInterface dsi = new LFCDataStorage();

        SEList SEs = dsi.getSEList();
        for(int i=0;i<SEs.size();i++) {
            System.out.println(SEs.get(i));
        }
        System.out.println("Returned "+SEs.size()+" entries");
    }
}
```



Copy and register file(s) on the SE /1

BioinfoGRID

```
import yu.ac.bg.rcub.grid.dataManagement.*;

/**
 * Copies and registers file to grid.
 * @author Dragan Okiljević
 * @version 1.2 06/10/15
 * @since 1.1
 */

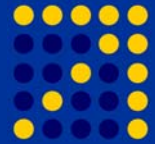
public class copyAndRegister {

    public static void main(String args []){

        int numArgs = args.length;
        if(numArgs!=4){
            System.out.println("Copies and registers file to grid");
            System.out.print("Usage: copyAndRegister sourceFilePath ");
            System.out.print("gridDestinationDir gridDestFileName SENAME");
            System.out.println();
            System.exit(-1);
        }

        DataStorageInterface dsi = new LFCDDataStorage();
        DirectoryItem di = new LFCDirectoryItem("",args[1],null,dsi);

        boolean success;
        success = di.copyAndRegister(args[0],args[2],args[3]);
    }
}
```



Copy and register file(s) on the SE /2

BioinfoGRID

```
if (success) {  
    System.out.print("File "+args[0]+" copied and registd as: ");  
    System.out.print(args[1]+"/"+args[2]+".");  
    System.out.println();  
}  
else{  
    System.err.println("Unable to copy and register file.");  
    System.exit(-1);  
}  
}
```




SEE-GRID File Management Java API Documentation

<http://grid02.rcub.bg.ac.yu/LFCJavaAPI/files/docs/javadoc/version1.2/index.html>



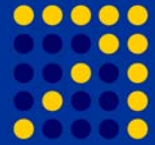
Source code (version 1.2)

<http://grid02.rcub.bg.ac.yu/LFCJavaAPI/files/downloads/SEE-GRIDFileManagementAPIv1.2.zip>



Source code (version 1.1)

<http://grid02.rcub.bg.ac.yu/LFCJavaAPI/files/downloads/SEE-GRIDFileManagementAPIv1.1.zip>



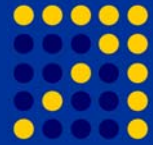
BioinfoGRID

Download



SEE-GRID File Management Java API ver. 1.1

<https://grid.ct.infn.it/twiki/bin/view/GILDA/LFCJAVAAPI>

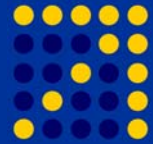


- Overview of the WMPProxy Java API
- Overview of the SEE-GRID File Management Java API
- Overview of the GFAL Java API



G F A L: Grid File Access Library J a v a A P I

**GFAL Overview
GFAL Available APIs
GFAL Java API Overview
GFAL Java API Practical**



GFAL (Grid File Access Library)

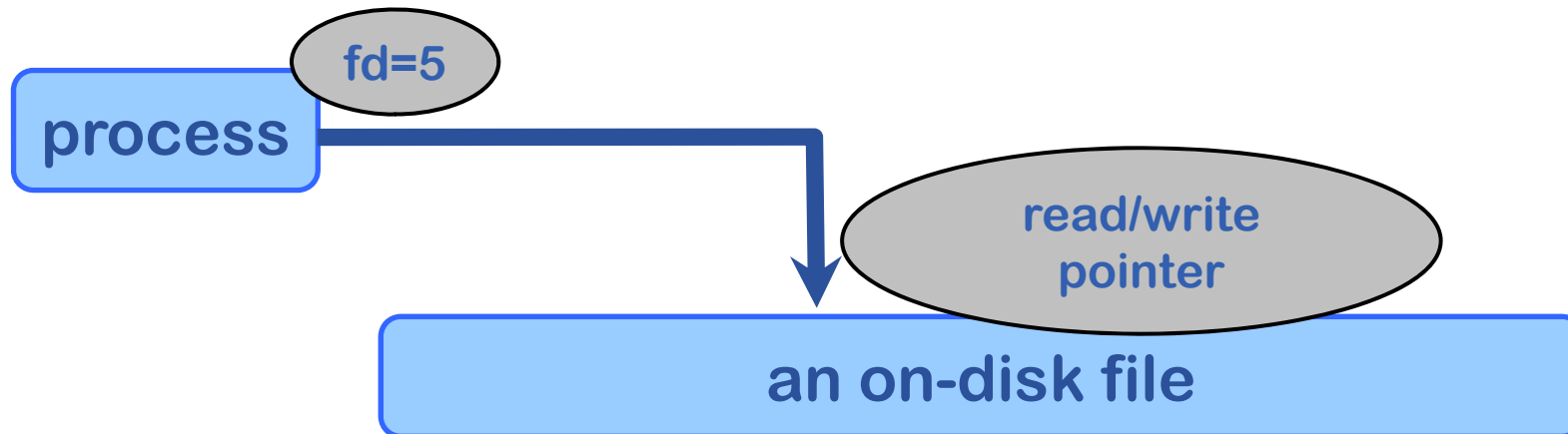
- **GFAL is a POSIX-like file access layer**
- **Allows for access to files on remote Storage Elements**
- **Implemented as a C library**
 - Java bindings are also available, implemented by GILDA
- **The destination SE must support secure rfi**
 - Not available for Classic SEs
- **Many details in the man page (man gfal)**



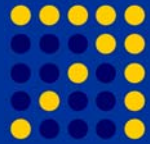
Why use GFAL?

- Hides away all complexity of Grid data management behind the “open file” abstraction
 - Supports many transport protocols
 - file, dcap,gsidcap, kdcap, rfio TURL types
- Can be used to opens file at any level (LFN, GUID, SURL, or TURL)
 - GFAL will take care of the needed steps to acquire a Transport URL
 - It will automatically select the most appropriate transfer protocol, depending on the kind of SE where the file is located on.

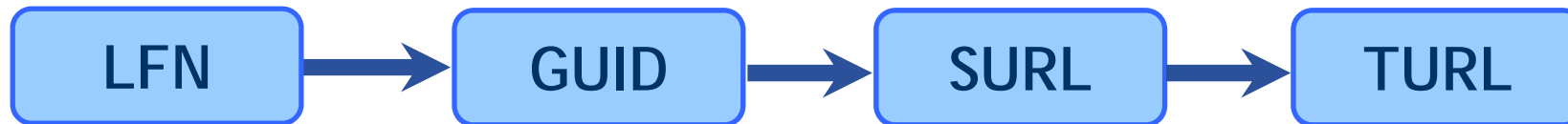
In the case LFNs or GUIDs are used, GFAL needs to contact the LFC Catalog to obtain the corresponding TURL. For GFAL to be able to discover the LFC endpoints and to find out information about the Storage Elements, the user must set the environment variables **LCG_GFAL_VO** and **LCG_GFAL_INFOSYS** to the VO name and the BDII hostname and port.



- a file is a collection of bytes
- an open file is represented by a **file descriptor** (an integer)
- **open()** is used to get a file descriptor
- the process may **read()** from or **write()** to the file
- **lseek()** moves the file read/write pointer
- ... and **close()** to finally release the open file structure



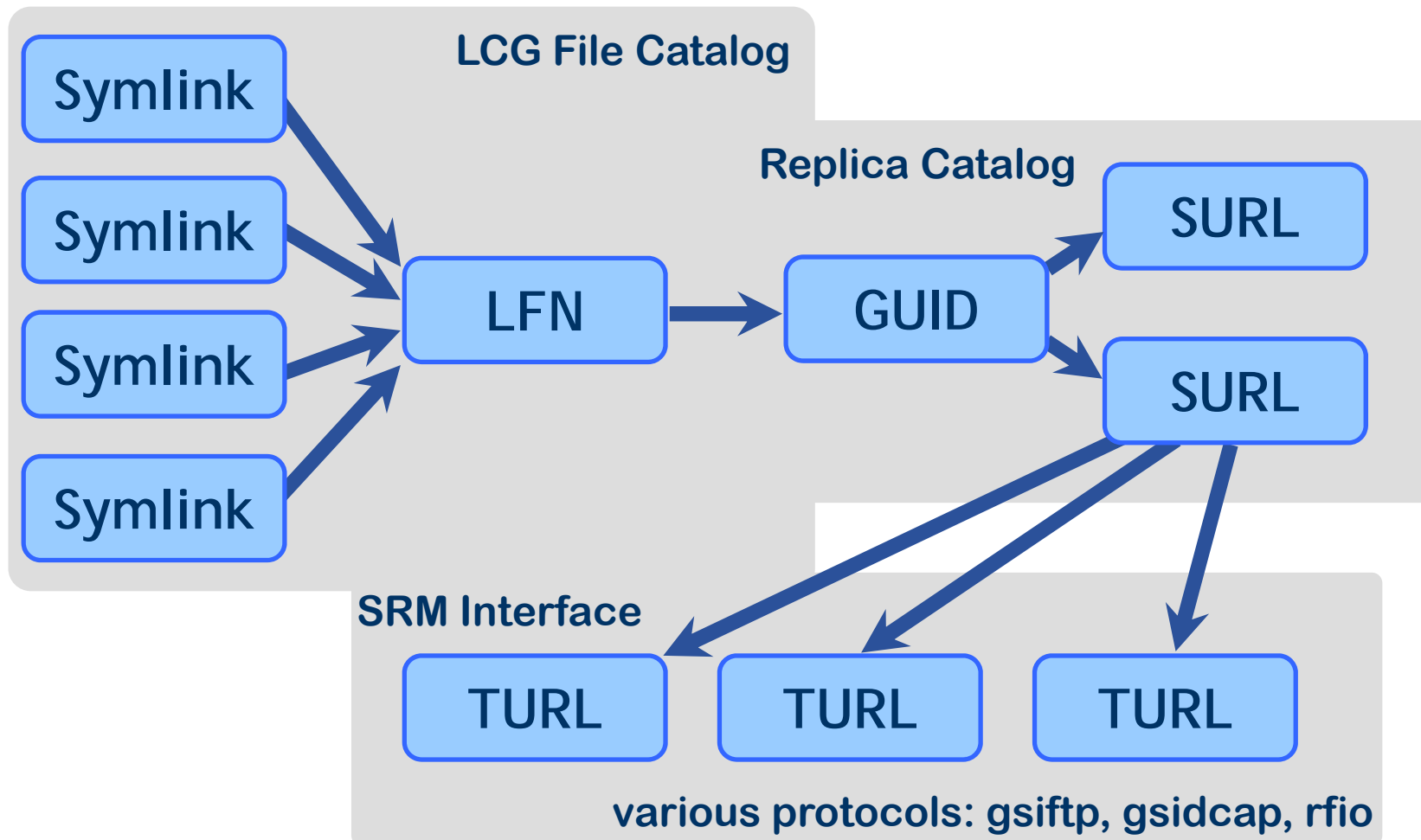
Grid file referencing schemes

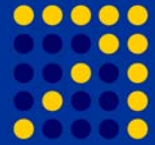


- **Logical File Name**
 - `lfn:/grid/gilda/vkoukis-gfal-test2`
- **Grid Unique Identifier (GUID)**
 - `guid:4f74b453-5aaf-4af5-af2d-b438d081bb63`
- **Storage URL (for a specific replica, on a specific Storage Element)**
 - `srm://se01.athena.hellasgrid.gr/pnfs/athena.hellasgrid.gr/data/gilda/generated/2007-04-15/file775276c2-7bbf-4f3d-92cf-5d5cdbaba254`
- **Transport URL (for a specific replica, on an SE, with a specific protocol)**
 - `gsidcap://se01.athena.hellasgrid.gr:22128//pnfs/athena.hellasgrid.gr/data/gilda/generated/2007-04-15/file775276c2-7bbf-4f3d-92cf-5d5cdbaba254`



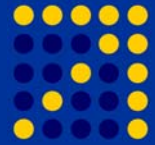
Grid file management





C compilation using GFAL

- **GFAL implemented as a shared library**
 - /opt/lcg/lib/libgfal.so
 - /opt/lcg/lib/libgfal_pthr.so
- **All interface declarations in a single header file**
 - /opt/lcg/include/gfal_api.h
- **Needs a few extra compiler flags**
 - gcc -I/opt/lcg/include -L/opt/lcg/lib -o executable source.c -lgfal



GFAL: POSIX-like file access

- `open()`
- `read()`
- `write()`
- `lseek()`
- `close()`
- `gfal_open()`
- `gfal_read()`
- `gfal_write()`
- `gfal_lseek()`
- `gfal_close()`

Many more defined in `/opt/lcg/include/gfal_api.h!`



Open file: `gfal_open()`

- **Syntax**

```
int gfal_open(const char *pathname, int flags, mode_t mode)
```

- Returns an open file descriptor

- “**mode**” argument is mandatory, even when requesting `O_RDONLY` access

- “**pathname**” can be anything, from an LFN to an TURL

- **Usage example:**

```
if ((fd = gfal_open("lfn:/grid/eumed/myfile1, O_RDONLY, 0))  
< 0) {  
    perror("gfal_open");  
    exit(1);  
}
```



- **Syntax**

```
int gfal_close (int fd)
```

- **Closes an open file descriptor**
- **Releases resources reserved on remote SEs**
- **Usage example:**

```
if (gfal_close(fd) < 0) {  
    perror("gfal_close");  
    exit(1);  
}
```



Fetch data: `gfal_read()`

- **Syntax**

```
ssize_t gfal_read (int fd, void *buf, size_t len)
```

- Reads data from the file pointed to by “**fd**”
- At most “**len**” bytes are retrieved from the current file read/write location and stored in “**buf**”
- The number of bytes actually retrieved is returned, -1 otherwise
- Usage example:

```
ssize_t ret; char buf[1024];  
if ((ret = gfal_read(fd, buf, 1024)) < 0) {  
    perror("gfal_read");  
    exit(1);  
}
```



Store data: `gfal_write()`

- **Syntax**

```
ssize_t gfal_write(int fd, void *buf, size_t len)
```

- Writes data to the file pointed to by “**fd**”
- At most “**len**” bytes are retrieved from the memory location pointed to by “**buf**” and stored in the current file read/write location
- The number of bytes actually stored is returned, -1 otherwise
- Usage example:

```
ssize_t ret; char buf[1024];  
if ((ret = gfal_write(fd, buf, 1024)) < 0) {  
    perror("gfal_write");  
    exit(1);  
}
```



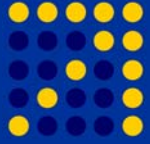
Change file pos: `gfal_lseek()`

- **Syntax**

```
off_t lseek(int fildes, off_t offset, int whence)
```

- Changes the position on the file read/write pointer
- **whence** is one of { `SEEK_SET`, `SEEK_CUR`, `SEEK_END` }
- If the call is successful, the new file offset is returned, -1 otherwise
- Useful for reading only part of a very large file
- Usage example:

```
/* Skip the first 2MB of a file */  
if ((off = gfal_lseek(fd, 2*1048576, SEEK_SET)) < 0) {  
    perror("gfal_lseek");  
    exit(1);  
}
```

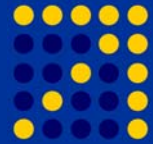



- Java API (C API Wrapper)
- It provides three main objects that abstract the interactions with files on remote SEs
 - **GFalFile** : for reading and writing data from/to files
 - **GFalDirectory** : for manipulation of directories (create, delete, list entries)
 - **GFalUtilities** : for file management operations (rename, stat, lstat, delete)



– **GFalFile** : for reading and writing data from/to files

Method Summary	
void	closeFile () Closes the file opened by openFile.
void	createFile (java.lang.String name, int mode, boolean isSurl, boolean isLargeFile) Creates a file
java.lang.String	getLFN () Returns the LFN of the file associated with this object .
java.lang.String	getSurl () Returns the SURL of the file associated with this object .
void	lfcRegisterFile (java.lang.String logicalFileName) Registers the file associated with this object in the lfc catalog.
long	lseekFile (long offset, int seekMode, boolean isLargeFile) Positions/repositions to offset the file associated with this object.
void	openFile (java.lang.String fileName, int flags, int mode, boolean isLargeFile) Opens a file according to the value of flags.
byte[]	readFile (int size) Reads size bytes from the file
int	writeFile (byte[] buffer) Writes buffer data in the file



GFalDirectory Java Class

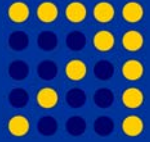
- **GFalDirectory** : for manipulation of directories (create, delete, list entries)

Method Summary	
void	closeDir () Close the directory associated with this object.
static void	makeDir (java.lang.String dirName, int mode) Creates a new directory with permission bits taken from mode.
void	openDir (java.lang.String dirName) Open a directory.
java.lang.String[]	readDir () Read the directory associated with this object.
static void	rmDir (java.lang.String dirName) Removes a directory if it is empty.



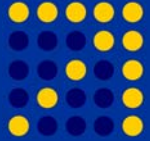
- **GFalUtilities** : for file management operations (rename, stat, lstat, delete)

Method Summary	
int	accessFile (java.lang.String fileName, int mode) Checks the existence or the accessibility of the file/directory path according to the bit pattern in amode using the real user ID.
void	chmodFile (java.lang.String fileName, int mode) Change access mode of a file/directory .
void	deleteFile (java.lang.String fileName) Remove a file entry.
long[]	lstatFile (java.lang.String fileName, boolean isLargeFile) Gets information about a file or directory.
void	renameFile (java.lang.String oldName, java.lang.String newName) Rename a file or a directory.
long[]	statFile (java.lang.String fileName, boolean isLargeFile) Gets information about a file or directory.



Pre-requisites for using GFAL

- Certain environment variables must be set for GFAL to know the context it operates in
 - **LCG_GFAL_INFOSYS**, points to the BDI
 - **LCG_GFAL_VO**, the name of the user's VO
 - **LCG_CATALOG_TYPE=lf**, LFC will be used for locating replicas
 - **LFC_HOST**, the name of the catalog to be used
 - **LCG_RFIO_TYPE=dpm**, otherwise nothing works 😊
- Needs a valid VOMS proxy for authentication
 - Try voms-proxy-init, voms-proxy-info



Read a file directly from the SE /1

BioinfoGRID

```
/*
 * GFALDownloadFileTest.java
 * Created on 24 luglio 2006, 18.43
 * Salvatore Scifo, INFN sez. CT, salvatore.scifo@ct.infn.it
 */

import it.infn.catania.gfal.GFalFile;
import it.infn.catania.gfal.GFalUtilities;
import java.io.FileOutputStream;
import java.io.BufferedOutputStream;

public class GFALDownloadFileTest {

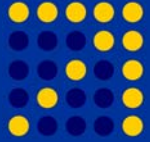
    public static void main(String[] args) throws Exception {

        if(args == null || args.length < 2)
        {
            System.out.println("\nWrong Arguments Passed!");
            System.out.println("\nUsage: GFALDownloadFileTest LFN LocalFileName");
            System.out.println("\nEs:GFALDownloadFileTest lfn:/grid/gilda/file.dat
/home/user/file.dat");
            System.exit(-1);
        }

        String LFN = args[0];
        System.out.println("\nLFN : " + LFN);

        String localFileName = args[1];
        System.out.println("\nLocal File Name : " + localFileName);

        /* Create a new instance of the Class */
        GFalUtilities gfalUtils = new GFalUtilities();
```



Read a file directly from the SE /2

BioinfoGRID

```
/* Gets information about a file or directory */
long[] stat = gfalUtils.statFile(LFN, false);

/* This routine returns an array of long containing the
   following information: mode, nlink, uid, gid and size. */
int fileSize = (int) stat[4];

GFalFile gfalFile = new GFalFile();

/* Opens a file according to the value of flags
   flags : READONLY WRITEONLY CREAT LARGEFILE
   mode : access permission of the new file (e.g. 644)
   isLargeFile : set this flag=true if you want to open a large file
*/
gfalFile.openFile(LFN, GFalFile.READONLY, 644, false);

/* Reads size bytes from the file */
byte[] buffer = gfalFile.readFile(fileSize);

/* Closes the file opened by openFile */
gfalFile.closeFile();

FileOutputStream fo = new FileOutputStream(localFileName);
BufferedOutputStream bfo = new BufferedOutputStream(fo);

bfo.write(buffer); bfo.flush(); bfo.close(); fo.close();
bfo = null; fo = null; buffer = null; gfaUtils = null;
}
}
```



Write and register a file directly to an SE /1

BioinfoGRID

```
import it.infn.catania.gfal.GFalFile;

/*
 * GFALUploadFileTest.java
 * Created on 24 luglio 2006, 18.43
 * Salvatore Scifo, INFN sez. CT, salvatore.scifo@ct.infn.it
 */

public class GFALUploadFileTest {

    public static void main(String[] args) throws Exception {

        if(args == null || args.length < 4)
        {
            System.out.println("\nWrong Arguments Passed!");
            System.out.println("\nUsage: GFALUploadFileTest LocalFileSystemName seURL LFN aMode");
            System.out.println("\nEs: GFALUploadFileTest /home/user/file.dat aliserv6.ct.infn.it
lfn:/grid/gilda/file.dat 644");
            System.exit(-1);
        }

        String fileName = args[0];
        System.out.println("\nLocalFileName : " + fileName);
        String seURL = args[1];
        System.out.println("\nDestination SE URL : " + seURL);
        String lfn = args[2];
        System.out.println("\nLFN : " + lfn);
        String mod = args[3];
        System.out.println("\nInsert File Permission (es: 644) : " + mod);
    }
}
```




Write and register a file directly to an SE /2

BioinfoGRID

```
FileToByteArray fileTBA;

/* Create a new instance of the class */
GFalFile gfalFile = new GFalFile();

String SURL = null;
try {
    fileTBA = new FileToByteArray(fileName);

    /* Create a new file with the following parameter(s):
    name - the name of the storage element where the file will be
        created or the name of the new file (a logical file name, a guid,
        a SURL or a TURL).
    mode - access permission of the new file (e.g. 644)
    isSurl - false if the first parameter is a name of a storage element
        (SURL automatically generated),
        true if the first parameter is a logical file name, a guid, a SURL or a TURL
    isLargeFile - set this flag=true if you want to create a large file
    */
    gfalFile.createFile(seURL, Integer.parseInt(mod.trim()), false, false);

    /* Writes buffer data in the file */
    int ret = gfalFile.writeFile(fileTBA.toByteArray());

    if(ret == -1) {
        SURL = "No SURL has been provided!";
        throw new Exception("Error has been detected during file writing onto SE : " + seURL);
    }
}
```



Write and register a file directly to an SE /3

BioinfoGRID

```
else {  
    /* Returns the SURL of the file associated with this object */  
    SURL = gfalFile.getSurl();  
  
    /* Closes the file opened by openFile */  
    gfalFile.closeFile();  
  
    /* Registers the file associated with this object in the lfc catalog */  
    gfalFile.lfcRegisterFile(lfn);  
}  
  
} catch(Exception e) {  
    System.out.println("\n" + e.getMessage());  
    if(SURL == null) SURL = "No SURL has been provided!";  
}  
  
System.out.println("\nFollowing SURL has been created : " + SURL);  
fileTBA = null;  
}  
}
```



Examples in gLite3 User Guide (Appendix F)

- <https://edms.cern.ch/file/722398//gLite-3-UserGuide.pdf>



GFAL C API Description:

- http://grid-deployment.web.cern.ch/grid-deployment/documentation/LFC_DPM/gfal/html/



GFAL JAVA API

- <https://grid.ct.infn.it/twiki/bin/view/GILDA/APIGFAL>



On-line JavaDoc of Java API:

- <https://grid.ct.infn.it/twiki/GFAL/>

