



Enabling Grids for E-science

# WMPProxy Java API & SEE-GRID File Management Java API

Giuseppe La Rocca  
INFN – Catania  
giuseppe.larocca@ct.infn.it

Training the Trainers – in conjunction with  
ISGC 19-23 March 2007 - TAIPEI

[www.eu-egee.org](http://www.eu-egee.org)



- **Overview of the WMPProxy Java API**
  - Software Requirements
  - Class `org.glite.wms.wmproxy.WMPProxyAPI`
  - An use case
  
- **Overview of the SEE-GRID File Management Java API**
  - Installation & Configuration
  - The Classes : `LFCDataStorage`, `LFCDirectoryItem` and `ItemIterator`
  - Two different examples

# WMPProxy Java API

## version 3.1.0

- **The WMPProxy is the service responsible to provide access to the WMS functionality through a Web Service Interface**
- **The gLite WMproxy 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>

- **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 download 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**

- In order to allow grid user to invoke, within Java code, the **UrlCopy** class to copy files from the WMPProxy 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](http://dev.globus.org/wiki/CoG_JGlobus_1.4)



- 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;

- 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>
```

- 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  
wmpoxy_url,user_proxy,CACertsPath);
```

```
String proxy = client.grstGetProxyReq (delegationId);  
client.grstPutProxy(delegationId, proxy);
```

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

- 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 WMPProxy node;
- call the jobStart service to trigger the submission:

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

```

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**





```
[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> <wmproxy_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

<wmproxy_server>     ... the entry point of the WMPProxy Server to contact
                       (e.g. : https://glite-rb3.ct.infn.it:7443/glite_wms_wmproxy_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-rb3.ct.infn.it:7443/glite_wms_wmproxy_server with the proxy..
/tmp/x509up_u512
```

**Your job has been successfully submitted.**

```
jobID = [ https://glite-rb3.ct.infn.it:9000/XAoY7FZgLLJjgCp4U9grsBw ]
```



```

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-rb3 root]# cd /var/glite/SandboxDir/XA/
```

```
[root@glite-rb3 root]# ll https_3a_2f_2fglite-  
rb3.ct.infn.it_3a9000_2fXAoY7FZgLGc4U9grsBw/input/
```

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



With the job finish its execution you can retrieve the output files 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-rb3.ct.infn.it:7443/glite_wms_wmproxy_server]
proxyFile    = [/tmp/x509up_u512]
JobID        = [https://glite-rb3.ct.infn.it:9000/XAoY7FZgLJjgCp4U9grsBw]
dirPath      = [/home/larocca/API/]
CA certs     = [/etc/grid-security/certificates/]
+-----+
```

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

```
Start downloading output file(s)..
file n. 1
-----
name = [gsiftp://glite-rb3.ct.infn.it:2811/var/glite/SandboxDir/XA/https_3a_2f_2fglite-
rb3.ct.infn.it_3a9000_2fXAoY7FZgLJjgCp4U9grsBw/output/hostname.err]
size = [0]

file n. 2
-----
name = [gsiftp://glite-rb3.ct.infn.it:2811/var/glite/SandboxDir/XA/https_3a_2f_2fglite-
rb3.ct.infn.it_3a9000_2fXAoY7FZgLJjgCp4U9grsBw/output/hostname.out]
size = [28]
```



- **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>

# **SEE-GRID File Management Java API**

## **version 1.1**

- **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.**

- **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.**

- Then you have to:
  - download the SEE-GRID File Management API
  - 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. only for v.1.2
  - Optionally, set the value of property SE.ignore.list with the list of SEs to be ignored when obtaining available 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 grid038.ct.infn.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
```



- Check if the following environment variable(s) have been correctly exported:

```
export LCG_GFAL_VO=gilda
```

```
export LCG_GFAL_INFOSYS=grid004.ct.infn.it:2170
```

- **LFCDataStorage** class is an implementation of **DataStorageInterface** which provides information and access to data storage resources of a grid.
  - Information includes list of available Storage Element, Virtual Organization and the grid file catalogue.
- **LFCDirectoryItem** class provides methods to retrieve information about file catalogue directory.
- **ItemIterator** class allows iteration of grid catalogue items.

```

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

public class ExerLFC1{
    public static void main(String [] args){

        if(args.length == 0){
            System.out.println("Usage: java ExerLFC1 <grid_path> where e.g.: grid_path=/grid/gilda/ \n");
            System.exit(-1);
        }

        long startTime = System.currentTimeMillis();
        DataStorageInterface dsi = new LFCDataStorage();
        DirectoryItem di = new LFCDirectoryItem(args[0],null,dsi);
        long startMain = System.currentTimeMillis();
        ItemIterator iter = di.itemIterator();
        java.lang.String type = "";

        while(iter.moreChildren())
        {
            Item nextItem = iter.next();
            if (nextItem.isDir()) type = "Directory: ";
            else type = "File: ";

            System.out.println( type+ nextItem.getName()+"\t"+
                (new LFCFileMode(nextItem.getFileMode()))+"\t"+
                nextItem.getUID()+"\t"+
                nextItem.getGID()+"\t"+
                nextItem.getSize()+" bytes");
        }
        long endTime = System.currentTimeMillis();
        System.out.println("\nExecution time: "+(endTime-startTime)+" ms\n");
    }
}
    
```



```

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

public class ExerLFC2{
    public static void main(String args []){
        int numArgs = args.length;

        if(numArgs != 4){
            System.out.println("\nUsage: ExerLFC2 sourceFilePath gridDestinationDir gridDestFileName SENAME");
            System.out.println("es.: ExerLFC2 /home/vardizzo/myfile.txt /grid/gilda/valeria mylastfile.txt aliserv6.ct.infn.it");
            System.exit(-1);
        }

        DataStorageInterface dsi = new LFCDataStorage();
        DirectoryItem di = new LFCDirectoryItem("",args[1],null,dsi);
        boolean success = di.copyAndRegister(args[0],args[2],args[3]);

        if (!success) {
            System.err.println("Unable to copy and register file.");
            System.exit(-1);
        }
        else{
            FileItem fi = new LFCFileItem(args[1],args[1]+"/"+args[2],di,dsi);
            System.out.println(args[2]+
                "\t"+(new LFCFileMode(fi.getFileMode()))+
                "\t"+fi.getUID()+
                "\t"+fi.getGID()+
                "\t"+fi.getSize()+
                " bytes");
        }
    }
}
    
```



- **SEE-GRID File Management Java API Documentation**

<http://grid02.rcub.bg.ac.yu/LFCJavaAPI/files/docs/java/doc/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>

## How access the GILDA User Interface

**Login** : taipeiXX@glite-tutor.ct.infn.it  
where XX=40,..60

**Passwd** : GridTAIXX XX=40,..,60

**PEM PASSPHRASE** : TAIPEI

## WMPProxy Java API

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

## SEE-GRID File Management Java API

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

