
csgf Documentation

Release latest

June 08, 2016

1	INTRODUCTION	3
2	INSTALLATION & CONFIGURATION GUIDE	5
2.1	Configuring MySQL Database for liferay	5
2.2	Installing Liferay 6.1.1 on Glassfish 3.1.x	7
2.3	Enabling LDAP Authentication	17
2.4	Configuring a Service Provider for Shibboleth Authentication	28
2.5	Configuring the MySQL Database for the Grid & Cloud Engine	30
2.6	Configuring the Grid & Cloud Engine on Liferay	32
2.7	Installing the eTokenServer	38
3	APPLICATION REGISTRY	41
3.1	About	41
3.2	Usage	41
3.3	Support	44
4	ETOKEN	45
4.1	About	45
4.2	Installation	46
4.3	Usage	46
4.4	Contributor(s)	46
5	FEDERATED LOGIN	71
5.1	About	71
5.2	Installation	71
5.3	Usage	76
5.4	Contributors	76
6	GLIBRARY	79
6.1	GLIBRARY 1.0	79
6.2	gLibrary 2.0	90
7	GRID & CLOUD ENGINE	103
7.1	About	103
7.2	Installation	104
7.3	Usage	111
7.4	Contributors	111
8	JSAGA ADAPTOR FOR ROCCI-BASED CLOUDS	113

8.1	About	113
8.2	Installation	114
8.3	Usage	116
8.4	Contributor(s)	122
9	MYJOBS	123
9.1	About	123
9.2	Installation	123
9.3	Usage	125
9.4	Contributors	125
10	OPENID CONNECT FOR LIFERAY	127
10.1	About	127
10.2	Installation	127
10.3	Usage	128
10.4	Contributors	128
11	SG-MON - INSTALLATION AND CONFIGURATION	129
11.1	About	129
11.2	Introduction to SG-Mon	129
11.3	Requirements	129
11.4	Installation	129
11.5	Configuration	130
12	ABINIT WITH DATA MANAGEMENT	135
12.1	About	135
12.2	Installation	135
12.3	Usage	137
12.4	References	139
12.5	Contributor(s)	139
13	ALEPH ANALYSIS	141
13.1	About	141
13.2	Installation	142
13.3	Usage	144
13.4	Contributor(s)	147
14	ALICE ANALYSIS	149
14.1	About	149
14.2	Installation	150
14.3	Usage	150
14.4	Contributor(s)	152
15	ASTRA	153
15.1	About	153
15.2	Installation	153
15.3	Usage	155
15.4	References	156
15.5	Contributors	157
16	CLOUDAPPS	159
16.1	About	159
16.2	Installation	159
16.3	Usage	162
16.4	Contributor(s)	164

17	CLUSTALW	165
17.1	About	165
17.2	Installation	165
17.3	Usage	167
17.4	Contributor(s)	169
18	CORSIKA-LAGO	171
18.1	About	171
18.2	Installation	171
18.3	Usage	172
18.4	References	175
18.5	Contributors	175
19	GLIBRARY-REPO-BROWSER-PORTLET	177
19.1	Installation	177
19.2	Usage	179
19.3	Contributor(s)	179
20	GROMACS	181
20.1	About	181
20.2	Installation	181
20.3	Usage	183
20.4	References	185
20.5	Contributor(s)	185
21	INFECTION MODEL PORTLET	187
21.1	About	187
21.2	Installation	187
21.3	Usage	190
21.4	Contributor(s)	191
22	INFECTION MODEL PARALLEL PORTLET	193
22.1	About	193
22.2	Installation	193
22.3	Usage	197
22.4	Contributor(s)	197
23	INTEROPERABILITY DEMO	199
23.1	About	199
23.2	Contributors	199
24	IORT	201
24.1	About	201
24.2	Installation	201
24.3	Usage	203
24.4	References	205
24.5	Contributor(s)	205
25	KNOWLEDGE BASE	207
25.1	About	207
25.2	Installation	209
25.3	Support	209
26	MPI	211
26.1	About	211

26.2	Installation	211
26.3	Usage	213
26.4	Contributor(s)	213
27	MULTI-INFRASTRUCTURE PARALLEL JOB	215
27.1	About	215
27.2	Installation	215
27.3	Usage	218
27.4	Contributors	219
28	MULTI-INFRASTRUCTURE “HELLO WORLD!”	221
28.1	About	221
28.2	Installation	221
28.3	Usage	223
28.4	Contributor(s)	223
29	NUCLEMD	225
29.1	About	225
29.2	Installation	225
29.3	Usage	227
29.4	Contributor(s)	229
30	OCTAVE	231
30.1	About	231
30.2	Installation	231
30.3	Usage	233
30.4	Contributor(s)	233
31	PARALLEL SEMANTIC SEARCH	235
31.1	About	235
31.2	Installation	235
31.3	Usage	236
31.4	References	239
31.5	Contributors	239
32	PI CALCULATION	241
32.1	About	241
32.2	Installation	241
32.3	Usage	243
32.4	Contributor(s)	244
33	R - STATISTICAL COMPUTING	245
33.1	About	245
33.2	Installation	245
33.3	Usage	247
33.4	Contributor(s)	248
34	SEMANTIC SEARCH	249
34.1	About	249
34.2	Installation	249
34.3	Usage	249
34.4	References	251
34.5	Contributors	252
35	SONIFICATION	253

35.1	About	253
35.2	Installation	253
35.3	Usage	255
35.4	References	257
35.5	Contributor(s)	257
36	TEMPLATE PORTLET	259
36.1	About	259
36.2	Installation	259
36.3	Usage	263
36.4	Contributor(s)	263
37	TEMPLATE SPECIAL JOB PORTLET	265
37.1	About	265
37.2	Installation	265
37.3	Usage	269
37.4	Contributor(s)	270
38	TRODAN	271
38.1	About	271
38.2	Installation	272
38.3	Usage	274
38.4	Contributor(s)	277
39	VISUALIZE INFECTION PORTLET	279
39.1	About	279
39.2	Installation	279
39.3	Usage	280
39.4	Contributor(s)	280
40	WRF	281
40.1	About	281
40.2	Installation	281
40.3	Usage	285
40.4	Contributor(s)	287
41	AGINFRA SG MOBILE	289
41.1	About	289
41.2	Installation	289
41.3	Usage	290
41.4	Contributors	290
42	DCH-RP ECSG MOBILE	295
42.1	About	295
42.2	Installation	295
42.3	Usage	296
42.4	Contributors	296
43	EARTHSERVER SG MOBILE	297
43.1	About	297
43.2	Installation	297
43.3	Usage	298
43.4	Contributors	298
44	Semantic Search	301

44.1	About	301
44.2	Installation	301
44.3	Usage	302
44.4	Contributors	303
45	SCOAP3 HARVESTER API	305
45.1	About	305
45.2	Usage	305
45.3	Contributors	305
46	SEMANTIC SEARCH API	307
46.1	About	307
46.2	Usage	307
46.3	Contributors	309
47	TRAINING MATERIALS	311
47.1	Installation and configuration of the development environment	311
47.2	Creation of a simple parameterised portlet	317
47.3	Creation of a portlet that submits a job	323
47.4	Creation of a portlet that submits a special job	328

The CSGF is open source and released under the Apache 2.0 license. All code is available on [GitHub](#)

The documentation is organized in the following sections:

- ***INTRODUCTION*** This section provides an introduction on the CSGF, including its global architecture.
- ***INSTALLATION AND CONFIGURATION*** This section provides a step-by-step guide to install and configure a server hosting a CSGF-based Science Gateway.
- ***CORE SERVICES*** This section includes the documentation of all the core services of the CSGF.
- ***web-docs*** This section includes the documentation of all the web applications that have been integrated in the Science Gateways powered by the CSGF.
- ***MOBILE APPS*** This section includes the documentation of the apps for mobile appliances which are part of the CSGF
- ***API SERVICES*** This section includes the documentation of the APIs written to use some of the CSGF services.
- ***TRAINING MATERIALS*** This section contains a collection of training materials for developers, including the instructions about how to setup the CSGF development environment.

INTRODUCTION

INSTALLATION & CONFIGURATION GUIDE

Sections below explain how to install and configure a Science Gateway and all its components

2.1 Configuring MySQL Database for liferay

2.1.1 Prerequisites - Machine

These instructions describes how to install and configure a mysql server on a Debian 6.0.3

```
root@sg-database:~# lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 6.0.3 (squeeze)
Release:       6.0.3
Codename:      squeeze
```

Install vim:

```
apt-get install vim
```

2.1.2 MySQL Installation & Configuration

Install MySQL server and client:

```
apt-get mysql-client mysql-server
```

Create databases

First you need to set mysql password:

```
mysqladmin -u root password 'rootPassword'
```

Then you can create an user and a database for liferay, and an user and a database for the Catania Grid Engine.

Database lportal

Access as root using the password you have just set:

```
root@sg-database:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3430
Server version: 5.1.49-3 (Debian)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Add a new user and a new database named lportal. Give the user the privileges to access the database. It's important to grant the privileges even for the access from the science gateway

```
CREATE USER 'liferayadmin' IDENTIFIED BY 'fillWithYourPassword';
Query OK, 0 rows affected (0.00 sec)

CREATE DATABASE lportal;
Query OK, 1 row affected (0.00 sec)

GRANT ALL PRIVILEGES ON lportal.* TO 'liferayadmin'@'localhost'
IDENTIFIED BY 'yourPasswd';
Query OK, 0 rows affected (0.05 sec)

GRANT ALL PRIVILEGES ON lportal.* TO 'liferayadmin'@'IPOfsg-server'
IDENTIFIED BY 'yourPasswd';
Query OK, 0 rows affected (0.05 sec)

FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.04 sec)

exit
Bye
```

2.1.3 Troubleshooting

Firewall

In case you are not able to connect to the database from the server check the firewall rules, in particular for the port 3306

Access the database remotely

Check that mysql is enabled to accept remote connections:

```
root@sg-database:~# vim /etc/my.cnf

bind-address          = 0.0.0.0
```

2.2 Installing Liferay 6.1.1 on Glassfish 3.1.x

2.2.1 Preliminar steps

Server(s) requirements

The Science Gateway and the database can be installed either on different machines or on the same one. As we chose the first approach, this is the one that is explained below. From now in on we will refer to these machines as:

Server Name	Value
Science Gateway Server	sg-server
Database Server	sg-database

The next table shows physical configurations for both machines

Server Name	Arch	CPU	RAM	Disk Space
Science Gateway Server	x86_64 GNU/Linux	>= 4 cores (8 recommended)	>= 8 gigas	>= 1 TB
Database Server	x86_64 GNU/Linux	>= 1 core	>= 1 giga	>= 20 GB

The next table shows other configurations for machines. Of course you can choose the operative system you prefer, this is simply our choice.

Server Name	Operative System	TERENA Host Certificate	Network Public Interface
Science Gateway Server	CentOs 6.2	Yes, CoMoDo	Yes
Database Server	Debian 6.0.3	No	No

Verify that your machine has direct and reverse address resolution (check your DNS configuration). Use host command to verify everything works properly:

```
host sg-server.yourdomain.foo
sg-server.yourdomain.foo has address 10.0.0.1

host 10.0.0.1
Host 1.0.0.10.in-addr.arpa domain name pointer sg-server.yourdomain.foo
```

The full list of the hardware server requirements can be downloaded from [here](#)

Software requirements

You must be **root** on the machine to perform next steps.

Let's add some repo files to the existing ones in order to install the required software. Add these three repo files to your system:

- [CentOS-Shibboleth.repo](#)
- [EGI-trustanchors.repo](#)
- [EUGRIDPMA-fetch-crl3.repo](#)

You can use wget command to download the three files into:

```
/etc/yum.repos.d
```

Now you can install the following software:

```
yum clean all
yum update
yum install shibboleth httpd lcg-CA java-1.7.0-openjdk-devel.x86_64 \
```

```
fontpackages-tools mysql.x86_64 mod_ssl.x86_64 \
php.x86_64 vim-enhanced.x86_64 fetch-crl
```

Apache Configuration

To allow the Science Gateway to accept connections on port 80, you need to configure and start Apache. At these steps you need to specify the certificate file for the machine for SSL connection. This certificate file should be supplied from your CA authority.

Edit apache configuration files (you must be **root** to perform the next steps)

Download into

```
/etc/httpd/conf.d/
```

The following files:

- Proxy ajp configuration file
- Virtual host configuration file

Edit the configuration file:

```
vim /etc/httpd/conf.d/virtualhost.conf
...
ServerAdmin sg-serveradminlist@yourdomain.foo
ServerName sg-server.yourdomain.foo
```

Edit the configuration file:

```
vim /etc/httpd/conf/httpd.conf
```

If you find a line like the following:

```
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
```

comment it.

Make sure that

```
KeepAlive is Off
```

After this editing start the server:

```
/etc/init.d/httpd start
```

or

```
service httpd start
```

configure apache to start at boot

```
chkconfig --level 2345 httpd on
```

Create liferayadmin user

It is important to install liferay and its application server (i.e. glassfish) as a **normal** user and not root. For this reason, before continuing with the installation, create a specific user and use it to execute the next commands:

```
adduser liferayadmin
su - liferayadmin
```

2.2.2 Glassfish Installation

Download [Glassfish](#) source files. The version we currently in use in our production server is [GlassFish Server Open Source Edition 3.1 \(build 43\)](#). Otherwise the release supported by liferay 6.1.1 is [GlassFish Server Open Source Edition 3.1.2.2 \(build 5\)](#). Unpack the zip archive in:

```
/opt/
```

You may have to use *chown* and *chgrp* to change the directory permissions to the normal user. e.g. “*chown -R liferayadmin /opt/glassfish3/*”.

When you create a domain for liferay in glassfish, you will be asked for a username and password. This is the admin user for your application server.

```
[liferayadmin@sg-server ~]$ cd /opt/glassfish3/bin/
[liferayadmin@sg-server bin]$ sh asadmin create-domain liferay
Enter admin user name [Enter to accept default "admin" / no password]> liferayadmin
Enter the admin password [Enter to accept default of no password]>
Enter the admin password again>
Using port 4848 for Admin.
Using default port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Using default port 9009 for JAVA_DEBUGGER.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=oldliferay2,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=oldliferay2-instance,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=Calif...
No domain initializers found, bypassing customization step
Domain test created.
Domain test admin port is 4848.
Domain test allows admin login as user "liferayadmin" with no password.
Command create-domain executed successfully.
```

Remember to edit the firewall rules using iptables to open the correct ports (4848, 8080).

Edit the configuration file in other to increase the size of the virtual machine used by glassfish (search the section of *jvm-options*). This can also be done through the glassfish administration interface.

```
vim /opt/glassfish3/glassfish/domains/liferay/config/domain.xml
```

```
<jvm-options>-server</jvm-options> <!-- change this, the original value is -client -->
<jvm-options>-XX:MaxPermSize=512m</jvm-options>
<jvm-options>-Xms4096m</jvm-options>
<jvm-options>-Xmx4096m</jvm-options>
<jvm-options>-XX:MaxNewSize=700m</jvm-options>
<jvm-options>-XX:NewSize=700m</jvm-options>
<jvm-options>-XX:SurvivorRatio=10</jvm-options>
<jvm-options>-Dfile.encoding=UTF8</jvm-options>
<jvm-options>-Djava.net.preferIPv4Stack=true</jvm-options>
```

```
<jvm-options>
  -Dorg.apache.catalina.loader.WebappClassLoader.ENABLE_CLEAR_REFERENCES=false
</jvm-options>
<jvm-options>-Duser.timezone=GMT</jvm-options>
```

Configure glassfish to access the database

Liferay needs a database to run. Instead of accessing it directly, Liferay can use a Connection Pool defined in Glassfish to open a connection to the database server. Running the following command the connections will be created.

Before initiating the command, you need to start the glassfish instance:

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin start-domain liferay
Waiting for liferay to start .....
Successfully started the domain : liferay
domain Location: /opt/glassfish3/glassfish/domains/liferay
Log File: /opt/glassfish3/glassfish/domains/liferay/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
```

Now you can run the command:

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin \
-u liferayadmin create-jdbc-connection-pool \
--datasourceclassname com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource \
--restype javax.sql.ConnectionPoolDataSource \
--property \
"user=liferayadmin:password=liferayadminMySQLPasswrod:\
url='jdbc:mysql://sg-database:3306/lportal'" LiferayPool

[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin -u \
liferayadmin create-jdbc-resource \
--connectionpoolid LiferayPool jdbc/liferay
```

In this way, we are setting a connection pool able to connect to a machine with the hostname **sg-database** using the default port **3306** for the database. In the database there is a table called **lportal** that can be read/write by a user named **liferayadmin** identified by the password **liferayadminMySQLPasswrod**. From now in on we will be able to refer to this resource thanks to the name we assigned: **jdbc/liferay**. In order to configure the database properly, please refer to the [Configuring MySQL Database for liferay](#).

Create a proxy ajp listener

In order to bind glassfish with apache, you must create a proxy ajp listener. After the connector is created, you need to stop the server.

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin create-network-listener \
--listenerport 8009 --protocol http-listener-1 --jkenabled true apache
Command create-network-listener executed successfully.
```

Now stop the server:

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin stop-domain liferay
Waiting for the domain to stop .....
Command stop-domain executed successfully.
```

2.2.3 Liferay Installation

Liferay is a web application, and so we need to deploy it on Glassfish. Before the deployment, we need to provide the correct library in Glassfish.

Liferay files

Considering Liferay needs to use a MySQL database, a driver is needed. Copy the mysql connector in the path:

```
[liferayadmin@sg-server ~]$ /opt/glassfish3/glassfish/domains/liferay/lib/
```

You can download the java connector for your version of mysql server from the [official site](#) or download [ours](#).

Now you can copy liferay's jar. Liferay refers to these file as liferay portal dependencies. [From here](#) you can find the full list of liferay files. There are different dependencies corresponding to the different liferay version. To install *Liferay 6.1.1 CE GA2* download dependencies from [this link](#). After downloading, extract the archive and copy the jar file into the same path of mysql java connector (see the example below):

```
[liferayadmin@sg-server ~]$cp liferay-portal-dependencies-6.1.1-ce-ga2/*.jar \
/opt/glassfish3/glassfish/domains/liferay/lib
[liferayadmin@sg-server ~]$ tree /opt/glassfish3/glassfish/domains/liferay/lib
/opt/glassfish3/glassfish/domains/liferay/lib
-- applibs
-- classes
-- databases
-- ext
-- hsql.jar
-- mysql-connector-java-5.1.35-bin.jar
-- portal-service.jar
-- portlet.jar
```

Liferay deploy

A web application is identified by an archive with extension .war. Download the liferay portal .war from [the Liferay sourceforge repository](#)

Start glassfish in order to deploy the .war:

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin start-domain liferay
```

Once you get back the prompt, you can deploy the .war file with the command (supposing you downloaded it into the liferayadmin home)

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin -u liferayadmin deploy \
--contextroot / --verify=true \
--name liferay611cega2 ~/liferay-portal-6.1.1-ce-ga2-20120731132656558.war
```

You will be asked for the glassfish admin user password. To check the status of the deploy you can refer to the glassfish log file.

```
tail -f /opt/glassfish3/glassfish/domains/liferay/logs/server.log
```

You can also type

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin list-domains
```

Once the deployment is finished we can stop the server to customise the liferay installation:

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin stop-domain liferay
```

If the deployment has been completed successfully you will find the liferay files in:

```
/opt/glassfish3/glassfish/domains/liferay/applications/liferay61lcega2
```

Edit the liferay portal properties file to connect it to the database:

```
vim /opt/glassfish3/glassfish/domains/liferay/applications/liferay61lcega2/\
WEB-INF/classes/portal-ext.properties

jdbc.default.jndi.name=jdbc/liferay

web.server.http.port=80
web.server.https.port=443

# Parameter in other to avoid Lifery to append sessionId to link
session.enable.url.with.session.id=false

# In order not to show portlet that can't be visualized by the user
layout.show.portlet.access.denied=false

# Set this to true to convert the tracked paths to friendly URLs.
#session.tracker.persistence.enabled=true
#session.tracker.friendly.paths.enabled=true
#
# Set this to true to enable the ability to compile tags from the URL.
# Disabling it can speed up performance.
#
tags.compiler.enabled=false

#
# Disable locale in friendly url
#
locale.prepend.friendly.url.style=0

# Configure email notification settings.
admin.email.from.name=Liferay Administrator Name
admin.email.from.address=LiferayAdministratorMail@yourdomain

## Live Users
## Set this to true to enable tracking via Live Users.
live.users.enabled=false

session.tracker.persistence.enabled=true
```

Now you can start glassfish again:

```
[liferayadmin@sg-server ~]$ sh /opt/glassfish3/bin/asadmin start-domain liferay
```

If everything is ok you should find the default liferay instance at:

<http://sg-server:8080>

2.2.4 Post Installations

Make glassfish domain start at boot

Edit the rc.local file in order to make glassfish start in case the server reboots:

```
[root@sg-server ~]# vim /etc/rc.local
...
su -c "sh /opt/glassfish3/glassfish/bin/asadmin start-domain liferay" - liferayadmin
```

Where you specify that the user **liferayadmin** (and not root) will start the process automatically at boot.

Install Marketplace Portlet

Download the [Marketplace portlet](#) and deploy on the portal using the following command:

```
[liferayadmin@sg-server ~]$ cp marketplace-portlet-6.1.2.4.war /opt/glassfish3/deploy/
```

Check the log file to see if the portlet is correctly deployed, you should see some line like the following in the server.log file:

```
...
...Successfully autodeployed : \
/opt/glassfish3/glassfish/domains/liferay/autodeploy/marketplace-portlet.1#]
...
```

In order to use the Marketplace portlet you need to create your own account, please create a new one, if you don't already have it. Then open your portal installation, select *Go to -> Control Panel* from the top right corner and *Stro* from the left menu. Fill the fields with your Liferay credentials, look for *Web form* and select the free Web Form CE portlet, click on *Purchase* button (this just make available this portlet for your Liferay account). Now from the left side menu select *Purchased* and click the Install button on the Web Form portlet, wait until the installation process ends.

2.2.5 Troubleshooting

Glassfish Port

If your network is not configured properly you could not be able to start glassfish and you will get this error:

```
There is a process already using the admin port 4848 -- \
it probably is another instance of a GlassFish server.
Command start-domain failed.
```

If you are sure there is no process using that port (use `nmap -sT -O localhost` or a variation), check that the address configured for your machine is correct and that it corresponds to the correct hostname configured.

As a good rule, you should set them in the /etc/hosts files as below:

```
[root@sg-server ~]# vim /etc/hosts
...
10.0.0.1    sg-server.yourdomain.foo    sg-server
```

Glassfish Connection Pools

It is important to configure the connection pools properly. If you don't, Liferay will not be able to start, or it's possible it will still use the database on file, that should not be used on a production server.

Glassfish has a web interface. Access it and check if the connection to the database works properly. To access glassfish: <http://sg-server:4848>

and fill with username liferayadmin and the password you set for the glassfish administrator.

Navigating on the left tree you can check the resources you created during the configuration process. Check the list of the JDBC Resources:

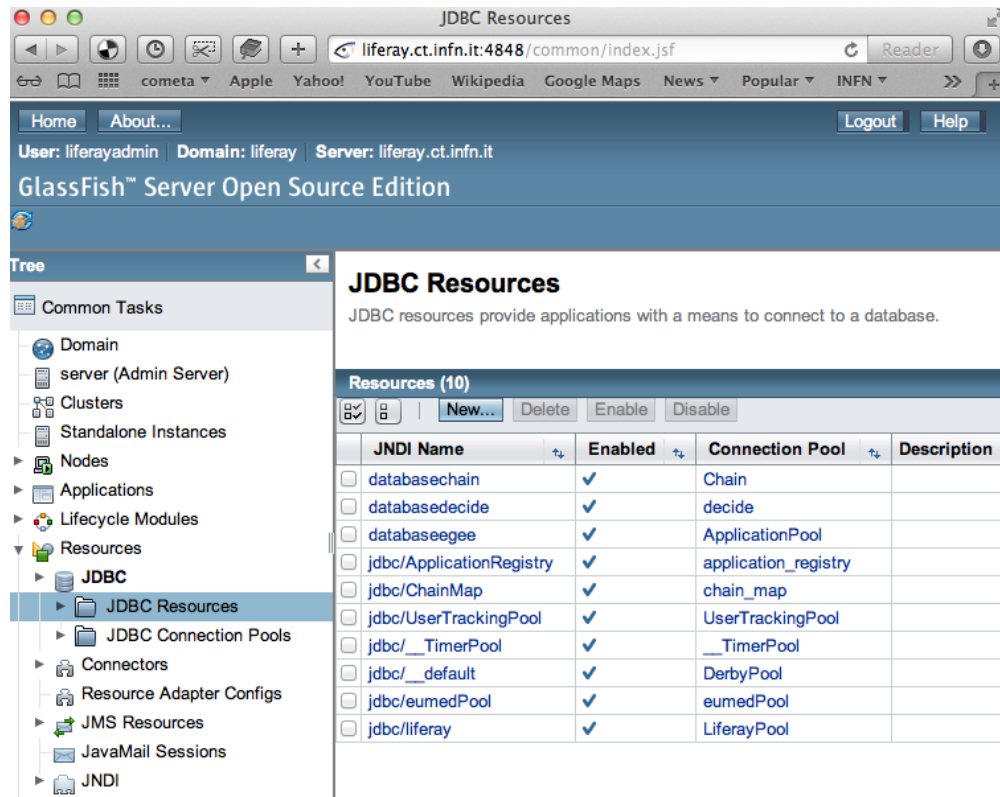


Fig. 2.1: JDBC Resources

JDBC Connection Pools:

Check the additional properties for Liferay Pool:

In case all the parameters are set correctly try to ping the database:

Liferay theme not loaded properly

If the start page is not loaded properly, before or after the configuration wizard, there could be some files created by liferay directory that have the wrong write permissions.

As root check the /tmp directory:

```
[root@sg-server ~]# cd /tmp/
[root@sg-server tmp]# ls -l
total 16
drwxr-xr-x  2 liferayadmin liferayadmin 4096 Mar  4 18:46 hsperrdata_liferay
drwxr-xr-x.  3 root      root      4096 Mar  4 18:48 liferay
drwxr-xr-x.  2 liferayadmin liferayadmin 4096 Feb 28 17:40 xuggle
```

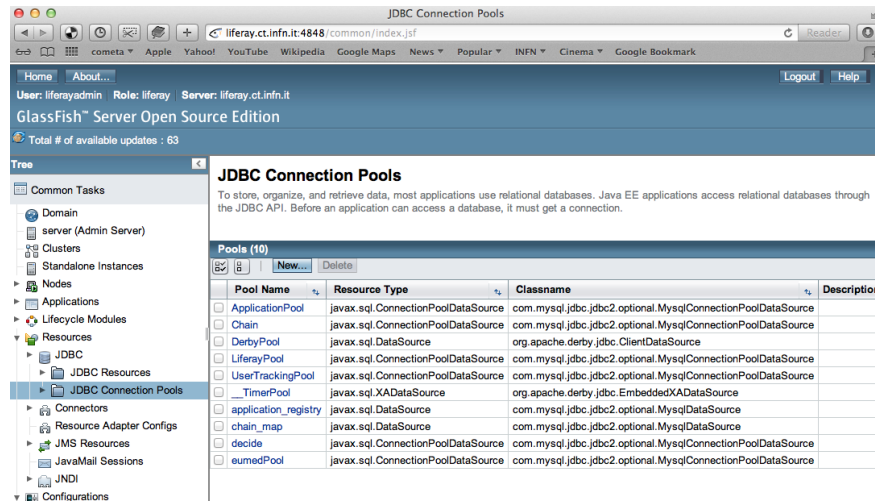


Fig. 2.2: JDBC Connection Pools

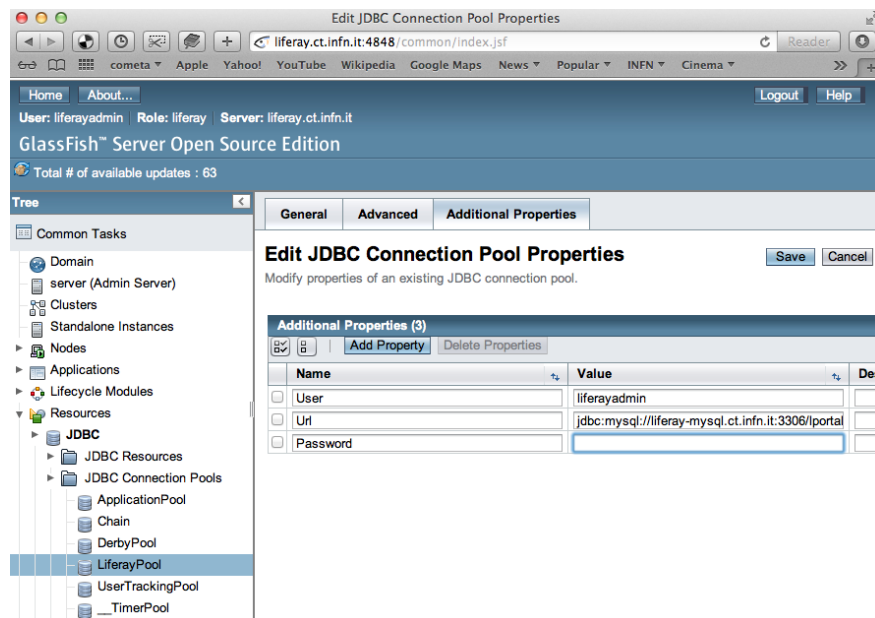


Fig. 2.3: Liferay Pool additional properties

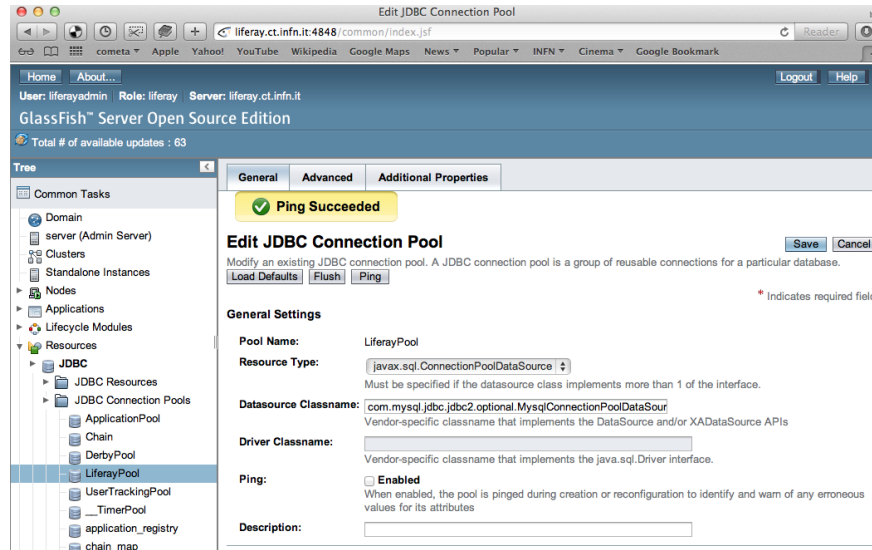


Fig. 2.4: Liferay Pool Ping test

and if you have a content like the one above change the owner of the liferay directory:

```
[root@science-gateway tmp]# chown -R liferayadmin.liferayadmin liferay/
[root@science-gateway tmp]# ls -l
total 16
drwxr-xr-x  2 liferayadmin liferayadmin 4096 Mar  4 18:46 hspcrfdata_liferay
drwxr-xr-x. 3 liferayadmin liferayadmin 4096 Mar  4 18:48 liferay
drwxr-xr-x. 2 liferayadmin liferayadmin 4096 Feb 28 17:40 xuggle
```

Maximum Number of file

Check what is the maximum number of file the operative system can open:

```
[liferayadmin@sg-server ~]$ cat /proc/sys/fs/file-max
1610813
```

In case the number is too low set an higher value in the variable:

```
vim /etc/sysctl.conf
# Controls the maximum number of opened files
fs.file-max=2000000
```

SELinux

In case you are not able to start apache server properly you should check you SELinux configurations.

To view your SELinux status type

```
[liferayadmin@sg-server ~]$ getenforce
Enforcing
```

In this case SELinux is enabled. You should edit its policy in order to allow apache and shibboleth work properly. Otherwise you have to disable it.

To temporary disable it, as root, run:

```
[root@sg-server ~]# setenforce 0
```

In case you want to permanent disable it, you need to edit this file and reboot (always as root):

```
vim /etc/selinux/config
....
SELINUX=disables
```

2.3 Enabling LDAP Authentication

This page explains how our LDAP server is configured in order to allow authentication and authorisation of users by an Identity Provider and a Service Provider.

2.3.1 LDAP Configuration

The following sections describes branches present in our LDAP and what any branch is meant to. For each branch is shown all user **attributes (UA)** and all **operational attributes (OA)**.

LDAP root <dc=local>

It's the route of the LDAP server

```
# extended LDIF
#
# LDAPv3
# base <dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#
# local
dn: dc=local
objectClass: dcObject
objectClass: organization
dc: local
o: INFN
# search result
search: 2
result: 0 Success
# numResponses: 2
# numEntries: 1
```

Administrator user for idp <cn=idp,dc=local>

It's the administrator user who register users in the IDP

```
# extended LDIF
#
# LDAPv3
# base <cn=idp,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#
```

```
# idp, local
dn: cn=idp,dc=local
objectClass: top
objectClass: person
objectClass: simpleSecurityObject
sn: IDPCT
cn: idp
description:: QWNjb3VudCB1c2F0byBwZXIgbGEgZ2VzdGlvbmUgZGVsbGUgaWRlbnRpdMOgIGZh
dHRIIGRhIEleEUE9QRU4=

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Administrator user for the Science Gateway <cn=liferayadmin,dc=local>

It's the administrator user who is configured in the Science Gateway

```
# extended LDIF
#
# LDAPv3
# base <cn=liferayadmin,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#
# liferayadmin, local
dn: cn=liferayadmin,dc=local
objectClass: top
objectClass: person
objectClass: simpleSecurityObject
cn: liferayadmin
sn: Liferay

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Country grouping organisations <c=IT,ou=Organisations,dc=local>

We group all users' organisations according to their country. For example all Italian organisations are stored in this branch.

```
# extended LDIF
#
# LDAPv3
# base <c=IT,ou=Organisations,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#
# IT, Organisations, local
dn: c=IT,ou=Organisations,dc=local
objectClass: top
```

```
objectClass: country
objectClass: friendlyCountry
c: IT
co: Italy
description: Europe, Southern Europe

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Example of organisations <o=INFN,c=IT,ou=Organisations,dc=local>

The example below shows the entry for the INFN which is an Italian organisation.

```
# extended LDIF
#
# LDAPv3
# base <o=INFN,c=IT,ou=Organisations,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#

# INFN, IT, Organisations, local
dn: o=INFN,c=IT,ou=Organisations,dc=local
objectClass: top
objectClass: organization
description: National Institute of Nuclear Physics
o: INFN
registeredAddress: http://www.infn.it

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Example of division of organisation <ou=Catania,o=INFN,c=IT,ou=Organisations,dc=local>

When an organisation has many divisions they are grouped inside it. The example below shows the Division of Catania of INFN

```
# extended LDIF
#
# LDAPv3
# base <ou=Catania,o=INFN,c=IT,ou=Organisations,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#

# Catania, INFN, IT, Organisations, local
dn: ou=Catania,o=INFN,c=IT,ou=Organisations,dc=local
objectClass: top
objectClass: organizationalUnit
ou: Catania
registeredAddress: www.ct.infn.it
postalAddress: Via Santa Sofia 64, 95123, Catania
```

```
# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
~~~~~
```

Example of Services <dc=idpct,ou=Services,dc=local>

In this branch all users who can be authenticated by the idp are stored. These are distinguished by the administrator because they are stored in a different group.

```
# extended LDIF
#
# LDAPv3
# base <dc=idpct,ou=Services,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#

# idpct, Services, local
dn: dc=idpct,ou=Services,dc=local
objectClass: dcObject
objectClass: domainRelatedObject
objectClass: domain
objectClass: top
dc: idpct
associatedDomain: idp.ct.infn.it

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

IDP Administrators <cn=Administrator,ou=Group,dc=idpct,ou=Services,dc=local>

In the example below, the list of all users able to administrate the ldap is shown (i.e. to edit it).

User's group is configured so that user's dn is unique.

As you can see, together with administrator1, administrator2 and administrator3, there is the liferayadmin users that is configured in liferay to make the two services communicate.

Differently, administrator1, as a real user, is present the group People.

```
# extended LDIF
#
# LDAPv3
# base <cn=Administrator,ou=Group,dc=idpct,ou=Services,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#

# Administrator, Group, idpct, Services, local
dn: cn=Administrator,ou=Group,dc=idpct,ou=Services,dc=local
objectClass: top
objectClass: groupOfUniqueNames
```



```

uniqueMember: cn=administrator1,ou=People,dc=local
uniqueMember: cn=administrator2,ou=People,dc=local
uniqueMember: cn=administrator3,ou=People,dc=local
uniqueMember: cn=liferayadmin,dc=local
cn: Administrator
description: Users in this group have administrative privileges on this server

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1

```

List of IDP Users <cn=Users,ou=Group,dc=idpct,ou=Services,dc=local>

In the same group we have the list of users who can be authenticated by the IDP. In order to make these users log in the Science Gateway (SG) is possible to connect the SG to this services or create a brand new services. This new branch then will be the one responsible for the authorisation. Liferay will do a map 1:1 between the group items present in the services and the different roles assigned to the user of the SG. For example let's suppose we use IDP branch for both authentication and authorisation. In this special case the user with cn=administrator1 will be administrator both of IDP and of the Science Gateway.

```

# extended LDIF
#
# LDAPv3
# base <cn=Users,ou=Group,dc=idpct,ou=Services,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: ALL
#

# Users, Group, idpct, Services, local
dn: cn=Users,ou=Group,dc=idpct,ou=Services,dc=local
uniqueMember: cn=fmarco76,ou=People,dc=local
uniqueMember: cn=rotondo,ou=People,dc=local
uniqueMember: cn=barbera,ou=People,dc=local
uniqueMember: cn=brunor,ou=People,dc=local
....
....
cn: Users
description: List of users using this server for authentication
objectClass: top
objectClass: groupOfUniqueNames

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1

```

User branch example <cn=rotondo,ou=People,dc=local>

Below we have all the information related to an user.

```

# extended LDIF
#
# LDAPv3
# base <cn=rotondo,ou=People,dc=local> with scope baseObject

```

```
# filter: (objectClass=*)
# requesting: ALL
#

# rotondo, People, local
dn: cn=rotondo,ou=People,dc=local
cn: rotondo
displayName: rotondo
initials: RR
mail: riccardo.rotondo@ct.infn.it
mail: riccardo.rotondo@garr.it
mail: net.ricky@gmail.com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
sn: Rotondo
givenName: Riccardo
registeredAddress: riccardo.rotondo@ct.infn.it
o: o=INFN,c=IT,ou=Organisations,dc=local
o: o=GARR,c=IT,ou=Organisations,dc=local
ou: ou=Catania,o=INFN,c=IT,ou=Organisations,dc=local
postalAddress: Via Santa Sofia 64, 95123, Catania
telephoneNumber: 00 39 095 3785519
mobile: Skype: riccardo.ro
title: Mr.

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Let's explain some fields in more detail

cn

Mandatory field

It's the screen name associated to the user. It appears in the dn and must be unique as it's used by the services group for authentication and/or authorisation.

display name

Mandatory field

It's used by the Science Gateway, it can be set equal to the cn.

sn

Mandatory field

User surname

givenName

User first name

mail

Mandatory field

As you can see from the example, this value accepts many variables. It's used to specify different mail corresponding to different identity provider.

registeredAddress

Mandatory field

It's the mail recognised by the IDP and it's the id used by the Science Gateway to identify to the user.

o

Mandatory field

It's the organisation the user belong to.

ou

Optional field

In case it exists, it's the division of the organisation.

Others fields

All others entry: initials, title, postalAddress, telephoneNumber, mobile are **optional field**.

******Operational attributes for user <cn=rotondo,ou=People,dc=local> +:

Once a user is present in the People branch, the dn can be inserted in services where the users need to access. This operation will modify automatically the user item adding the operational attributes corresponding to that services.

A query to ldap requesting the operational attributes for example, show that this user can be authenticated by the IDP as a User ***(memberOf: cn=Users,ou=Group,dc=idpct,ou=Services,dc=local)*** and is authorised to access to the Science Gateway that refers to the service sgw ***(memberOf: cn=GenericUser,ou=Group,dc=sgw,ou=Services,dc=local)***. Moreover he is not only a simple user but he owns other roles, Administrator and CloudManager ***(memberOf: cn=Administrator,ou=Group,dc=sgw,ou=Services,dc=local memberOf: cn=CloudManager,ou=Group,dc=sgw,ou=Services,dc=local)***.

```
# extended LDIF
#
# LDAPv3
# base <cn=rotondo,ou=People,dc=local> with scope baseObject
# filter: (objectClass=*)
# requesting: +
#
# rotondo, People, local
dn: cn=rotondo,ou=People,dc=local
structuralObjectClass: inetOrgPerson
entryUUID: ac78cfbc-4bbe-1030-8b49-c57783a62e4d
creatorsName: cn=admin,dc=local
createTimestamp: 20110726103506Z
memberOf: cn=Users,ou=Group,dc=idpct,ou=Services,dc=local
memberOf: cn=GenericUser,ou=Group,dc=sgw,ou=Services,dc=local
memberOf: cn=Administrator,ou=Group,dc=sgw,ou=Services,dc=local
memberOf: cn=Administrator,ou=Group,dc=sgw,ou=Services,dc=local
pwdChangedTime: 20110726104012Z
entryCSN: 20130313091536.316389Z#000000#000#000000
modifiersName: cn=rotondo,ou=People,dc=local
modifyTimestamp: 20130313091536Z
entryDN: cn=rotondo,ou=People,dc=local
subschemaSubentry: cn=Subschema
hasSubordinates: FALSE
```

```
# search result
search: 2
result: 0 Success
```

Service Provider Configuration

Add CA Certificate to the keystore

Before configuring Liferay you should verify, for example with a utility such as `ldapsearch`, that SSL communication is possible between the service provider (from now on `sg-server`) and the ldap server (from now on `ldap-server`).

Before doing the first test, make sure you have added the certificate of the certification authority that released your `ldap-server` certificate. This certificate must be added to the glassfish keystore, which is the application server we use in the science gateway, our service provider.

The path of the keystore can be seen with:

```
[liferayadmin@sg-server ~]$ ps aux |grep glassfish
500          1659   1.4 19.6 8295936 3208572 pts/1 Sl   Feb19 465:35 /usr/lib/jvm/java-1.7.0-openjdk-1.7.
500 3990 0.0 0.0 103236 880 pts/0 S+ 10:59 0:00 grep glassfish
```

Suppose that your CA certificate location is:

```
/home/liferayadmin/INFN_CA.cer
```

execute:

```
liferayadmin@sg-devel ~]$ keytool -import -alias infn-ca -file /home/lif\
erayadmin/INFN_CA.cer -keystore /opt/glassfish3/glassfish/domains/liferay/config/cacerts.jks
```

Test if you are able to get the list of users with:

```
liferayadmin@sg-devel ~]$ ldapsearch -x -H ldaps://ldap-server -b ou=People,dc=local
```

2.3.2 Shibboleth Configuration

You will need to configure Shibboleth, which is the federated identity solution that Catania-SG uses.

The file `shibboleth2.xml` in `/etc/shibboleth` contains the configurations necessary.

```
root@sg-devel ~]$ vim /etc/shibboleth/shibboleth2.xml
```

Locate the `SSO entityID=` and replace the information below.

```
<SSO entityID="https://idp.someaddress.com/idp/shibboleth">
  SAML2 SAML1
</SSO>
```

Secondly, you will need to request that the LDAP provider sends you their LDAP metadata, in XML form, which is then added to a file called **partner-metadata.xml** in the `/etc/shibboleth` directory.

```
root@sg-devel ~]$ vim /etc/shibboleth/partner-metadata.xml
```

Thirdly, you will need to edit `proxy_ajp.conf` file

```
root@sg-devel ~]$ vim /etc/httpd/conf.d/proxy_ajp.conf
```

and uncomment the following lines

```
ProxyPass /shibboleth/ !
ProxyPass /Shibboleth.sso/ !
ProxyPass / ajp://localhost:8009/
```

Then ensure that the **shibd** daemon has been started, and will start again if the server reboots.

```
:::shell
root@sg-devel ~]$ service shibd start
root@sg-devel ~]$ chkconfig shibd on
```

2.3.3 Liferay Configuration

You need to edit the liferay portal-ext.properties file:

```
vim /opt/glassfish3/glassfish/domains/liferay/applications/liferay611cega2/WEB-INF/classes/portal-ext.properties

# LDAP server configuration

ldap.import.method=group
ldap.import.enabled=true
ldap.import.create.role.per.group=true
ldap.import.interval=3
```

Restart the domain.

Now access your liferay server from the web interface.

Click on the top right *Go to -> Control Panel*

Then on the left *Portal Settings*

On the right *Authentication*

In the top bar click LDAP

Set the option as in figure:

Portal Settings

Authentication

General

LDAP

CAS

Facebook

NTLM

OpenID

Open SSO

SiteMinder



☒ Enabled

☐ Required

LDAP Servers

Add

GARR Idap server

Import / Export


☒ Import Enabled

☒ Import on Startup Enabled

☐ Export Enabled

Password Policy

☐ Use LDAP Password Policy

 **LIFERAY.**

GARR Science Gateway

Configuration

[General](#)

Authentication

[Users](#)

[Mail Host Names](#)

[Email Notifications](#)

Identification

[Addresses](#)

[Phone Numbers](#)

[Additional Email Addresses](#)

[Websites](#)

Miscellaneous

[Display Settings](#)

[Google Apps](#)

Save

Cancel

Now click on ADD to add and Idap server that liferay will contact to authorise users.

Portal Settings

edit-ldap-server

[« Back](#)

Server Name

Default Values

- ☐ Apache Directory Server
- ☐ Fedora Directory Server
- ☐ Microsoft Active Directory Server
- ☐ Novell eDirectory
- ☐ OpenLDAP
- ☐ Other Directory Server

Connection

Base Provider URL

Base DN

Principal

Credentials

Users

Authentication Search Filter

Import Search Filter

If you organised your Idap as ours here there is the list of value you need to add:

Connection

Server Name: Your ldap server name

Base Provider URL: ldaps://your-ldap-hostname:636

Base DN: dc=local

Principal: cn=liferayadmin,dc=local

Credentials: liferayadmin-password

Users

Authertication Search Filter: (&(cn=@screen_name@)(memberOf=cn=GenericUser,ou=Group,dc=sgw,ou=Services,dc=local))

Import Search Filter: (&(objectClass=inetOrgPerson)(memberOf=cn=GenericUser,ou=Group,dc=sgw,ou=Services,dc=local))

Screen Name: cn

Password: userPassword

Email Address: registeredAddress

First Name: givenName

Last Name: sn

Job Title: title

Group

memberOf

Groups

Import Search Filter: (&(objectClass=groupOfUniqueNames)(o=dc=sgw,ou=Services,dc=local))

Group Name: cn

Description: description

User: uniqueMember

Export

Users DN: ou=People,dc=local

User Default Object Classes: top,person,inetOrgPerson,organizationalPerson

Groups DN: ou=Group,dc=sgw,ou=Services,dc=local

Group Default Object Classes: top,groupOfUniqueNames

Click Save

This pages explains how to configure your Service Provider in order to demand authentication to Shibboleth

2.4 Configuring a Service Provider for Shibboleth Authentication

An apache server and shibboleth installed on the machine

Configure Apache files

You should have some apache files configure as follow (usually store in */etc/httpd/conf.d*)

*shib.conf


```
vim shib.conf
...
LoadModule mod_shib /usr/lib64/shibboleth/mod_shib_22.so

<Location /secure>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user
</Location>
...
```

*shibSec.conf

```
vim shibSec.conf
...
#
# Configuration for Liferay Login
#

<Location /c/portal/login>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user
</Location>
<Location /not_authorized>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user
</Location>

## Configuration for metadata

Alias /shibboleth/ "/var/www/metadata/"

<Directory "/var/www/metadata">
</Directory>
...
```

*proxy_ajp.conf

Once you configure the custom url you need to avoid them to contact the glassfish listener

```
vim proxy_ajp.conf
...
ProxyPass /shibboleth/ !
ProxyPass /Shibboleth.sso/ !
ProxyPass / ajp://localhost:8009/
...
```

Configure Liferay to contact Shibboleth for authentication

Install Shibboleth plugin

Download the [shibboleth plugin from here](<http://sourceforge.net/projects/ctsciencegtwys/files/catania-science-gateway/plugins/ShibbolethLib-1.0.jar/download>) and copy it to:

```
/opt/liferay/glassfish3/glassfish/domains/liferay/applications/liferay61lcega2/WEB-INF/lib
```

Now edit the portlet-ext.properties adding these lines:

```
vim /opt/liferay/glassfish3/glassfish/domains/liferay/applications/liferay611cega2/WEB-INF/classes/po
...
# Shibboleth Config (Remember to install the Shibboleth plugin)

auto.login.hooks=it.infn.ct.security.shibboleth.ShibbolethAutoLogin,com.liferay.portal.security.auth
auth.login.url=/c/portal/login

default.logout.page.path=/Shibboleth.sso/Logout
logout.events.post=com.liferay.portal.events.LogoutPostAction,it.infn.ct.security.shibboleth.Shibbol
...
```

Finally insert the filter in web.xml

```
vim /opt/glassfish3/glassfish/domains/liferay/applications/liferay611cega2/WEB-INF/web.xml

...
<filter>
  <filter-name>Shibboleth Filter</filter-name>
  <filter-class>it.infn.ct.security.shibboleth.filters.ShibbolethFilter</filter-class>
  <init-param>
    <param-name>auth_failure_redirect</param-name>
    <param-value>/not_authorized</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>Shibboleth Filter</filter-name>
  <url-pattern>/c/portal/login</url-pattern>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>FORWARD</dispatcher>
</filter-mapping>
...
```

References

<https://wiki.shibboleth.net/confluence/display/SHIB2/MetadataForSP>

2.5 Configuring the MySQL Database for the Grid & Cloud Engine

Let's suppose you already installed the MySQL server on the machine and you configured the root user and password. If you don't refer to the [Configuring MySQL Database for liferay](#) guide. Access as root using the password you have just set:

```
root@sg-database:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3430
Server version: 5.1.49-3 (Debian)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Add a new user and a new database named usertracking. Give the user the privileges to access the database. It's important to grant the privileges even for the access from the science gateway

```
CREATE USER 'tracking_user' IDENTIFIED BY 'usertracking';
Query OK, 0 rows affected (0.00 sec)

CREATE DATABASE usertracking;
Query OK, 1 row affected (0.00 sec)

GRANT ALL PRIVILEGES ON usertracking.* TO 'tracking_user'@'localhost'
IDENTIFIED BY 'usertracking';
Query OK, 0 rows affected (0.05 sec)

FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.04 sec)

exit
Bye
```

You need to copy the empty schema for the database. Download [this file](#) and:

```
root@s-g-database:~# mysql -u tracking_user -p usertracking
Enter password:

mysql> source UsersTrackingDB.sql;
Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.02 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Database changed
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.26 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.28 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.18 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.18 sec)
```

```
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.24 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.26 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 722 rows affected (0.06 sec)
Records: 722 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.23 sec)
```

2.6 Configuring the Grid & Cloud Engine on Liferay

2.6.1 Prerequisites

Check java version using the following command:

```
[liferayadmin@centos6 ~]$ java -version
java version "1.7.0_79"
OpenJDK Runtime Environment (rhel-2.5.5.3.el6_6-x86_64 u79-b14)
OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
```

the version installed should be 1.7, if you have installed a previous one, please update it before to proceed.

As root user, download the [vomkdir.tar.gz](#)

```
[root@centos6 ~]# mv vomkdir /etc/grid-security/
```

Create the directory (as liferayadmin)

```
[liferayadmin@centos6 ~]$ mkdir /tmp/jobOutput/
```

MySQL Server Configuration

Remember that you need to configure the database. You can do that following [Configuring the MySQL Database for the Grid & Cloud Engine](#)

2.6.2 Grid & Cloud Engine Installation

Before starting the installation make sure your liferay domains is stopped.

```
[liferayadmin@centos6 ~]$ /opt/glassfish3/bin/asadmin stop-domain liferay
Waiting for the domain to stop .....
Command stop-domain executed successfully.
```

Dependencies

Download Grid & Cloud Engine and JSAGA libraries from [here](#)

Unzip GridEngine_v1.5.9.zip:

```
[liferayadmin@centos6 ~]$ unzip GridEngine_v1.5.10.zip
```

copy the extracted lib folder under the liferay domain folder:

```
[liferayadmin@centos6 ~]$ cp -r lib /opt/glassfish3/glassfish/domains/liferay/
```

Configuration

LogFile

Download the attached GridEngineLogConfig.xml [link](#), and move this file to the Liferay config folder:

```
[liferayadmin@centos6 ~]$ mv GridEngineLogConfig.xml \
/opt/glassfish3/glassfish/domains/liferay/config
```

Glassfish Configuration

Restart the Glassfish server and when the server is up access the web administration console:

[\[http://sg-server:4848\]](http://sg-server:4848)

fill with username liferayadmin and the password you set for the glassfish administrator and create the required resources.

JNDI Resources

Select *Resources* -> *JNDI* -> *Custom Resources* from left panel. Then on the right panel you can create the resources by clicking the **New...** button.

1. Create **GridEngine-CheckStatusPool** with the following parameters [\[1\]](#):

- **JNDI Name:** GridEngine-CheckStatusPool
- **Resource Type:** it.infn.ct.ThreadPool.CheckJobStatusThreadPoolExecutor
- **Factory Class:** it.infn.ct.ThreadPool.CheckJobStatusThreadPoolExecutorFactory
- **Additional Properties:**
 - **corePoolSize:** 50

- **maximumPoolSize:** 100
- **keepAliveTime:** 4
- **timeUnit:** MINUTES
- **allowCoreThreadTimeOut:** true
- **prestartAllCoreThreads:** true

Edit Custom Resource
Edit an existing JNDI custom resource.
[Load Defaults](#)

JNDI Name: GridEngine-CheckStatusPool

Resource Type: ☐ java.util.Properties ☒ it.infn.ct.ThreadPool.CheckJobStatusThreadPoolExecutor
Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

Factory Class:
Factory class for resource; implements javax.naming.spi.ObjectFactory

Description:

Status: ☒ Enabled

Additional Properties (6)

Name	Value
timeUnit	MINUTES
allowCoreThreadTimeOut	true
prestartAllCoreThreads	true
corePoolSize	50
maximumPoolSize	100
keepAliveTime	4

Fig. 2.5: GridEngine-CheckStatusPool JNDI Resource

2. Create GridEngine-Pool with the following parameters [2]:

- **JNDI Name:** GridEngine-Pool
- **Resource Type:** it.infn.ct.ThreadPool.ThreadPoolExecutor
- **Factory Class:** it.infn.ct.ThreadPool.ThreadPoolExecutorFactory
- **Additional Properties:**
 - **corePoolSize:** 50
 - **maximumPoolSize:** 100
 - **keepAliveTime:** 4
 - **timeUnit:** MINUTES
 - **allowCoreThreadTimeOut:** true
 - **prestartAllCoreThreads:** true

3. Create JobCheckStatusService with the following parameters [3]:

- **JNDI Name:** JobCheckStatusService
- **Resource Type:** it.infn.ct.GridEngine.JobService.JobCheckStatusService
- **Factory Class:** it.infn.ct.GridEngine.JobService.JobCheckStatusServiceFactory
- **Additional Properties:**

Edit Custom Resource

Edit an existing JNDI custom resource.

[Load Defaults](#)

JNDI Name: GridEngine-Pool

Resource Type: ☐ java.util.Properties

☒ it.infn.ct.ThreadPool.ThreadPoolExecutor

Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

Factory Class: it.infn.ct.ThreadPool.ThreadPoolExecutorFactory

Factory class for resource; implements javax.naming.spi.ObjectFactory

Description:

Status: ☒ Enabled

Additional Properties (6)		
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>

Fig. 2.6: GridEngine-Pool JNDI Resource

– jobsupdatinginterval: 900

Edit Custom Resource

Edit an existing JNDI custom resource.

[Load Defaults](#)

JNDI Name: JobCheckStatusService

Resource Type: ☐ java.util.Properties

☒ it.infn.ct.GridEngine.JobService.JobCheckStatusService

Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

Factory Class: it.infn.ct.GridEngine.JobService.JobCheckStatusServiceFactory

Factory class for resource; implements javax.naming.spi.ObjectFactory

Description:

Status: ☒ Enabled

Additional Properties (1)		
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>

Fig. 2.7: JobCheckStatusService JNDI Resource

4. Create JobServices-Dispatcher with the following parameters [4]:

- JNDI Name: JobServices-Dispatcher
- Resource Type: it.infn.ct.GridEngine.JobService.JobServicesDispatcher
- Factory Class: it.infn.ct.GridEngine.JobService.JobServicesDispatcherFactory
- Additional Properties:
 - retrycount: 3;
 - resubnumber: 10;

- **myproxyservers:** gridit=myproxy.ct.infn.it; prod.vo.eu-eela.eu=myproxy.ct.infn.it;
cometa=myproxy.ct.infn.it; eumed=myproxy.ct.infn.it; vo.eu-decide.eu=myproxy.ct.infn.it;
sagrid=myproxy.ct.infn.it; euindia=myproxy.ct.infn.it; see=myproxy.ct.infn.it;

Edit Custom Resource

Edit an existing JNDI custom resource.

[Load Defaults](#)

JNDI Name: JobServices-Dispatcher

Resource Type: ☐ java.util.Properties ☒ it.infn.ct.GridEngine.JobService.JobServicesDispatcher

Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

Factory Class: * it.infn.ct.GridEngine.JobService.JobServicesDispatcherFactory

Factory class for resource; implements javax.naming.spi.ObjectFactory

Description:

Status: ☒ Enabled

Additional Properties (2)

☒ ☐ [Add Property](#) [Delete Properties](#)

Name	Value
retrycount	3
resubnumber	10

Fig. 2.8: JobServices-Dispatcher JNDI Resource

JDBC Resources

Now you have to create the required JDBC Connection Pools. Select *Resources -> JDBC -> JDBC Connection Pools* from left panel. On the right panel you can create the resources by clicking the **New...** button.

- **Create UserTrackingPool with the following parameters:**
 - **General Settings (Step 1/2) see [5]:**
 - * **Pool Name:** UserTrackingPool
 - * **Resource Type:** select javax.sql.ConnectionPoolDataSource
 - * **Database Driver Vendor:** select MySQL
 - * Click Next
 - **Advanced Settings (Step 2/2) [6]:**
 - * **Edit the default parameters in Pool Settings using the following values:**
 - **Initial and Minimum Pool Size:** 64
 - **Maximum Pool Size:** 256
 - * **Select all default Additional properties and delete them**
 - Add the following properties:

Name	Value
Url	jdbc:mysql://sg-database:3306/userstracking
User	tracking_user
Password	usertracking

* Click Finish

Please pay attention to the Url property, ***sg-database*** should be replaced with the correct Url of your database machine. You can check if you have correctly configured the Connection Pool by clicking on Ping button, you should see the message **Ping Succeeded**, otherwise please check your configuration.

Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

[Load Defaults](#) [Flush](#) [Ping](#)

General Settings

Pool Name: UserTrackingPool

Resource Type: javax.sql.ConnectionPoolDataSource

Must be specified if the datasource class implements more than 1 of the interface.

Datasource Classname: com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource

Vendor-specific classname that implements the DataSource and/or XADataSource APIs

Driver Classname:

Vendor-specific classname that implements the java.sql.Driver interface.

Ping: ☒ Enabled

When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

Description:

Pool Settings

Initial and Minimum Pool Size: 64 Connections

Minimum and initial number of connections maintained in the pool

Maximum Pool Size: 256 Connections

Maximum number of connections that can be created to satisfy client requests

Pool Resize Quantity: 2 Connections

Number of connections to be removed when pool idle timeout expires

Idle Timeout: 300 Seconds

Maximum time that connection can remain idle in the pool

Max Wait Time: 60000 Milliseconds

Amount of time caller waits before connection timeout is sent

Transaction

Non Transactional Connections: ☒ Enabled

Returns non-transactional connections

Transaction Isolation:

If unspecified, use default level for JDBC Driver

Isolation Level: ☒ Guaranteed

All connections use same isolation level; requires Transaction Isolation

Fig. 2.9: UsersTrackingPool JDBC General settings

General	Advanced	Additional Properties								
<h3>Edit JDBC Connection Pool Properties</h3> <p>Modify properties of an existing JDBC connection pool.</p>										
<p>Additional Properties (3)</p> <p><input checked="" type="checkbox"/> <input type="checkbox"/> Add Property Delete Properties</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> Password</td> <td>usertracking</td> </tr> <tr> <td><input type="checkbox"/> User</td> <td>tracking_user</td> </tr> <tr> <td><input type="checkbox"/> Uri</td> <td>jdbc:mysql://...:3306/usertracking</td> </tr> </tbody> </table>			Name	Value	<input type="checkbox"/> Password	usertracking	<input type="checkbox"/> User	tracking_user	<input type="checkbox"/> Uri	jdbc:mysql://...:3306/usertracking
Name	Value									
<input type="checkbox"/> Password	usertracking									
<input type="checkbox"/> User	tracking_user									
<input type="checkbox"/> Uri	jdbc:mysql://...:3306/usertracking									

Fig. 2.10: UsersTrackingPool JDBC Advanced settings

Finally, you have to create the required JDBC Resources. Select *Resources* -> *JDBC* -> *JDBC Resources* from left panel. On the right panel you can create the resources by clicking the **New...** button.

- Create jdbc/UserTrackingPool with the following parameter [7]:
 - JNDI Name: jdbc/UserTrackingPool

- **Pool name:** select usertrackingPool

Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

JNDI Name: jdbc/UserTrackingPool

Pool Name:

Use the [JDBC Connection Pools](#) page to create new pools

Description:

Status: ☒ **Enabled**

Additional Properties (0)		
Add Property	Delete Properties	
	Name	Value
No items found.		

Fig. 2.11: jdbcUsersTrackingPool JDBC Resource

- Create jdbc/gehibernatepool with the following parameter [8]:
 - **JNDI Name:** jdbc/gehibernatepool
 - **Pool name:** select usertrackingPool

Finalize installation

From the left side menu, select *Applications*, find and check marketplace-portlet on the right panel and click the **Disable** button

Now, restart glassfish to finalize installation.

2.7 Installing the eTokenServer

For the testing of the Catania Science Gateway, the installation of the eTokenServer is not mandatory.

You can decide either to:

- **CONFIGURE** your portlet to point to an existing eTokenServer available in your region. The current list of eTokenServer per region is the following:

Region	Hostname	Status	Contacts
Europe	etokenserver.ct.infn.it	OK	sg-licence@ct.infn.it
Europe	etokenserver2.ct.infn.it	OK	sg-licence@ct.infn.it
Europe	sg-etoken.garr.itt	OK	sgwadmin@garr.it
Latin A.	fesc06.lemdist.unam.mx	OK	jlgr@super.unam.mx, cruz@unam.mx
Africa	etoken.grid.arn.dz	OK	o.bentaleb@grid.arn.dz
Africa	etokensrv.magrid.ma	OK	rahim@cnrst.ma
S. Africa	etoken.sagrid.ac.za	OK	vanecka@gmail.com

Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

JNDI Name: jdbc/gehibernatepool

Pool Name: ▼

Use the [JDBC Connection Pools](#) page to create new pools

Description:

Status: ☒ **Enabled**

Additional Properties (0)	
Add Property	Delete Properties
Name	Value
No items found.	

Fig. 2.12: jdbcgehibernatepool JDBC Resource

- Or, you can INSTALL your local eTokenServer and CONFIGURE your portlet to consume local robot proxy generated by your local server. The instructions for installing your eTokenServer instance is available on the [GILDA](<https://gilda.ct.infn.it/wikimain>) wiki under the Training material section. This is a user guide based on personal and other peoples experiences.

Feedback is welcome!

APPLICATION REGISTRY

3.1 About

The DB contains the applications deployed on regional Grid infrastructures available outside Europe and developed in the context of projects funded by the European Commission.

3.2 Usage

Table view

Status

Domain

Vo

Country

Copy

Print

Save

Search:

Show

10

entries

First

Previous

1

2

3

4

5

Next

Last

NAME	RUN PAGE	DOMAIN	STATUS	INSTITUTION
3D_Simulps		Earth Sciences	In production	CNRST
DRI/Mammogrid		Life Sciences	Ready for validation	CETA-CIEMAT MAAT International
Ab initio Calc		Others	Ready for gLite middleware	GRINM
AERMOD		Earth Sciences	In production	CUBAENERGIA CITMATEL
AeroVANT		Others	In production	UNRC
AFP3DFace		Computer Science and Mathematics	Ready for standalone use/running in a local cluster	CCK
Aiuri		Computer Science and Mathematics	In production	UFRJ
ALICE		High-Energy Physics	In production	CETA-CIEMAT UNAM UNINETT Sigma
AMBER		Computational Chemistry	Ready for standalone use/running in a local cluster	NECTEC
APBS		Computational Chemistry	Ready for gLite middleware	ASGC

Showing 1 to 10 of 128 entries

First

Previous

1

2

3

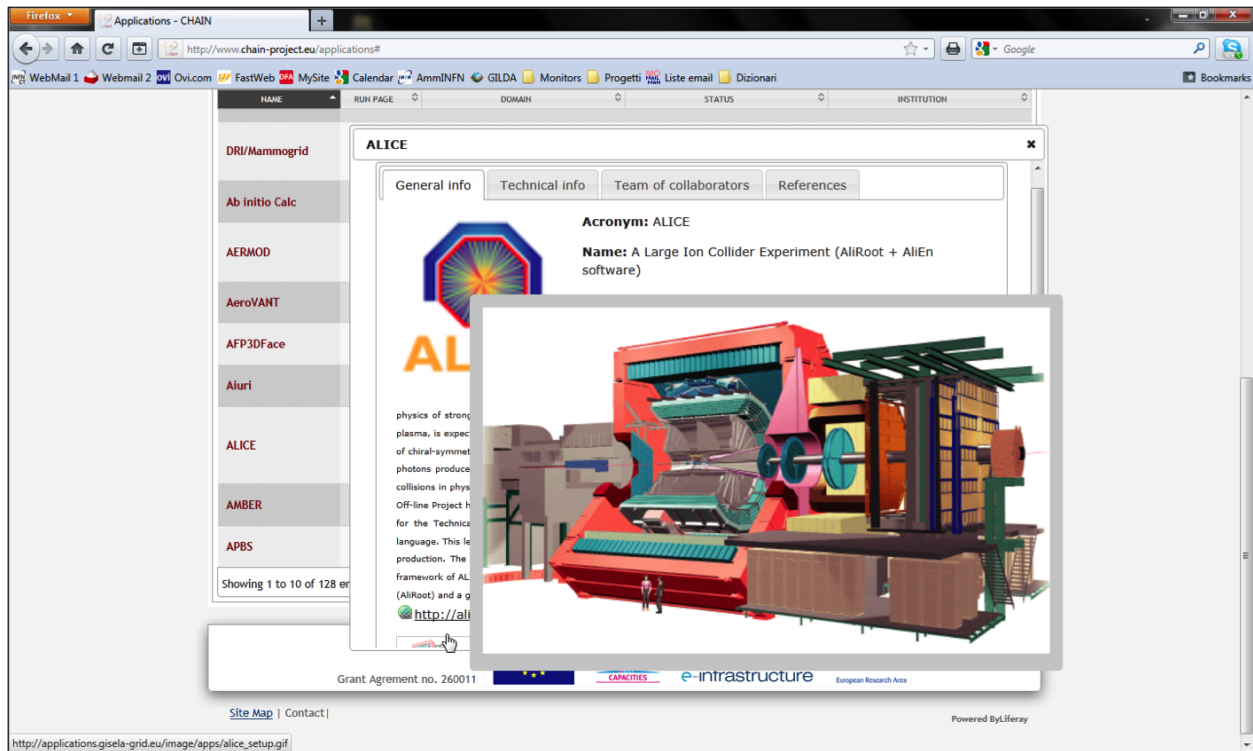
4

5

Next

Last

Application Details



Project specific Science Gateways can be accessed by the CHAIN AppDB

Status

Domain

Vo

Country

Copy

Print

Save

Search:

Show

10

entries

First

Previous

1

2

3

4

5

Next

Last

NAME	RUN PAGE	DOMAIN	STATUS	INSTITUTION
Phylogenetics		Life Sciences	Ready for gLite middleware	UPV
3D_Simulps		Earth Sciences	In production	CNRST
DRI/Mammogrid		Life Sciences	Ready for validation	CETA-CIEMAT MAAT International
Ab initio Calc		Others	Ready for gLite middleware	GRINM
AERMOD		Earth Sciences	In production	CUBAENERGIA CITMATEL
AeroVANT		Others	In production	UNRC
AFP3DFace		Computer Science and Mathematics	Ready for standalone use/running in a local cluster	CCK
Aiuri		Computer Science and Mathematics	In production	UFRJ
ALICE		High-Energy Physics	In production	CETA-CIEMAT UNAM UNINETT Sigma
AMBER		Computational Chemistry	Ready for standalone use/running in a local cluster	NECTEC

Showing 1 to 10 of 128 entries

First

Previous

1

2

3

4

5

Next

Last

3.3 Support

Please feel free to contact us any time if you have any questions or comments.

Authors [Roberto BARBERA](#) - Italian National Institute of Nuclear Physics (INFN),

[Rita RICCERI](#) - Italian National Institute of Nuclear Physics (INFN),

[Salvatore MONFORTE](#) - Italian National Institute of Nuclear Physics (INFN)

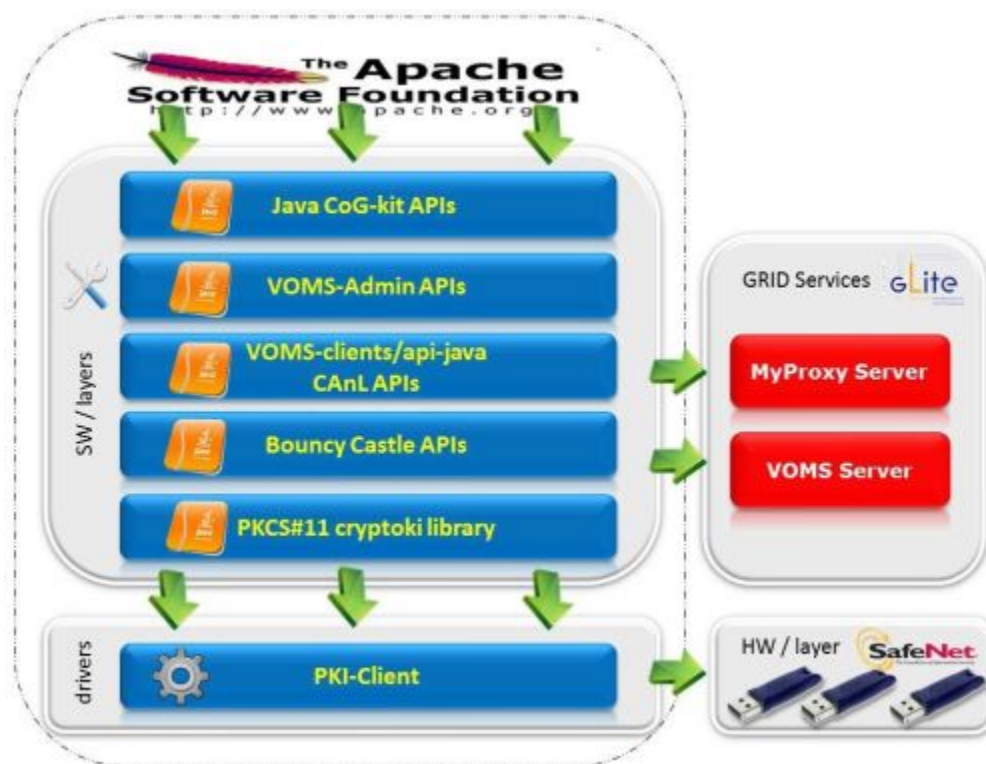
4.1 About

A standard-based solution developed by the INFN Catania for central management of robot credentials and provisioning of digital proxies to get seamless and secure access to computing e-Infrastructures supporting the X.509 standard for Authorisation.

This is a servlet based on the Java™ Cryptographic Token Interface Standard (PKCS#11). For any further information, please visit the official Java™ PKCS#11 Reference Guide [1]. By design, the servlet is compliant with the policies reported in these docs [1][2].

The business logic of the library, deployed on top of an Apache Tomcat Application Server, combines different programming native interfaces and standards.

The high-level architecture of the eToken servlet is shown in the below figure:



The business logic has been conceived to provide “resources” (e.g. complaint VOMS proxies) in a “web-manner” which can be consumed by authorized users, client applications and by portals and Science Gateways. In the current implementation, robot certificates have been safely installed on board of SafeNet [3] eToken PRO [4] 32/64 KBytes USB smart cards directly plugged to a remote server which serve, so far, six different Science Gateways.

The complete list of software, tools and APIs we have used to implement the new crypto library interface are listed below:

- Apache Application Server [5],
- JAX-RS, the Java API for RESTful Web Services (JSR 311 standard) [6],
- Java Technology Standard Edition (Java SE6) [7],
- The Cryptographic Token Interface Standard (PKCS#11) libraries [8],
- The open-source BouncyCastle Java APIs [9],
- The JGlobus-Core Java APIs [10],
- The VOMS-clients Java APIs [11],
- The VOMS-Admin Java APIs [12].

4.2 Installation

For more details about how to configure and install the servlet, please refer to the installation document.

4.3 Usage

For more details about how to work with the servlet, please refer to the installation document.

4.4 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

Authors Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),
Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics (INFN),
Salvatore MONFORTE - Italian National Institute of Nuclear Physics (INFN)

4.4.1 HOW TO INSTALL AND CONFIGURE THE ETOKEN & THE MYPROXY SERVLETS

About this document

This is the official documentation to configure and install the eTokenServer servlet (v2.0.4).

This document provides an in-depth overview of the light-weight crypto library, a standard-based solution developed by INFN Catania for central management of robot credentials and provisioning of digital proxies to get seamless and secure access to computing e-Infrastructures supporting the X.509 standard for Authorisation.

In this solution robot certificates are available 24h per day on board of USB eToken PRO [1] 32/64 KBytes smart cards having the following technical specification:

eToken PRO Smartcard Specifications	
• Operating systems	Windows 98/98SE/Me/2000/XP/NT4.0 SP6 and later/Vista; Mac OS X; Linux
• API & standards support	PKCS#11 v2.01, CAPI (Microsoft Crypto API), Siemens/Infineon APDU commands PC/SC, X.509 v3 certificate storage, SSL v3, IPsec/IKE
• Models (by memory size)	32K, 64K
• On board security algorithms	RSA 1024-bit / 2048-bit, DES, 3DES (Triple DES), SHA1
• Security certifications	Common Criteria EAL5/EAL5+ (smart card chip) / EAL4+ (smart card OS)
• ISO specification support	Support for ISO 7816 1 to 4 specifications
• Memory data retention	At least 10 years
• Memory cell rewrites	At least 500,000

We appreciate attribution. In case you would like to cite the Java light-weight crypto library in your papers, we recommend that you use the following reference:

V. Ardizzone, R. Barbera, A. Calanducci, M. Fargetta, E. Ingra', I. Porro, G. La Rocca, S. Monforte, R. Ricceri, R. Rotondo, D. Scardaci and A. Schenone **The DECIDE Science Gateway* Journal of Grid Computing (2012) 10:689-70 DOI 10.1007/s10723-012-9242-3*

We also would like to be notified about your publications that involve the use of the Java light-weight crypto libraries, as this will help us to document its usefulness. We like to feature links to these articles, with your permission, on our Web site. Additional reference to the Java light-weight crypto library and other relevant activities can be found at [2].

Licence

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Conventions used in this document

The following typographical conventions are used in this document:

Italic Indicates new terms, URLs, filenames, and file extensions

Constant width italic Shows text that should be replaced with user-specific values



This icon indicates a warning or caution.



This icon indicates that there are files to be downloaded.

Chapter I - System and Software Requirements

This chapter provides the list of requirements and the basic information that you need to know to install and configure the servlet.

#	Server	OS and Arch.	Host. Cert	Disk Space	CPU RAM
1	Physical machine with at least 2 USB ports perfectly working	SL release 5.10 (Boron) x86_64 GNU/Linux	Yes	>= 80 GB	>= 4 core GB RAM >=4 GB

Comments:

The server must be registered to the DNS with direct adn reverse resolution;

Please set a **human readable** server hostname for your server (e.g. etoken<your-domain>);

The OS installation should include the X-server since it is needed to open etProps app;

This installation has been successfully tested with eToken PRO 32/64 KBytes USB smart cards;

At least 1 USB eToken PRO 75 KBytes must be available before the installation (contact SafeNet Inc. [3] to find a neighbor reseller and get prices).

OS and repos

Start with a fresh installation of Scientific Linux 5.X (x86_64).

- ```
]# cd /etc/redhat-release
Scientific Linux release 5.10 (Boron)
```

- Configure the EGI Trust Anchor repository

```
]# cd /etc/yum.repos.d/
]# cat egi-trustanchors.repo
[EGI-trustanchors]
name=EGI-trustanchors
baseurl=http://repository.egi.eu/sw/production/cas/1/current/
gpgkey=http://repository.egi.eu/sw/production/cas/1/GPG-KEY-EUGridPMA-RPM-3
gpgcheck=1
enabled=1
```

- Install the latest EUGridPMA CA rpms

```
]# yum clean all
]# yum install -y ca-policy-egi-core
```

- Configure the EPEL repository:

```
]# cd /etc/yum.repos.d/
]# cat /etc/yum.repos.d/epel.repo
[epel]
name=Extra Packages for Enterprise Linux 5 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/5/$basearch
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=epel-5&arch=$basearch
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL

[epel-debuginfo]
name=Extra Packages for Enterprise Linux 5 - $basearch - Debug
#baseurl=http://download.fedoraproject.org/pub/epel/5/$basearch/debug
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=epel-debug-5&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux 5 - $basearch - Source
#baseurl=http://download.fedoraproject.org/pub/epel/5/SRPMS
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=epel-source-5&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL
gpgcheck=1
```

- Install the latest epel release

```
]# yum clean all
]# yum install -y epel-release --nogpgcheck
```

## SELinux configuration

Be sure that SELinux is disabled (or permissive). Details on how to disable SELinux are here [\[12\]](#)

```
]# getenforce
Disabled
```

## sendmail

Start the sendmail service at boot. Configure access rules to allow connections and open the firewall on port 25.

```
]# /etc/init.d/sendmail start
]# chkconfig --level 2345 sendmail on

]# cat /etc/hosts.allow
sendmail: localhost

]# cat /etc/sysconfig/iptables
[...]
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 25 -s 127.0.0.1 -j ACCEPT
```

## **NTP**

Use NTP to synchronize the time of the server

```
]# ntpdate ntp-1.infn.it
]# /etc/init.d/ntpd start
]# chkconfig --level 2345 ntpd on
```

## **fetch-crl**

Install and configure the fetch-crl

```
]# yum install -y fetch-crl
]# /etc/init.d/fetch-crl-cron start
]# chkconfig --level 2345 fetch-crl-cron on
```

## **Host Certificates**

Navigate the interactive map and search for your closest Certification Authorities [13] or, alternatively, buy a multi-domain COMODO [14] SSL certificate.

Public and Private keys of the host certificate have to be copied in */etc/grid-security/*

```
]# ll /etc/grid-security/host*
-rw-r--r-- 1 root root 1627 Mar 10 14:55 /etc/grid-security/hostcert.pem
-rw----- 1 root root 1680 Mar 10 14:55 /etc/grid-security/hostkey.pem
```

## **Configure VOMS Trust Anchors**

The VOMS-clients APIs need local configuration to validate the signature on Attribute Certificates issued by trusted VOMS servers.

The VOMS clients and APIs look for trust information in the */etc/grid-security/vomsdir* directory.

The *vomsdir* directory contains a directory for each trusted VO. Inside each VO two types of files can be found:

- An *LSC* file contains a description of the certificate chain of the certificate used by a VOMS server to sign VOMS attributes.
- An *X509* certificates used by the VOMS server to sign attributes.

These files are commonly named using the following pattern:

```
<hostname.lsc>
<hostname.pem>
```

where *hostname* is the host where the VOMS server is running.

When both *.lsc* and *.pem* files are present for a given VO, the *.lsc* file takes precedence. The *.lsc* file contains a list of X.509 subject strings, one on each line, encoded in OpenSSL slash-separate syntax, describing the certificate chain (up and including the CA that issued the certificate). For instance, the *voms.cnaf.infn.it* VOMS server has the following *.lsc* file:

```
/C=IT/O=INFN/OU=Host/L=CNAF/CN=voms.cnaf.infn.it
/C=IT/O=INFN/CN=INFN CA
```



Install in */etc/grid-security/vomsdir/* directory the *.lsc* for each trusted VO that you want to support.



An example of */etc/grid-security/vomsdir/* directory can be downloaded from here [15].

### Configure VOMS server endpoints

The list of known VOMS server is maintained in *vomses* files. A *vomses* file is a simple text file which contains one or more lines formatted as follows:

```
"vo_name" "hostname" "port" "dn" "aliases"
```

Where:

- *vo\_name* is the name of the VO served by the VOMS server,
- *hostname* is the hostname where the VOMS server is running,
- *port* is the port where the VOMS server is listening for incoming requests,
- *dn* is the subject of certificate of the VOMS server, and the
- *aliases* is an alias that can be used for this VOMS server (this is typically identical to the *vo\_name*).

System wide VOMSES configuration is maintained in the */etc/vomses* file or directory. If the */etc/vomses/* is a directory, all the files contained in such directory are parsed looking for VOMS contact information.



Install in the */etc/vomses* the contact information for each trust VO you want to support!



An example of VOMS contact information can be downloaded from [16]

## Chapter II - Installation & Configuration

This chapter introduces the manual installation of the SafeNet eToken PKI client library on a Linux system, the software that enables eToken USB operations and the implementation of eToken PKI-based solutions.

### Software Requirements

The software also includes all the necessary files and drivers to support the eToken management. During the installation, the needed libraries and drivers will be installed in */usr/local/bin*, */usr/local/lib* and */usr/local/etc*.



Before to start, please check if pcsc- packages are already installed on your server.

```
]# rpm -e pcsc-lite-1.4.4-4.el5_5 \
 pcsc-lite-libs-1.4.4-4.el5_5 \
 pcsc-lite-doc-1.4.4-4.el5_5 \
 pcsc-lite-devel-1.4.4-4.el5_5 \
 ccid-1.3.8-2.el5.i386 \
 ifd-egate-0.05-17.el5.i386 \
 coolkey-1.1.0-16.1.el5.i386 \
 esc-1.1.0-14.el5_9.1.i386
```



Download the correct software packages:

- pcsc-lite-1.3.3-1.el4.rf.i386.rpm [17]
- pcsc-lite-libs-1.3.3-1.el4.rf.i386.rpm [18]
- pcsc-lite-ccid-1.2.0-1.el4.rf.i386.rpm [19]

```
]# rpm -ivh pcsc-lite-1.3.3-1.el4.rf.i386.rpm \
 pcsc-lite-ccid-1.2.0-1.el4.rf.i386.rpm \
 pcsc-lite-libs-1.3.3-1.el4.rf.i386.rpm

Preparing... ##### [100%]
1:pcsc-lite-libs ##### [33%]
2:pcsc-lite-ccid ##### [67%]
3:pcsc-lite ##### [100%]
```

Before installing the eToken PKI Client, please check if the PC/SC-Lite pcscd daemon is running:

```
]# /etc/init.d/pcscd start
```

### Install PKI\_Client library



Contact the SafeNet Inc. and install the latest eToken PKI Client (ver. 4.55-34) software on your system.



```

]$ rpm -ivh pkiclient-full-4.55-34.i386.rpm

Preparing... ##### [100%]
Stopping PC/SC smart card daemon (pcscd): [OK]
 1:pkiclient-full ##### [100%]
Checking installation of pcsc from source... None.
Starting PC/SC smart card daemon (pcscd): [OK]
Adding eToken security provider...Done.
PKIClient installation completed.

```

### Configure additional libraries



Download the appropriate libraries [20] for your system and save it as *Mkproxy-rhel4.tar.gz*.

The archive contains all the requires libraries for RHEL4 and RHEL5.

```

]$ tar zxf Mkproxy-rhel4.tar.gz
]$ chown -R root.root etoken-pro/
]$ tree etoken-pro/
etoken-pro/
|-- bin
| |-- cardos-info
| |-- mkproxy
| |-- openssl
| `-- pkcs11-tool
|-- etc
| |-- hotplug.d
| | `-- usb
| | `-- etoken.hotplug
| |-- init.d
| | |-- etokend
| | `-- etsrvd
| |-- openssl.cnf
| |-- reader.conf.d
| | `-- etoken.conf
| `-- udev
| `-- rules.d
| `-- 20-etoken.rules
`-- lib
 |-- engine_pkcs11.so
 |-- libcrypto.so.0.9.8
 `-- libssl.so.0.9.8

```

Untar the archive and copy the files to their respective locations.

- Copy binary files

```

]$ cp -rp etoken-pro/bin/cardos-info /usr/local/bin/
]$ cp -rp etoken-pro/bin/mkproxy /usr/local/bin/
]$ cp -rp etoken-pro/bin/pkcs11-tool /usr/local/bin/
]$ cp -rp etoken-pro/bin/openssl /usr/local/bin/

```

- Copy libraries

```
]# cp -rp etoken-pro/lib/engine_pkcs11.so /usr/local/lib
]# cp -rp etoken-pro/lib/libssl.so.0.9.8 /usr/local/lib
]# cp -rp etoken-pro/lib/libcrypto.so.0.9.8 /usr/local/lib
```

- Copy configuration files

```
]# cp -rp etoken-pro/etc/openssl.cnf /usr/local/etc
```

- Set the PKCS11\_MOD environment variable

Edit the `/usr/local/bin/mkproxy` script and change the PKCS11\_MOD variable settings:

```
export PKCS11_MOD="/usr/lib/libeTPkcs11.so"
```

- Create symbolic links

```
]# cd /usr/lib/
]# ln -s /usr/lib/libpcsclite.so.1.0.0 libpcsclite.so
]# ln -s /usr/lib/libpcsclite.so.1.0.0 libpcsclite.so.0

]# ll libpcsclite.so*
lrwxrwxrwx 1 root root 29 Feb 17 09:47 libpcsclite.so -> /usr/lib/libpcsclite.so.1.0.0
lrwxrwxrwx 1 root root 29 Feb 17 09:52 libpcsclite.so.0 -> /usr/lib/libpcsclite.so.1.0.0
lrwxrwxrwx 1 root root 20 Feb 17 09:04 libpcsclite.so.1 -> libpcsclite.so.1.0.0
-rwxr-xr-x 1 root root 92047 Jan 26 2007 libpcsclite.so.1.0.0
```

To administer the USB eToken PRO 64KB and add a new robot certificate, please refer to the Appendix I.

- Testing

```
]# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
]# pkcs11-tool -L --module=/usr/lib/libeTPkcs11.so

Available slots:
Slot 0 AKS ifdh 00 00
 token label: **eToken**
 token manuf: Aladdin Ltd.
 token model: eToken
 token flags: rng, login required, PIN initialized, token initialized, other flags=0x200
 serial num : 001c3401
Slot 1 AKS ifdh 01 00
 token label: **eToken1**
 token manuf: Aladdin Ltd. token model: eToken
 token flags: rng, login required, PIN initialized, token initialized, other flags=0x200
 serial num : 001c0c05
[...]
```

The current version of PKI\_Client supports up to **16** different slots! Each slot can host a USB eToken PRO smart card.

- Generating a standard proxy certificate

```
]# mkproxy
Starting Aladdin eToken PRO proxy generation
Found X.509 certificate on eToken:
 label: (eTCAPI) MrBayes's GILDA ID
 id: 39453945373335312d333545442d343031612d384637302d3238463636393036363042303a30
```

```

Your identity: /C=IT/O=GILDA/OU=Robots/L=INFN Catania/CN=MrBayes
Generating a 512 bit RSA private key
.+++++
..+++++
writing new private key to 'proxykey.FM6588'

engine "pkcs11" set. Signature ok
subject=/C=IT/O=GILDA/OU=Robots/L=INFN Catania/CN=MrBayes/CN=proxy Getting CA Private Key
PKCS#11 token PIN: *****
Your proxy is valid until: Wed Jan 16 01:22:01 CET 2012

```

## Chapter III - Installing Apache Tomcat

- The instructions below are for installing version Java 7 Update 1 (7u1).

```

]# rpm -e java-1.4.2-gcj-compat-1.4.2.0-40jpp.115 \
 antlr-2.7.6-4jpp.2.x86_64 gjdoc-0.7.7-12.e15.x86_64

]# rpm -ivh jdk-7u1-linux-i586.rpm

```

- Download and extract the eTokens-2.0.5 directory with all the needed configuration files in the root's home directory.



Download an example of configuration files for the eToken from here [21] and save it as **eTokens-2.0.5.tar.gz**.

```

]# tar xzf eTokens-2.0.5.tar.gz
]# tree -L 2 eTokens-2.0.5
eTokens-2.0.5
|-- config
| |-- eToken.cfg
| |-- eToken1.cfg
| |-- ..

```

The **config** directory MUST contain a configuration file for each USB eToken PRO 32/64KB smart card plugged into the server.

```

]# cat eTokens-2.0.5/config/eToken.cfg
name = **eToken** *Insert here an unique name for the new etoken*
library = /usr/lib/libeTPkcs11.so
description = **Aladdin eToken PRO 64K 4.2B**
slot = **0** *Insert here an unique slot id for the new token*

attributes(*,CKO_PRIVATE_KEY,*) = { CKA_SIGN = true }
attributes(*,CKO_PRIVATE_KEY,CKK_DH) = { CKA_SIGN = null }
attributes(*,CKO_PRIVATE_KEY,CKK_RSA) = { CKA_DECRYPT = true }

```



If you are using USB eToken PRO 32KB, please change the description as follows:

```
description = **Aladdin eToken PRO 32K 4.2B**
```



Download the latest release of Apache Tomcat. For this wiki we used the following version: *apache-tomcat-7.0.34*

- Creating a Java Keystore from scratch containing a self-signed certificate

Make a temporary copy of *hostcert.pem* and *hostkey.pem* files

```
]# cp /etc/grid-security/hostcert.pem /root
]# cp /etc/grid-security/hostkey.pem /root
```

Convert both, the key and the certificate into DER format using openssl command:

```
]# openssl pkcs8 -topk8 -nocrypt \
 -in hostkey.pem -inform PEM \
 -out key.der -outform DER

]# openssl x509 -in hostcert.pem \
 -inform PEM \
 -out cert.der \
 -outform DER
```

- Import private and certificate into the Java Keystore



Download the following Java source code [22] and save it as *ImportKey.java*

Edit the *ImportKey.java* file containing the following settings for the Java JKS

```
// Change this if you want another password by default
String keypass = "***changeit**"; <== Change it!

// Change this if you want another alias by default
String defaultalias = "***giular.trigrid.it**"; <== Change it!

If (keystorename == null)
 Keystorename = System.getProperty("user.home")
 + System.getProperty("file.separator")
 + "***eTokenServerSSL**"; // <== Change it!
```



Please change “*giular.trigrid.it*” with the host of the server you want to configure.

- Compile and execute the Java file:

```
]# javac ImportKey.java
]# java ImportKey key.der cert.der
Using keystore-file : /root/eTokenServerSSL One certificate, no chain.
Key and certificate stored.
Alias: giular.trigrid.it Password: changeit
```

Now we have a JKS containig:

- the key and the certificate stored in the **eTokenServerSSL** file,
- using **giular.trigrid.it** as alias and
- **changeit** as password.

Move the JKS to the Apache-Tomcat root directory

```
]# mv /root/eTokenServerSSL apache-tomcat-7.0.34/eTokenServerSSL
```

- SSL Configuration

Add the new SSL connector on port 8443 in the server.xml file

```
]# cat apache-tomcat-7.0.34/conf/server.xml
[..

<Connector port="8082" protocol="HTTP/1.1" connectionTimeout="20000" redirectoPrt="8443">
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
 SSLEnabled="true"
 maxThreads="150" scheme="https" secure="true"
 clientAuth="false" sslProtocol="TLS"
 useSendfile="false"
 keystoreFile="/root/apache-tomcat-7.0.34/eTokenServerSSL"
 keyAlias="giular.trigrid.it" keystorePass="changeit"/>
[..]
```

Edit the /etc/sysconfig/iptables file in order to accept incoming connections on ports 8082 and 8443.

- How to start, stop and check the Apache Tomcat server

#### 1. Configure the JAVA\_HOME env. variable

```
]# cat ~/.bash_profile
.bash_profile

Get the aliases and functions
if [-f ~/.bashrc]; then
 . ~/.bashrc
fi

User specific environment and startup programs

PATH=$PATH:$HOME/bin
JAVA_HOME=/usr/java/latest

export PATH
export JAVA_HOME
unset USERNAME
```

#### 2. Start and check the application server as follows:

```
]# cd /root/apache-tomcat-7.0.34/
]# ./bin/startup.sh
Using CATALINA_BASE: /root/apache-tomcat-7.0.34
Using CATALINA_HOME: /root/apache-tomcat-7.0.34
Using CATALINA_TMPDIR: /root/apache-tomcat-7.0.34/temp
```

```
Using JRE_HOME: /usr
Using CLASSPATH: /root/apache-tomcat-7.0.34/bin/bootstrap.jar:\
 /root/apache-tomcat-7.0.34/bin/tomcat-juli.jar
```

### 3. Stop the application server as follows:

```
]# ./bin/shutdown
Using CATALINA_BASE: /root/apache-tomcat-7.0.34
Using CATALINA_HOME: /root/apache-tomcat-7.0.34
Using CATALINA_TMPDIR: /root/apache-tomcat-7.0.34/temp
Using JRE_HOME: /usr
Using CLASSPATH: /root/apache-tomcat-7.0.34/bin/bootstrap.jar:\
 /root/apache-tomcat-7.0.34/bin/tomcat-juli.jar
```

- Install external libraries



Download and save the external libraries [23] as lib.tar.gz

```
]# tar xzf lib.tar.gz
]# cp ./lib/*.jar /root/apache-tomcat-7.0.34/lib
```

- Deploy the WAR files

```
]# cd /root/apache-tomcat-7.0.34/
```

Create the following **\*\*eToken.properties\*\*** configuration file with the following settings:

```
VOMS Settings
Standard location of configuration files
VOMSES_PATH=/etc/vomses
VOMS_PATH=/etc/grid-security/vomsdir
X509_CERT_DIR=/etc/grid-security/certificates
Default VOMS proxy lifetime (default 12h)
VOMS_LIFETIME=24

Token Settings
ETOKEN_SERVER=giular.trigrid.it # <== Change here
ETOKEN_PORT=8082
ETOKEN_CONFIG_PATH=/root/eTokens-2.0.5/config
PIN=***** # <== Add PIN here

Proxy Settings
Default proxy lifetime (default 12h) PROXY_LIFETIME=24
Number of bits in key {512|1024|2048|4096}
PROXY_KEYBIT=1024

Administrative Settings
SMTP_HOST=smtp.gmail.com # <== Change here
SENDER_EMAIL=credentials-admin@ct.infn.it # <== Change here
DEFAULT_EMAIL=credentials-admin@ct.infn.it # <== Change here
EXPIRATION=7
```

Create the following **\*\*Myproxy.properties\*\*** configuration file with the following settings:

```
MyProxy Settings
MYPROXY_SERVER=myproxy.cnaf.infn.it # <== Change here
MYPROXY_PORT=7512
Default MyProxy proxy lifetime (default 1 week)
MYPROXY_LIFETIME=604800
Default temp long-term proxy path
MYPROXY_PATH=/root/apache-tomcat-7.0.53/temp # <== Change here
```



Download the servlet for the eTokenServer [24] and save it as eTokenServer.war



Download the servlet for the MyProxyServer [25] and save it as MyProxyServer.war

```
]# cp eTokenServer.war webapps/
]# cp MyProxyServer.war webapps/
]# ./bin/catalina.sh stop && sleep 5

]# cp -f eToken.properties webapps/eTokenServer/WEB-INF/classes/infn/eToken/
]# cp -f MyProxy.properties webapps/MyProxyServer/WEB-INF/classes/infn/MyProxy/

]# ./bin/catalina.sh start
]# tail -f logs/eToken.out
]# tail -f logs/MyProxy.out
```

- Configure tomcat to start-up on boot

Create the following script:

```
]# cat /etc/init.d/tomcat
#!/bin/bash
chkconfig: 2345 91 91
description: Start up the Tomcat servlet engine.

. /etc/init.d/functions
RETVAL=$?
CATALINA_HOME="/root/apache-tomcat-7.0.34"

case "$1" in
 start)
 if [-f $CATALINA_HOME/bin/startup.sh];
 then
 echo $"Starting Tomcat"
 /bin/su root $CATALINA_HOME/bin/startup.sh

 fi
 ;;
 stop)
 if [-f $CATALINA_HOME/bin/shutdown.sh];
 then
 echo $"Stopping Tomcat"
 /bin/su root $CATALINA_HOME/bin/shutdown.sh

 fi
 ;;
 *)

```

```

 echo $"Usage: $0 {start|stop}"
 exit 1
 ;;
esac
exit $RETVAL

]# chmod a+x tomcat

```

- Update the run level for the tomcat service

```

]# chkconfig --level 2345 --add tomcat
]# chkconfig --list tomcat
tomcat 0:off 1:off 2:on 3:on 4:on 5:on 6:off

```

## Chapter IV - Usage

In this chapter is show the administrator (only restricted access) web interface to interact with the RESTful “ligh-weight” crypto library which is configured for:

1. browsing the digital certificates available on the different smart cards;
  2. generating VOMS-proxy for a given X.509 digital certificate.
- Accessing the RESTful crypto library via WEB


The root resource of the library is deployed at the URL [https://<etoken\\_server>:8443/eTokenServer](https://<etoken_server>:8443/eTokenServer) as shown in the figure below:

### Create your proxy request in few steps!

A standard-based solution developed by the INFN Catania for central management of robot credentials and provisioning of digital proxies to get seamless and secure access to computing e-Infrastructures supporting the X.509 standard for Authorisation.



This is a servlet based on the Java™ Cryptographic Token Interface Standard (PKCS#11). For any further information, please visit the official Java™ PKCS#11 Reference Guide [\[1\]](#)

-  **EGI-Engage project**  
The eToken servlet will be officially used by the EGI-Engage project (EINFRA-1). The mission is to accelerate the implementation of the Open Science Commons within the European Grid Infrastructure (EGI)



1
Choose the certificate

Select one certificate from the drop-down list below

Select one certificate from the drop-down list below

/C=IT/O=INFN/OU=Robot/L=Catania/CN=Robot: Catania Science Gateway - Roberto Barbera  
/C=IT/O=INFN/OU=Robot/L=Catania/CN=Robot: Gisela Science Gateway - Roberto Barbera  
/C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Giuseppe La Rocca  
/C=IT/O=INFN/OU=Robot/L=GARR/CN=Robot: DECIDE Science Gateway - Fulvio Galeazzi  
/C=IT/O=INFN/OU=Robot/L=COMETA/CN=Robot: EarthServer Science Gateway - Roberto Barbera  
/C=IT/O=INFN/OU=Robot/L=Catania/CN=Robot: EUMEDGRID Science Gateway - Roberto Barbera  
/C=IT/O=INFN/OU=Robot/L=Catania/CN=Robot: VESPA Science Gateway - Roberto Barbera  
/C=IT/O=INFN/OU=Robot/L=Catania/CN=Robot: DCHRP eCulture Science Gateway - Roberto Barbera  
/C=IT/O=INFN/OU=Robot/L=Catania/CN=Robot: agINFRA Science Gateway - Roberto Barbera

5
Get your request

6
About the servlet

Copyright © 2010 - 2015. All rights reserved



The creation of a request to access the generic USB smat card and generates a proxy certificate is performed in few steps.

- First and foremost we have to select a valid digital certificate from the list of available certificates (first according).
- Afterwards, depending by the selected certificate, it will be possible to select a list of FQANs attributes which will be taken into account during the proxy creation process.

2 Add AC attributes

? Select AC attributes to add to the generated proxy.

/vo.eu-decide.eu/GridANN4ND/Role=DataManager
/vo.eu-decide.eu/GridANN4ND/Role=Physician
/vo.eu-decide.eu/GridANN4ND/Role=Scientist
/vo.eu-decide.eu/GridEEG/Role=DataManager
/vo.eu-decide.eu/GridEEG/Role=Physician
/vo.eu-decide.eu/GridEEG/Role=Scientist
/vo.eu-decide.eu/GridGDI/Role=DataManager
/vo.eu-decide.eu/GridGDI/Role=Physician
/vo.eu-decide.eu/GridGDI/Role=Scientist
/vo.eu-decide.eu/GridMRISeg/Role=DataManager
/vo.eu-decide.eu/GridMRISeg/Role=Physician
/vo.eu-decide.eu/GridMRISeg/Role=Scientist
/vo.eu-decide.eu/GridSPM/Role=DataManager
/vo.eu-decide.eu/GridSPM/Role=Physician
/vo.eu-decide.eu/GridSPM/Role=Scientist
/vo.eu-decide.eu/Role=Neurologist
/vo.eu-decide.eu
/vo.eu-decide.eu/GridANN4ND
/vo.eu-decide.eu/GridEEG
/vo.eu-decide.eu/GridGDI
/vo.eu-decide.eu/GridMRISeg
/vo.eu-decide.eu/GridSPM
/vo.progettovespa.it

- If necessary FQANs order can be changed in step 3:

3 AC attributes order

? Choose the FQANs order that best fits your requirements

/vo.eu-decide.eu
/vo.eu-decide.eu/Role=Neurologist

- Before to complete, some additional options can be specified in the 4th. step to customize the proxy requestID:

4 Choose options

Use the options below as you need

Enable Proxy Renewal: ☒ Disable VOMS Proxy: ☐ Create RFC Proxy: ☒

Add some additional info to account users of robot proxy certificates

Enable CN: ☒

- At the end, the complete requestID is available in step 5:

5 Get your request

Here is your request

```
https://etokenserver3.ct.infn.it:8443/eTokenServer/eToken/b970fe11cf219e9c6644da0bc4845010?voms=vo.eu-decide.eu:/vo.eu-decide.eu+vo.eu-decide.eu:/vo.eu-decide.eu/Role=Neurologist&proxy-renewal=true&disable-voms-proxy=false&rfc-proxy=true&cn-label=eToken:LAROCCA
```

## Chapter V - Some RESTful APIs

REST is an architectural style which defines a set of constraints that, when applied to the architecture of a distributed system, induces desirable properties like loose coupling and horizontal scalability. RESTful web services are the result of applying these constraints to services that utilize web standards such as URIs, HTTP, XML, and JSON. Such services become part of the fabric of the web and can take advantage of years of web engineering to satisfy their clients' needs. The Java API for RESTful web services (JAX-RS) is a new API that aims to make development of RESTful web services in Java simple and intuitive.

In this chapter will be presented some examples of RESTful APIs used to request proxies certificates, list available robot certificates in the server-side and register long-term proxies on the MyProxy server.

### Create RFC 3820 complaint proxy (simple use case):

```
https://<etoken_server>:8443/eTokenServer/eToken/332576f78a4fe70a52048043e90cd11f?\
voms=fedcloud.egi.eu:/fedcloud.egi.eu\
proxy-renewal=true\
disable-voms-proxy=false\
rfc-proxy=true&cn-label=Empty
```

### Create RFC 3820 complaint proxy (with some additional info to account real users):

```
https://<etoken_server>:8443/eTokenServer/eToken/332576f78a4fe70a52048043e90cd11f?\
voms=fedcloud.egi.eu:/fedcloud.egi.eu\
proxy-renewal=true\
disable-voms-proxy=false\
rfc-proxy=true\
cn-label=LAROCCA
```

**Create full-legacy Globus proxy (old fashioned proxy):**

```
https://<etoken_server>:8443/eTokenServer/eToken/43ddf806454eb55ea32f729c33cc1f07?\
voms=eumed:/eumed&\
proxy-renewal=true&\
disable-voms-proxy=false&\
rfc-proxy=false&\
cn-label=Empty
```

**Create full-legacy proxy (with more FQANs):**

```
https://<etoken_server>:8443/eTokenServer/eToken/b970fe11cf219e9c6644da0bc4845010?\
voms=vo.eu-decide.eu:/vo.eu-decide.eu/Role=Neurologist+vo.eu-decide.eu:/vo.eu-decide.eu&\
proxy-renewal=true&\
disable-voms-proxy=false&\
rfc-proxy=false&\
cn-label=Empty
```

**Create plain proxy (without VOMS ACs):**

```
https://<etoken_server>:8443/eTokenServer/eToken/332576f78a4fe70a52048043e90cd11f?\
voms=gridit:/gridit&\
proxy-renewal=true&\
disable-voms-proxy=true&\
rfc-proxy=false&\
cn-label=Empty
```

**Get the list of available robot certificates in the server (in JSON format):**

```
https://<etoken_server>:8443/eTokenServer/eToken?format=json
```

**Get the MyProxy settings used by the eToken server (in JSON format):**

```
https://<etoken_server>:8443/MyProxyServer/proxy?format=json
```

**Register long-term proxy on the MyProxy server (only for expert user):**

```
https://<etoken_server>:8443/MyProxyServer/proxy/x509up_6380887419908824.long
```

**Appendix I - Administration of the eToken smart cards**

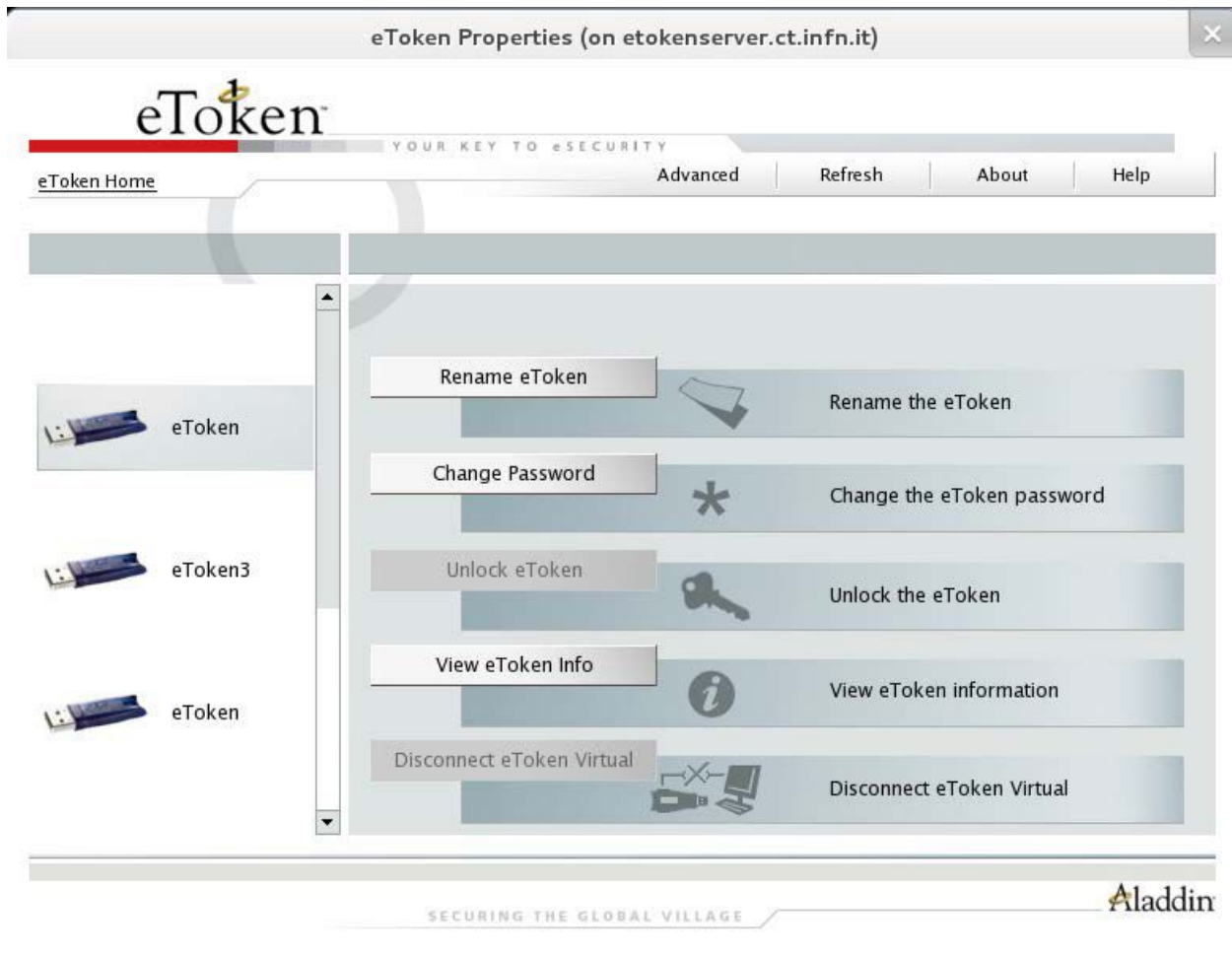
This appendix provides a brief explanation of the eToken Properties (*etProps*) and the various configuration options available to the user.

*eToken Properties* provides users with a configuration tool to perform basic token management such as password changes, viewing information, and viewing of certificates on the eToken.

This appendix includes the following sections:

- Initializing the eToken PRO 32/64 KBytes USB smart card;
- Importing new certificates;
- Renaming a token.

The *eToken Properties* application displays all the available tokens connected to the server as show in the below figure:



In the right pane, the user may select any of the following actions which are enabled:

- 1.) **Rename eToken** - set a label for the given token;
- 2.) **Change Password** - changes the eToken user password;
- 3.) **Unlock eToken** - resets the user password via a challenge response mechanism (pnly enabled when an administrator password has been initialized on the eToken);
- 4.) **View eToken Info** - provides detailed information about the eToken;
- 5.) **Disconnect eToken Virtual** - disconnects the eToken Virtual with an option for deleting it.

The toolbar along the top contains these functions:

- 1.) **Advanced** - switches to the Advanced view;

- 2.) **Refresh** - refreshes the data for all connected tokens;
- 3.) **About** - displays information about the product version;
- 4.) **Help** - launches the online help.

- **Renaming the eToken**

The token name may be personalized. To rename a token:

- 1.) In the left pane of the *eToken Properties* window, select the token to be renamed.
- 2.) Click **Rename eToken** in the right pane, and the Rename eToken dialog box is displayed as shown in the below figure:



- 3.) **Enter** the new name in the New eToken name field.
- 4.) Click **OK**. The new token name is displayed in the *eToken Properties* window.

- **Initializing the eToken**

The eToken initialization option restores an eToken to its initial state. It removes all objects stored on the eToken since manufacture, frees up memory, and resets the eToken password, allowing administrators to initialize the eToken according to specific organizational requirements or security modes.

The following data is initialized:

- eToken name;
- User password;
- Administrator password;
- Maximum number of login failures (for user and administrator passwords);
- Requirement to change the password on the first login;
- Initialization key.

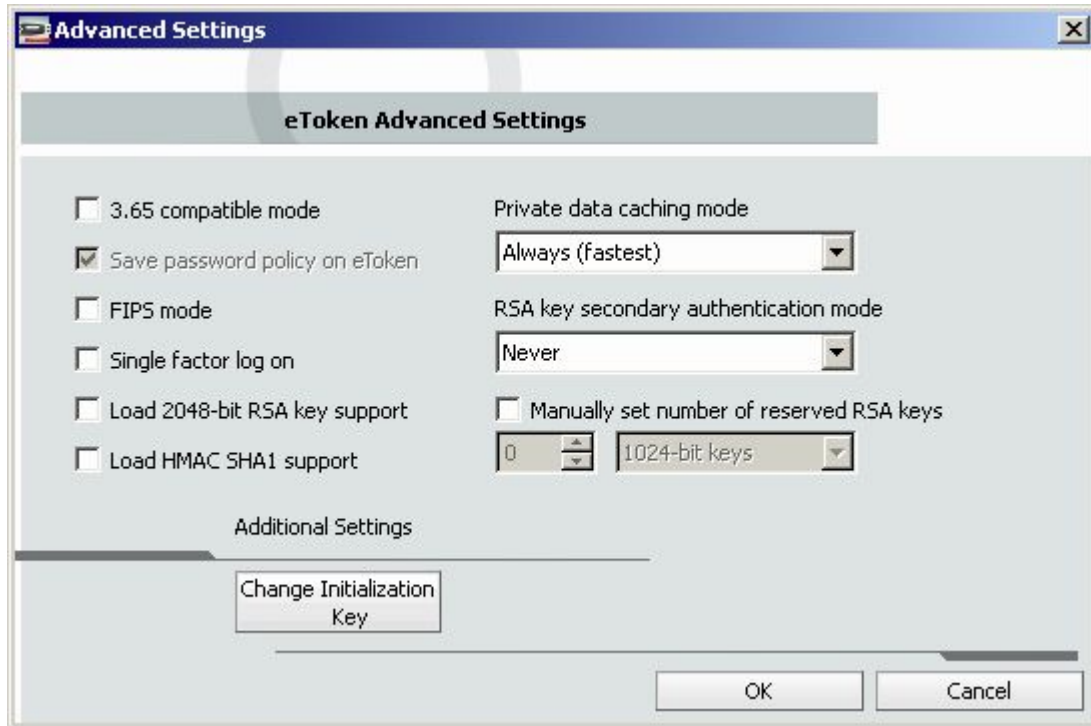
To initialize the eToken:

- 1.) Click on **Advanced** from the toolbar to switch to the Advanced view.
- 2.) **Select** the eToken you want to initialize.
- 3.) Click **Initialize eToken** on the toolbar, or right-click the token name in the left pane and select Initialize eToken from the shortcut menu. The eToken Initialization Parameters dialog box opens.

- 4.) Enter a name for the eToken in the eToken Name field. If no name is entered, the default name, “eToken”, is applied.
- 5.) Select **Create User Password** to initialize the token with an eToken user password. Otherwise, the token is initialized without an eToken password, and it will not be usable for eToken applications.
- 6.) If **Create User Password** is selected, enter a new eToken user password in the Create User Password and Confirm fields.
- 7.) In the Set maximum number of logon failures fields, enter a value between 1 and 15. This counter specifies the number of times the user or administrator can attempt to log on to the eToken with an incorrect password before the eToken is locked. The default setting for the maximum number of incorrect logon attempts is 15.
- 8.) To configure advanced settings, click **Advanced**. The eToken Advanced Settings dialog box opens.
- 9.) **Check Load 2048-bit RSA key support**



All eTokens are configured with the following default password **1234567890**.



- **To import a certificate**

- 1.) Click on **Advanced** from the toolbar to switch to the Advanced view.
- 2.) **Select** the eToken where you want to upload a new certificate.
- 3.) Click **Import Certificate** on the toolbar, or right-click the token name in the left pane and select **Import Certificate** from the shortcut menu. The Import Certificate dialog box opens.



- 4.) Select whether the certificate to import is on your personal computer store on the computer, or on a file. If you select the personal certificate store, a list of available certificates is displayed. Only certificates that can be imported on to the eToken are listed. These are:
  - Certificates with a private key already on the eToken;
  - Certificates that may be imported from the computer together with its private key.
- 5.) If you select Import a certificate from a file, the Choose a certificate dialog box opens.
- 6.) Select the certificate to import and click **Open**.

- 7.) If the certificate requires a password, a Password dialog box opens.
- 8.) Enter the certificate password. A dialog box opens asking if you want to store the CA certificate on the eToken.
- 9.) Select **No**. The only certificate is imported and a confirmation message is shown.

## Appendix II - Increase “Open Files Limit”



If you are getting the error “*Too many open files (24)*” then your application is hitting max open file limit allowed by Linux.

Check limits of the running process:

- Find the process-ID (PID):

```
]# ps aux | grep -i process-name
```

- Suppose XXX is the PID, then run the command to check limits:

```
]# cat /proc/XXX/limits
```

To increase the limit you have to:

1. Append the following settings to set the user-limit

```
]# cat /etc/security/limits.conf

* hard nofile 50000
* soft nofile 50000
root hard nofile 50000
root soft nofile 50000
```

Once you have saved the file, you have to logout and login again.

2. Set the higher than user-limit set above.

```
]# cat /etc/sysctl.conf

fs.file-max = 2097152
```

Run the command

```
]# sysctl -p
```

3. Verify the new limits. Use the following command to see max limit of the file descriptors:

```
]# cat /proc/sys/fs/file-max
```



## Appendix III - Configure GlassFish settings

To set JVM settings, please add the following GLASSFISH\_OPTS settings in *catalian.sh*

```
CATALINA_OPTS="$CATALINA_OPTS -Xmx2336m -Xms2336m \
-XX:NewSize=467215m -XX:MaxNewSize=467215m \
-XX:PermSize=467215m -XX:MaxPerSize=467215m \
-server"
```

## Troubleshooting

- Private key in PKCS#8

*Cannot load end entity credentials from certificate file: /etc/grid-security/hostcert.pem and key file: /etc/grid-security/hostkey.pem*

```
]# cd /etc/grid-security/
]# mv hostkey.pem hostkey-pk8.pem
]# openssl rsa -in hostkey-pk8.pem -out hostkey.pem
]# chmod 400 hostkey.pem

]# cd <apache-tomcat>
]# ./bin/catalina.sh stop
]# ./bin/catalina.sh start
```

For further information, please read the document [26]

## Log Files

- The log messages for the eTokenServer are stored in *<apache-tomcat>/logs/eToken.out*
- The log messages for the MyProxyServer are stored in *<apache-tomcat>/logs/MyProxy.out*
- In case of errors and debug, please check these additional log files:

```
]# <apache-tomcat>/logs/catalina.out
]# <apache-tomcat>/logs/localhost.<date>.log
```

## Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),  
 Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics (INFN),  
 Salvatore MONFORTE - Italian National Institute of Nuclear Physics (INFN)



---

## FEDERATED LOGIN

---

### 5.1 About

The *Federated Login portlet* is an extension plugin of [Liferay application framework](#) which introduce additional authentication schema to the many already bundled.

Currently, two new schema are supported

- SAML Federation
- [STORK Federation](#) (based on [STORK 1 project](#))

Other protocols will be added in the future

### 5.2 Installation

#### 5.2.1 Requirements

The plugin work only for Liferay 6.1. It is an ext so after the installation the source code of liferay is modified and it cannot be reverted to the original so before to install create a backup of your current installation.

#### SAML Federation

To perform authentication using SAML, Liferay has to be executed behind apache (or similar service) configured to perform SAML authentication. The attributes has to be provided to the application server in order for the module to read them. A common scenario is to use apache with [mod\\_shibboleth](#), which is already available in many linux distribution. Apache will communicate with the application server using [mod\\_proxy\\_ajp](#) or other proxies.

#### STORK federation

The STORK module includes the [opensaml-java libraries](#). In order to make the library working, please, you have to make available in your application context the endorsed libraries used by opensaml. These are Xerces and Xalan. For the installation in your application server you may refer to the official guideline provided for opensaml. Alternatively a copy of the libraries are included in the package and they are available in the path `./federated-login-ext/WEB-INF/ext-lib/portal/endorsed/` inside the autodeploy directory.

### 5.2.2 Deployment

In order to deploy the plugin you need to compile it first. The development and compilation environment is based on [liferay plugins sdk version 6.1](#). Download and configure Liferay plugin following its documentation and when ready clone or download the source code in the `ext` directory. The command `ant war` will build the war in the directory `dist`. It is also deploy the portlet from the plugin sdk using the command `ant deploy`.

The Federated Login portlet can be deployed in a Liferay instance as any other portlet copying the war file in the deploy directory or uploading the file using the marketplace portlet of Liferay and accessing the tab to deploy from war file.

***NOTE:** the plugin will create a `portal-ext.properties`, if already present this will be overwritten so if you have some options configured please take note and apply again after the installation*

### 5.2.3 Configuration

#### SAML

No configuration files need to be modified to use SAML federated login. However, be sure the server performing the SAML communication (i.e. Apache httpd) is properly configured.

#### STORK

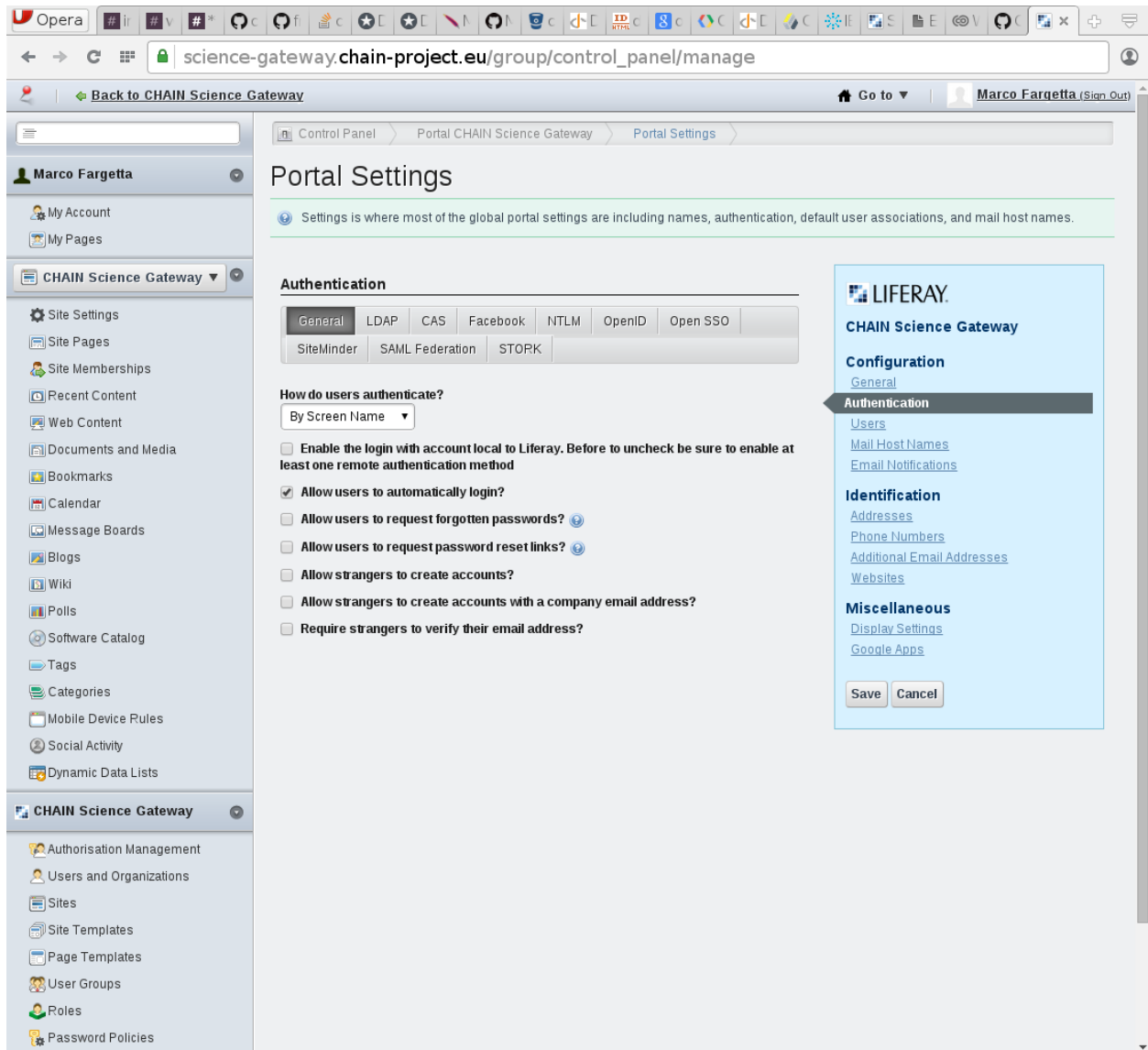
After the installation of the war file you need to stop the server and locate the file `SamLEngineSignConfig.xml` inside the `WEB-INF` directory of liferay application. This file contains the information about the certificate to use with STORK servers. Open the file and edit the path and other information to refer to a java keystone containing the certificate and key of the server and the trusted public keys of remote servers.

For more information on how to prepare the keystone look at STORK documentation.

### 5.2.4 Post Installation Configuration

After the installation and configuration of the plugin several new element should be visible in the authentication section of portal settings in Liferay configuration panel.

In the **General** tab there is the new option to disable the local login as shown in the following figure:



When the option “*Enable the login with account local to Liferay. Before .....*” is unchecked the login form will not show the username and password fields and the users have to use an alternative method to use. Other two tabs are available.

## SAML

The *SAML federation* tab contains the configuration to use for SAML and it has the option shown in the following figure:

CHAIN Science Gateway

Authentication

General LDAP CAS Facebook NTLM OpenID Open SSO SiteMinder **SAML Federation** STORK

☒ Enabled

**Authentication Parameters**

Screen Name Email Address  
mail

First Name Last Name  
givenName sn

Middle Name Job Title

Group UUID

Password

**Service Parameters**

SAML attribute to identify the user  
Email Address

☐ Create Account

☒ Check LDAP Account

Filter to identify the user in the LDAP  
mail=@email\_address@

Page to redirect users if they are not registered and autoregistration disabled  
/not\_authorised

Page to redirect users if saml attributes are missed to complete the authentication  
/attributes\_missed

Page behind external SAML authentication  
/saml/login

Page to exit from external SAML authentication (SAML logout)  
/Shibboleth.sso/Logout

**LIFERAY**

CHAIN Science Gateway

**Configuration**

General

**Authentication**

Users

Mail Host Names

Email Notifications

**Identification**

Addresses

Phone Numbers

Additional Email Addresses

Websites

**Miscellaneous**

Display Settings

Google Apps

Save Cancel

The first block of parameters, *Authentication Parameters*, allows to define the mapping among SAML and Liferay attributes. In each field user has to specify the SAML attribute to use for it. They are not mandatory but the one used to identify the user is needed.

After, the *Service Parameters* allows to customise the behaviour of the authentication. The first combo box, *SAML attribute to identify the user*, specify how the user has to be identified inside Liferay. *Create account* check-box when enabled will add a new user when SAML attribute does not identify any existing user (this feature is not implemented in this version of the plugin). The next check-box, *Check LDAP Account*, will search the user inside the LDAP, if account are managed in a LDAP server, using the query string provided in the following field. This is useful because Liferay copy locally only a subset of attributes available so could be useful to search in the target set. As an example, Liferay allows only a single mail value per user where LDAP Account may have multiple values. The filter has to be an LDAP compliant filter and can include several SAML attributes, from the page help:

Enter the search filter that will be used to identify the SAML a user. The tokens @company\_id@, @email\_address@, @screen\_name@, @last\_name@ and @first\_name@ are replaced at runtime with the correct values. If the IdP provides multiple values for some attributes the search will be performed several times, corresponding to the number of values obtained

Finally, there are four pages to specify:

- The page where redirect users not registered in the local DB (or LDAP if enabled). This page should be customised with a form to request access of useful information on how to be included.
- The page where redirect users in case the Identity Provider is not providing mandatory attributes used to identify the user.
- The page to protect with SAML protocol. This page could be modified but the same value has to be registered in the file `liferay-web.xml` in your deployment.
- The page where users can perform global/local SAML logout after the liferay session is destroyed.

## STORK

The STORK configuration has many similarities with SAML configuration above so the description will focus on the differences. The configuration tab has the options shown in the following figure:

The screenshot shows the STORK configuration page in the Liferay Science Gateway control panel. The page is titled "science-gateway.chain-project.eu/group/control\_panel/manage". It features a sidebar with navigation links like "Site Settings", "Site Pages", "Site Memberships", "Recent Content", "Web Content", "Documents and Media", "Bookmarks", "Calendar", "Message Boards", "Blogs", "Wiki", "Polls", "Software Catalog", "Tags", "Categories", "Mobile Device Rules", "Social Activity", and "Dynamic Data Lists". The main content area is divided into "Authentication" and "Service Parameters" sections.

The "Authentication" section includes a "General" tab with "Enabled" checked. Below it, the "Authentication Parameters" section maps Liferay attributes to STORK attributes:

- Screen Name: givenName
- Email Address: surname
- First Name: givenName
- Last Name: surname
- Middle Name: givenName
- Job Title: surname
- Group: givenName
- UUID: surname
- Password: givenName

The "Service Parameters" section includes the following fields:

- SP QAA Level: 3
- Provider Name: CHAIN-REDS
- Provider Identity: sector: RESEARCH
- Provider Identity: application: CHAIN-REDS Science Gateway
- Environment (TESTING/PRODUCTION): Production
- Country: Italy
- PEPS URL: https://it-peps-stork.polito.it/PEPS/ServiceProvider
- European Map: checked
- SAML attribute to identify the user: Email Address
- Create Account: unchecked
- Check LDAP Account: checked
- Filter to identify the user in the LDAP: mail=@email\_address@
- Page to redirect users if they are not registered and autoregistration disabled: /not\_authorised
- Page to redirect users if saml attributes are missed to complete the authentication: /attributes\_missed

On the right side, there is a "LIFERAY CHAIN Science Gateway" sidebar with links for "Configuration", "Authentication", "Users", "Mail Host Names", "Email Notifications", "Identification", "Addresses", "Phone Numbers", "Additional Email Addresses", "Websites", "Miscellaneous", "Display Settings", and "Google Apps". At the bottom of this sidebar are "Save" and "Cancel" buttons.

The first block of parameters, *Authentication Parameters*, defines the mapping among Liferay and STORK attributes.

The second block of parameter allows to define the behaviour of STORK. Differently from SAML, there is not need of an external authentication but everything is managed in Liferay. Therefore, it is requested to specify the parameter for

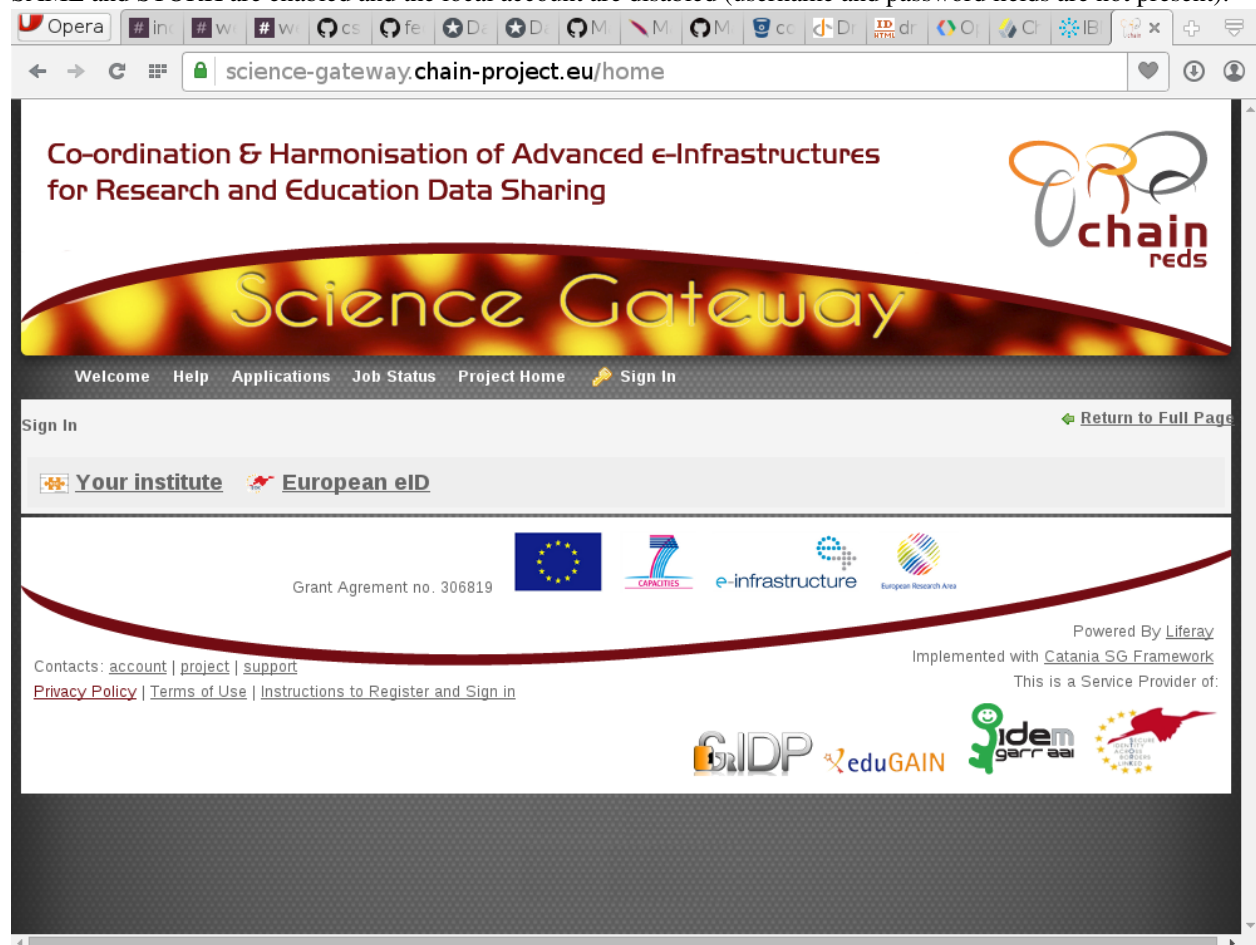
the communication with the STORK PEPS service. All these information should be provided from the organisation managing the PEPS. The check-box *European Map* has a graphical impact because it specifies if the origin country of the user has to be selected from a map or from a combo-box.

After the STORK specific configuration there are the options to customise how the user is identified and if the search is on the local DB or should follow in the LDAP, using the same filter defined for SAML. Like for the SAML implementation the option to create new account in case users are not identified is not implemented in this version.

Finally, the pages for the not identified user and for the authentication missing mandatory attributes. In this case there is no need to specify the protected page and the logout page because they are totally managed by Liferay.

## 5.3 Usage

After the installation and configuration, if everything is working properly, a user accessing the portal and trying to sign-in should see the Liferay login portlet with the new protocols, when enabled, shown with an icon and a link, such as when *OpenID* or *Facebook* authentication are enabled. The following figure shows the login portlet where both SAML and STORK are enabled and the local account are disabled (username and password fields are not present):



## 5.4 Contributors

- Marco Fargetta



- Roberto Barbera

### 5.4.1 Contribution

Many features and protocols are not supported and will be developed if and when needed. If you would contribute to this project with code, documentation, testing or other please fork the [project in GitHub](#) and send pull request with your changes.



---

## GLIBRARY

---

Contents:

### 6.1 GLIBRARY 1.0

#### 6.1.1 REST API v 1.0

gLibrary provides two endpoints depending on the type of operation:

- **Data Management (DM)**: operations that affect the underlying storage backend (Grid DPM servers or Swift Object Storage)
- **Metadata Management (MM)**: operations that affect the underlying Metadata and DB Services (AMGA and PostgreSQL currently)

DM endpoint:

```
https://glibrary.ct.infn.it/dm/
```

MM endpoint:

```
https://glibrary.ct.infn.it:4000/
```

Internally they have been deployed as two separate services, using two different technology, respectively, Flask, a Python microframework, and Node.js.

#### Authentication

At the moment, APIs can be accessed directly with a X.509 certificates from any HTTPs client (eg: command line scripts with `_curl_` or `_wget_`, Web or Desktop HTTP clients) or indirectly through a Science Gateway.

For **X.509 authentication**, currently INFN released Robot certificated are allowed, and a given set of users that requested access. If you need to have access to our APIs with a personal X.509 certificate, please contact us at [sg-license@ct.infn.it](mailto:sg-license@ct.infn.it).

To access gLibrary API from web applications or Science Gateway portels, we have implemented **server-to-server authentication**. We mantien a white list of server IP addresses that are allowed to access glibrary endpoints. To avoid CORS problems, your server should implement a proxy mechanism that blindly redirect API requests to our endpoints. Again, contact us at [sg-license@ct.infn.it](mailto:sg-license@ct.infn.it), to request access for your server.

## Data Management

### File Download

Download a file from a Storage Element to the local machine

```
GET https://glibrary.ct.infn.it/dm/<vo>/<se>/<path:path> HTTP/1.1
```

#### Parameters

Parameter	Description
vo	Virtual Organisation (eg: vo.aginfra.eu)
se	Storage Element (eg: prod-se-03.ct.infn.it)
path	absolute path of the file to be downloaded

#### Response

Short lived URL to download the file over http. Example:

```
$ curl -L -O https://glibrary.ct.infn.it/dm/vo.aginfra.eu/prod-se-03.ct.infn.it/dpm/ct.infn.it/home/v
```

### File Upload

Upload a local file to a given Storage Element of a Virtual Organization. The upload of a file requires two steps: the first one prepares the destination storage to receive the upload and returns a short-lived URL to be used by a second API request for the actual upload (second step).

```
GET https://glibrary.ct.infn.it/dm/put/<vo>/<filename>/<se>/<path:path> HTTP/1.1
```

```
PUT http://<storage_host>/<storage_path> HTTP/1.1
```

#### Parameters

Parameter	Description
vo	Virtual Organisation (vo.aginfra.eu)
se	Storage Element where upload the file (eg: prod-se-03.ct.infn.it)
filename	Name that will be used to store the file on the storage. Can be different by the original name
path	Absolute path where the file will be located on the storage

#### Response

Redirect short-live URL, authorized only for the requesting IP, where the actual file should be uploaded Status 307 (temporary redirect)

Example:

step-1:

```
$ curl http://glibrary.ct.infn.it/dm/put/vo.aginfra.eu/file-test.txt/prod-se-03.ct.infn.it/dpm/ct.in
```

Output:

```
{
 "redirect": "http://prod-se-03.ct.infn.it/storage/vo.aginfra.eu/2014-04-30/file-test.txt.53441.0",
 "status": 307
}
```

Example

step-2:

```
$ curl -T file-test.txt -X PUT "http://prod-se-03.ct.infn.it/storage/vo.aginfra.eu/2014-04-30/file-test.txt.53441.0" -H "Content-Type: text/plain"
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>201 Created</title>
</head><body>
<h1>Created</h1>
<p>Resource /storage/vo.aginfra.eu/2014-04-30/file-test.txt.53441.0 has been created.</p>
<hr />
<address>Apache/2.2.15 (Scientific Linux) Server at prod-se-03.ct.infn.it Port 80</address>
</body></html>
```

## File Download (Swift Object Storage)

```
GET https://glibrary.ct.infn.it/api/dm/cloud/<host>/<path> HTTP/1.1
```

### Parameters

Parameter	Description
host	Swift Object-storage front-end (or proxy)
path	Object full path, following the Swift format: /v1/<account>/<container>/<object>

Example:

```
$ curl https://glibrary.ct.infn.it/api/dm/cloud/stack-server-01.ct.infn.it/v1/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/grids
```

Returns:

```
{
 "url": "http://stack-server-01.ct.infn.it:8080/v1/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/grids"
}
```

The returned URL, that allows the direct download of the requested file from the containing server, has an expiration of 10 seconds.

## File Upload (Swift Object Storage)

```
PUT https://glibrary.ct.infn.it/api/dm/cloud/<host>/<path> HTTP/1.1
```

**Parameters**

Parameter	Description
host	Swift Object-storage front-end (or proxy)
path	Object full path, following the Swift format: /v1/<account>/<container>/<object>

Example:

```
$ curl -X PUT https://glibrary.ct.infn.it/api/dm/cloud/stack-server-01.ct.infn.it/v1/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/grids
```

Returns:

```
{
 "url": "http://stack-server-01.ct.infn.it:8080/v1/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/grids"
}
```

Again it returns a temporary URL valid 10 seconds to complete the upload directly to the storage with:

```
$ curl -X PUT -T prova.xml "http://stack-server-01.ct.infn.it:8080/v1/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/grids/prova.xml"
```

**File system namespace management**

These APIs expose a subset of WebDAV functionalities over eInfrastructure Storage Elements. They allow operations such as directory creation (MKCOL), file metadata retrieval (PROPFIND), file renaming (MOVE), file deleting (DELETE).

```
PROPFIND https://glibrary.ct.infn.it/dm/dav/<vo>/<se>/<path:path>
DELETE https://glibrary.ct.infn.it/dm/dav/<vo>/<se>/<path:path>
MOVE https://glibrary.ct.infn.it/dm/dav/<vo>/<se>/<path:path>
MKCOL https://glibrary.ct.infn.it/dm/dav/<vo>/<se>/<path:path>
```

**Parameters**

Parameter	Description
vo	Virtual Organisation (vo.aginfra.eu)
se	Storage Element where the file is located (eg: prod-se-03.ct.infn.it)
path	Absolute path where the file is located on the storage

**Directory Creation**

Example:

```
$ curl -X MKCOL http://glibrary.ct.infn.it/dm/dav/vo.aginfra.eu/prod-se-03.ct.infn.it/dpm/ct.infn.it/test2/
```

Output:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>201 Created</title>
</head><body>
<h1>Created</h1>
<p>Collection /dpm/ct.infn.it/home/vo.aginfra.eu/test2/ has been created.</p>
<hr />
```

```
<address>Apache/2.2.15 (Scientific Linux) Server at prod-se-03.ct.infn.it Port 443</address>
</body></html>
```

## File metadata retrieval

Example:

```
$ curl -X PROPFIND -H "Depth:1" http://glibrary.ct.infn.it/dm/dav/vo.aginfra.eu/prod-se-03.ct.infn.it
```

Output

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
 <D:response xmlns:lcgdm="LCGDM:" xmlns:lp3="LCGDM:" xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/prop/urn:ietf:params:xml:ns:carddav" xmlns:lp4="http://apache.org/dav/prop/urn:ietf:params:xml:ns:calendar" />
 <D:href>/dm/dav/vo.ag-infra.eu/prod-se-03.ct.infn.it/dpm/ct.infn.it/home/vo.aginfra.eu/test2/</D:href>
 <D:propstat>
 <D:prop>
 <lcgdm:type>0</lcgdm:type><lp1:resourcetype><D:collection/></lp1:resourcetype>
 <lp1:creationdate>2014-04-30T15:25:31Z</lp1:creationdate><lp1:getlastmodified>Wed, 30 Apr 2014 15:25:31Z</lp1:getlastmodified>
 <D:status>HTTP/1.1 200 OK</D:status>
 </D:propstat>
 </D:response>
</D:multistatus>
```

## File deletion

```
$ curl -X DELETE http://glibrary.ct.infn.it/dm/dav/vo.dch-rp.eu/prod-se-03.ct.infn.it/dpm/ct.infn.it
```

## Repository Management

### List of the available repositories

Returns the list of the available repositories

```
GET https://glibrary.ct.infn.it:3000/repositories HTTP/1.1
```

Example

```
$ curl https://glibrary.ct.infn.it:3000/repositories
```

Output:

```
{
 "result": [
 "/gLibTest",
 "/deroberto",
 "/gLibIndex",
 "/tmp",
 "/deroberto2",
```

```
 "/medrepo",
 "/ESArepo",
 "/EELA",
 "/EGEE",
 "/testRepo",
 "/ChinaRep",
 "/templaterepo",
 "/myTestRepo",
 "/ICCU",
 "/aginfra"
 "... "
]
}
```

## Repository Creation

Description Create a new repository

```
POST https://glibrary.ct.infn.it:3000/repositories/<repo> HTTP/1.1
```

Returns:

```
{
 "success": "true"
}
```

### Parameters

Parameter	Description
repo	Repository name

Example:

```
$ curl -X POST http://glibrary.ct.infn.it:3000/repositories/agInfra
```

## Retrieve repository information

Provides the list of types (model) of a given repository. A type describes the kind of digital objects using a schema (set of attributes).

```
GET https://glibrary.ct.infn.it:3000/repositories/<repo> HTTP/1.1
```

Returns an array of all the types available in the given repository. Each object represents a supported type, with some properties:

### Parameters

Parameter	Description
repo	Repository name

### Response



Property	Description
TypeName	a label that describes the type (to be shown in the gLibrary browser Interface)
Path	the absolute path of the entries in the underlying metadata server (AMGA)
VisibleAttrs	the set of attributes visible through the gLibrary browser (both Web and mobile)
FilterAttrs	a set of attributes that can be used to filter the entries (digital objects) of the given type
ColumnWidth	size of each column (attribute) in the gLibrary browser
ParentID	types can be organized in a hierarchical structure (tree), and a type can have a subtype. The root type has id 0
Type	a unique identifier assigned to a given type to refer to it in other API call

Example:

```
$ curl http://glibrary.ct.infn.it:3000/repositories/agInfra
```

Output

```
{
 "results": [
 {
 "TypeName": "Soil Maps",
 "Path": "/agInfra/Entries/SoilMaps",
 "VisibleAttrs": "Thumb title creator subject description type format language date",
 "FilterAttrs": "creator subject publisher contributor type format language rights",
 "ColumnWidth": "80 120 60 60 230 100 100 80 80",
 "ParentID": "0",
 "id": "1",
 "Type": "SoilMaps"
 }
]
}
```

### Add a type to a repository

Add a new Type to a given repository.

```
POST https://glibrary.ct.infn.it:3000/<repo> HTTP/1.1
```

#### URI Parameters

Parameter	Description
repo	The name of the repository to which we are adding the type to

#### Body Parameters

Parameter	Description
__Type	the unique identifier (string) to be assigned to the type
__VisibleAttrs	the set of attributes visible through the gLibrary browser (both Web and mobile)
__ColumnWidth	size of each column (attribute) in the gLibrary browser
__ParentID	types can be organized in a hierarchical structure (tree), and a type can have a subtype. The root type has id 0
{AttributeName}	a set of attributes with their data type (allowed data types are varchar, int, float, timestamp, boolean)

Example:

```
$ curl -X POST -d "__Type=Documents&__VisibleAttrs='Topic,Meeting,FileFormat,Size,Creator,Version'&__"
```

## Retrieve Type information

Returns the information about a given type of a given repository.

```
GET https://glibrary.ct.infn.it:3000/<repo>/<type> HTTP/1.1
```

Returns A JSON object with the information of a given type with a list of all its attributes and given data type

Example:

```
$ curl http://glibrary.ct.infn.it:3000/aginfra/SoilMaps
```

Output:

```
{
 TypeName: "Soil Maps",
 Path: "/aginfra/Entries/SoilMaps",
 VisibleAttrs: "Thumb title creator subject description type format language date",
 FilterAttrs: "creator subject publisher contributor type format language rights",
 ColumnWidth: "80 120 60 60 230 100 100 80 80",
 ParentID: "0",
 id: "1",
 Type: "SoilMaps",
 FileName: "varchar(255)",
 SubmissionDate: "timestamp",
 Description: "varchar",
 Keywords: "varchar",
 LastModificationDate: "timestamp",
 Size: "int",
 FileType: "varchar(10)",
 Thumb: "int",
 ThumbURL: "varchar",
 TypeID: "int",
 title: "varchar",
 creator: "varchar",
 subject: "varchar",
 description: "varchar",
 publisher: "varchar",
 contributor: "varchar",
 type: "varchar",
 format: "varchar",
 identifier: "varchar",
 source: "varchar",
 language: "varchar",
 date: "varchar",
 relation: "varchar",
 coverage: "varchar",
 rights: "varchar"
}
```

## List of all the entries of a given type

List all the entries and its metadata of a given Type in a repository (default limit to 100)

```
GET https://glibrary.ct.infn.it:3000/<repo>/<type>/entries HTTP/1.1
```

### Parameters

Parameter	Description
repo	The name of the repository
type	The name of type

Example:

```
$ curl http://glibrary.ct.infn.it:3000/aginfra/SoilMaps/entries
```

Output:

```
{
 results:
 [
 {
 id: "51",
 FileName: "",
 SubmissionDate: "2012-11-09 07:02:00",
 Description: "",
 Keywords: "",
 LastModificationDate: "",
 Size: "",
 FileType: "",
 Thumb: "1",
 ThumbURL: "",
 TypeID: "1",
 title: "CNCP 3.0 software",
 creator: "Giovanni Trapatoni",
 subject: "software|soil management",
 description: "CNCP 3.0 database with italian manual. CNCP is the program used",
 publisher: "E doardo A. C. Costantini",
 contributor: "Giovanni L'Abate",
 type: "application",
 format: "EXE",
 identifier: "http://abp.entecra.it/soilmaps/download/sw-CNCP30.exe",
 source: "http://abp.entecra.it/soilmaps/en/downloads.html",
 language: "it",
 date: "2011-08-03",
 relation: "",
 coverage: "world",
 rights: "All rights reserved"
 },
 {
 id: "53",
 FileName: "",
 SubmissionDate: "2012-11-09 09:37:00",
 Description: "",
 Keywords: "",
 LastModificationDate: "",
 Size: "",
```

```

 FileType: "",
 Thumb: "1",
 ThumbURL: "",
 TypeID: "1",
 title: "Benchmark at Beccanello dome, Sarteano (SI)",
 creator: "Edoardo A. C. Costantini",
 subject: "soil analysis|soil map|pedology",
 description: "Form: Soil profile, Survey: Costanza Calzolari, Reporter: Calzolari",
 publisher: "CRA-ABP Research centre for agrobiolgy and pedology, Florence, Italy",
 contributor: "Centro Nazionale di Cartografia Pedologica",
 type: "Soil map",
 format: "KML",
 identifier: "https://maps.google.com/maps/ms?ie=UTF8&hl=it&msa=0&msid=1151389",
 source: "https://maps.google.com/maps/ms?ie=UTF8&hl=it&authuser=0&msa=0&output=embed",
 language: "en",
 date: "2010-09-22",
 relation: "",
 coverage: "Italy",
 rights: "info@soilmaps.it"
 }
}

```

### Retrieve the metadata of a given entry

Retrieve all the metadata (and replica info) the a given entry

```
GET https://glibrary.ct.infn.it:3000/<repo>/<type>/id HTTP/1.1
```

Returns The metadata of the given entry and the replicas of the associated digital objects

#### Parameters

Parameter	Description
repo	The name of the repository
type	The name of type
id	The id of the entry to inspect

Example:

```
$ curl http://glibrary.ct.infn.it:3000/aginfra/SoilMaps/56
```

Output:

```

{
 results: {
 id: "56",
 FileName: "",
 SubmissionDate: "2012-11-09 10:03:00",
 Description: "",
 Keywords: "",
 LastModificationDate: "",
 Size: "",
 FileType: "",
 Thumb: "1",
 }
}

```

```

ThumbURL: "",
TypeID: "1",
title: "ITALIAN SOIL INFORMATION SYSTEM 1.1 (ISIS)",
creator: "Costantini E.A.C.|L'Abate G.",
subject: "soil maps|pedology",
description: "The WebGIS and Cloud Computing enabled ISIS service is running for onl
publisher: "Consiglio per la Ricerca e la sperimentazione in Agricoltura (CRA)-(ABP)
contributor: "INFN, Division of Catania|agINFRA Science Gateway|",
type: "",
format: "CSW",
identifier: "http://aginfra-sg.ct.infn.it/isis",
source: "http://aginfra-sg.ct.infn.it/webgis/cncp/public/",
language: "en",
date: "2012-04-01",
relation: "Barbetti R. Fantappi M., L Abate G., Magini S., Costantini E.A.C. (2010).
coverage: "Italy",
rights: "giovanni.labate@entecra.it",
"Replicas": [
{
"url": "https://unipa-se-01.pa.pi2s2.it/dpm/pa.pi2s2.it/home/vo.aginfra.eu/aginfra/
"enabled": "1"
},
{
"url": "https://inaf-se-01.ct.pi2s2.it/dpm/ct.pi2s2.it/home/vo.aginfra.eu/aginfra/m
"enabled": "1"
},
{
"url": "https://unict-dmi-se-01.ct.pi2s2.it/dpm/ct.pi2s2.it/home/vo.aginfra.eu/agin
"enabled": "0"
}
]
}

```

### Add a new entry

Add a new entry with its metadata of a given type

```
POST https://glibrary.ct.infn.it:3000/<repo>/<type>/ HTTP/1.1
```

#### Parameters

Parameter	Description
repo	The name of the repository
type	The if of the type

#### Body Parameters

Example:

```
$ curl -X POST -d "__Replicas=https://prod-se-03.ct.infn.it/dpm/ct.infn.it/home/vo.aginfra.eu/test/m
```

### Delete an entry

Delete an entry from a repository of the given type

```
DELETE https://glibrary.ct.infn.it:3000/<repo>/<type>/id HTTP/1.1
```

**Parameters**

Parameter	Description
repo	The name of the repository
type	The name of type
id	Id of the entry to be deleted

## 6.2 gLibrary 2.0

### 6.2.1 Table of contents

- *Overview*
- *Authentication*
  - *Login*
  - *User creation*
- *Authorization*
  - *ACLs*
- *Repositories*
  - *List repositories*
  - *Create a repository*
- *Collections*
  - *Create a collection*
  - *Import a collection*
  - *List all the collections*
  - *Collection's schema*
  - *Delete a collection*
- *Items*
  - *Creation*
  - *Listing*
  - *Detail*
  - *Deletion*
  - *Update*
- *Replicas*
  - *Creation*
  - *List*
  - *Download*
  - *Upload*

- *Delete*
- *Relations*
  - *Retrieve related items*
- *Contacts*

## 6.2.2 Overview

gLibrary is a service that offers both access to existing data repositories and creation of new ones via a simple REST API.

A **repository** in the gLibrary lingo is a virtual container of one or more data **collection**.

A **collection** provides access to a relational DB table or to a non-relational (NoSQL) DB collection. Currently gLibrary supports MySQL, PostgreSQL, Oracle and MongoDB.

Each repository can group together one or more collections, providing a virtual and uniform interface to data tables coming from different databases that could be potentially of different types (for example one collection provides access to a PostgreSQL table and another to a MongoDB collection). JSON is used as the input and output data format.

Once collections are imported or created from scratch, the gLibrary RESTA APIs can be used to retrieve, create, update and delete collection's records, that in gLibrary lingo are called **items**. Moreover a powerful filtering system is available to make queries on collections. All the criteria are specified using the query string of the API GET call. (ex `/v2/repos/fantasy_company/orders?filter[where][userId]=acaland&filter[where][orderQuantity]=100` will search for 100 orders issued by the user acaland with a quantity of 100)

Each item can have one or more *attachment*, that we call **replica**. Replicas can be stored on Grid Storage Elements (Disk Pool Manager) or Cloud Storage (OpenStack Swift is supported).

**Relations** between two collections of the same repository can be created, if foreign keys are properly assigned. Currently we support one-to-many relations.

### Beta server endpoint

```
http://glibrary.ct.infn.it:3500
```

## 6.2.3 Authentication

Before to send any request, users should be authenticated. Currently authentication is based on username/password couple. This will return a **session token id** that needs to be used with any following request. There are two options to send the **access\_token**:

- via a query parameter:

```
curl -X GET http://glibrary.ct.infn.it:3500/v2/repos?access_token=6Nb2ti5QEXIoDBS5FQGWIz4poRFiBCMMYJbYXSGHWuulOuy0GTEuGx2VCEVvbpBK
```

- via HTTP headers:

```
ACCESS_TOKEN=6Nb2ti5QEXIoDBS5FQGWIz4poRFiBCMMYJbYXSGHWuulOuy0GTEuGx2VCEVvbpBK

curl -X GET -H "Authorization: $ACCESS_TOKEN" \
http://glibrary.ct.infn.it:3500/v2/repos
```

## Login

To obtain a session id, you need to pass a valid `username` and `password` to the following endpoint:

```
POST /v2/users/login HTTP/1.1
```

```
{
 "username": "admin",
 "password": "opensesame2015"
}
```

Alternatively you can use the `email` address instead of the `username`.

## User creation

New users are created issuing requests to the following endpoint:

```
POST /v2/users HTTP/1.1
```

The mandatory parameters are:

- **username**
- **email**
- **password**

Please notice that the created user, has no access to any repository yet. The admin user need to assign the created user to any repository and/or collections, setting properly the ACLs.

## 6.2.4 Authorization

Currently gLibrary allows to set separate permissions to repositories, collections and items per each user. The default permission set to a newly created user is *NO ACCESS* to anything. It's admin's responsibility to set properly the ACLs per each user. Currently an instance of gLibrary server has just one superadmin (the *admin* user), but in future releases you will have the option to define admins per repository.

## ACLs

To set ACLs, the super admin can issue requests to two separate endpoints:

```
POST /v2/repos/<repo_name>/_acls http/1.1
```

and/or

```
POST /v2/repos/<repo_name>/<collection_name>/_acls http/1.1
```

The body of each requests has the following attributes:

attribute	description
<i>username</i>	the username of the user to which we are adding permissions to
<i>permissions</i>	valid options are "R" and "RW"
<i>items_permissions</i>	(for collections only) valid options are "R" and "RW"



*permissions* refers to repository or collection permission, according to where the request is issued:

- Repository:
  - “R” grants a user the capability of listing its content (ie. list of collections)
  - “RW” grants a user the capability of creating (or importing) new collections or deleting them
- Collection:
  - “R” grants a user the capabilities to list the collection’s content (list of items)
  - “RW” grants a user the capabilities of creating, updating, deleting the collection’s items

*items\_permissions* is valid only for collections’s ACL and refers to:

- “R” grants a user the capability to download items’replicas
- “RW” grants a user the capality to create, update and upload replicas

## 6.2.5 Repositories

A gLibrary server can host one or more **repositories**. A repository should be created before creating new **collections** or importing existing db tables or NoSQL collections as gLibrary collections.

A repository has a name, a path, that represents the access point in the API path, and optionally a `coll_db` (*TODO*: rename as `default_collection_db`). If a default DB is defined at the moment of the creation, this will be the default backend DB for all the collections created or imported of the given repository. However, this can be overridden per each collection, if new DB info is provided when the collection is created

### List of all the repositories hosted on the server

```
GET /v2/repos/ HTTP/1.1
```

Returns a list of all the repositories managed by the given gLibrary server. Each repository has the following properties:

name	description
name	Repository name
path	Direct endpoint of the given repository
collection_db	Default database where collection data should be stored. Can be overridden per collection
host	FQDN of the default collection DB
port	port number of the default collection DB
username	username of the default collection DB
password	password of the default collection DB
database	name of the database to use for the default collection DB
type	type of the default collection db (mysql, postgresql, mongodb)

Example:

```
{
 "name": "infn",
 "path": "http://glibrary.ct.infn.it:5000/v2/infn",
 "coll_db": {
 "host": "giular.trigrid.it",
 "port": 3306,
 "username": "root",
 "password": "*****",
 "database": "test",
 }
}
```

```

 "type": "mysql"
 }
}

```

Each repository can have a `collection_db` where collections data will be stored. If no `collection_db` is specified, the repository will use the local non-relational mongoDB that comes with gLibrary. Each repository's collection can override the `collection_db`.

## Create a new repository

```
POST /v2/repos/ HTTP/1.1
```

Create a new repository. A default `collection_db` can be specified. It will store all the collections in case no `collection\_db` parameter is specified during collection creation. This property is optional. If missing it will use the local MongoDB server.

### Parameters

name	type	description
name	string	Name of the repository (will be the API path)
collection_db	object	(Optional) Default database where collection data should be stored. Can be overridden per collection
host	string	FQDN of the default collection DB
port	number	port number of the default collection DB
username	string	username of the default collection DB
password	string	password of the default collection DB
database	string	name of the database to use for the default collection DB
type	string	type of the default collection db (mysql, postgresql, mongodb)
default_storage	object	(Optional) specifies the default storage for replicas
baseURL	string	it's full path of Swift Container or Grid SURL for replica uploads
type	string	"swift" or "grid" storage

Note: name is a lowercase string. Numbers and underscores are allowed. No oyrjt special characters are permitted

Example:

```
POST /v2/repos/ HTTP/1.1
Content-Type: application/json
```

```

{
 "name": "infn",
 "collection_db": {
 "host": "glibrary.ct.infn.it",
 "port": 5432,
 "username": "infn_admin",
 "password": "*****",
 "database": "infn_db",
 "type": "postgresql"
 },
 "default_storage": {
 "baseURL": "http://stack-server-01.ct.infn.it:8080/v2/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/q",
 "type": "swift"
 }
}

```

```
}
}
```

Be sure to set `Content-Type` to `application/json` in the *Request Headers*.

## 6.2.6 Collections

Each repository contains one or more collections. Collections are abstractions over relational database tables or non-relational database “collections”, exposing their records over REST APIs and JSON format. The available APIs allow the repository administrator to create new collection, specifying a schema in the case of relational collection, or importing existing tables/NoSQL collections. If not specified, collections will be created/imported from the default `coll_db` (*TODO*: `default_collection_db`) of the containing repository. Otherwise, each collection can retrieve data from local or remote database, overriding the default repository value, using the `coll_db` (*TODO*: `collection_db`) property.

### Create a new collection

```
POST /v2/repos/<repo_name>/ HTTP/1.1
```

#### Parameters

name	type	description
name	string	Name of collection
schema	object	(Optional for non relational DB) define the schema of the new collection
collection_db	string	(Optional) Default database where collection data should be stored. Can be overridden per collection
host	string	FQDN of the default collection DB
port	number	port number of the default collection DB
username	string	username of the default collection DB
password	string	password of the default collection DB
database	string	name of the database to use for the default collection DB
type	string	type of the default collection db (mysql, postgresql, mongodb)

Schema is a JSON object listing the the name of the attributes and their types in case we want a non-relational collection. Each property represents the name of an attribute and the value is another object with the following keys:

name	description
type	type of the attribute's value. Example of allowed types are: string, number, 'boolean', 'date'
required	whether a value for the property is required
default	default value for the property
id	whether the property is a unique identifier. Default is false

For a full list of the supported type, please refer to <https://docs.strongloop.com/display/public/LB/LoopBack+types> and <https://docs.strongloop.com/display/public/LB/Model+definition+JSON+file#ModeldefinitionJSONfile-Generalpropertyproperties>.

**Example** (creation of a new collection on a relational db):

```
POST /v2/repos/infn/ HTTP/1.1
Content-Type: application/json

{
 "name": "articles",
```

```
"schema": {
 "title": {"type": "string", "required": true},
 "year": "integer",
 "authors": "array"
}
```

The previous request will create a collection named `articles` into the `infn` repository. The collection data will be stored into the default `coll_db` specified for the `infn` repository (that according to the previous example is a PostgreSQL db named `infn_db`)

### Import data from an existing relational database

If you want to create a collection that maps an existing db table, two additional properties are available:

name	description
import	it should set to <code>true</code>
tablename	name of the database table of the database to be imported

**Example** (creation of a new collection with data coming from an existing relational db):

```
POST /v2/repos/infn/ HTTP/1.1
Content-Type: application/json

{
 "name": "old_articles",
 "import": "true",
 "tablename": "pubs",
 "collection_db": {
 "host": "somehost.ct.infn.it",
 "port": 3306,
 "username": "dbadmin",
 "password": "*****",
 "database": "test_daily",
 "type": "mysql"
 }
}
```

The previous request will create the collection `old_articles` import data from an existing database, named `test_daily` and providing access to its table named `pubs`.

### List all the collections of a repository

```
GET /v2/repos/<repo_name>/ HTTP/1.1
```

This API will return a JSON array with all the collections of `<repo_name>`. Each collection will have a `schema` attribute, describing the schema of the underlying DB table. If the `schema` attribute is `null` it means the collection has been imported and it inherits the schema of the underlying DB table. An additional API is available to retrieve the schema of a given collection (see [next API](#)).

#### Example

```
GET /v2/repos/sports HTTP/1.1
```

```
[
 {
 "id": "560a60987ddae89366556d2",
 "name": "football",
 "path": "/sports/football",
 "location": "football",
 "coll_db": null,
 "import": "false",
 "schema": null
 },
 {
 "id": "560a60987ddae89366556d3",
 "name": "windsurf",
 "path": "/sports/windsurf",
 "location": "windsurf",
 "coll_db": null,
 "import": "false",
 "schema": {
 "rider": {
 "type": "string",
 "required": true
 },
 "nationality": {
 "type": "string",
 "required": false
 },
 "teamid": {
 "type": "number",
 "required": false
 }
 }
 }
]
```

The `sports` repository has two collections `football` and `windsurf`. The first one is stored on the default `coll_db` repository DB and it's schema-less, while the second one has a predefined schema.

### Retrieve the schema of a collection

```
GET /v2/repos/<repo_name>/<collection_name>/_schema HTTP/1.1
```

If the given `collection_name` is hosted in a relation database table, this API will return a JSON object with the schema of the underlying table.

#### Example

```
GET /v2/repos/comics/dylandog/_schema HTTP/1.1
```

```
{
 "id": {
 "required": true,
 "length": null,
 "precision": 10,
 "scale": 0,
```

```
 "id": 1,
 "mysql": {
 "columnName": "id",
 "dataType": "int",
 "dataLength": null,
 "dataPrecision": 10,
 "dataScale": 0,
 "nullable": "N"
 }
 },
 "fragebogenId": {
 "required": true,
 "length": null,
 "precision": 10,
 "scale": 0,
 "mysql": {
 "columnName": "fragebogen_id",
 "dataType": "int",
 "dataLength": null,
 "dataPrecision": 10,
 "dataScale": 0,
 "nullable": "N"
 }
 },
 "nummer": {
 "required": true,
 "length": 256,
 "precision": null,
 "scale": null,
 "mysql": {
 "columnName": "nummer",
 "dataType": "varchar",
 "dataLength": 256,
 "dataPrecision": null,
 "dataScale": null,
 "nullable": "N"
 }
 }
}
```

## Delete a collection

```
DELETE /v2/repos/<repo_name>/<collection_name> HTTP/1.1
```

This API will delete the given `collection_name` from `repo_name`. Actual data on the backend table should not be deleted. It's a sort of *unlinking*, so that the db table/nosql collection will not be accessible anymore from the gLibrary REST API.

## 6.2.7 Items (previously entries)

**Items** represents the content of a given collection. If a collection is hosted in a relational database, each item is a table record, while if it's non relational it's the document/object of the NoSQL collection. Items can be listed and queried via the filtering system, created/added, updated and deleted, using the REST APIs provided by gLibrary.

## Item creation

```
POST /v2/repos/<repo_name>/<collection_name> HTTP/1.1
```

This API add a new item into the given `collection_name`. Item content have to be provided as a JSON object. In case of the relational collection it should conform to the collection schema. In the case of attributes that have no corresponding column table, their values will be ignored silently. If the API will be successfull a new record or document will be added to the underlying table or NoSQL collection.

### Example

```
POST /v2/repos/infno/articles HTTP/1.1

{
 "title": "e-Infrastructures for Cultural Heritage Applications",
 "year": 2010,
 "authors": ["A. Calanducci", "G. Foti", "R. Barbera"]
}
```

## Item listing

```
GET /v2/repos/<repo_name>/<collection_name>/ HTTP/1.1
```

Retrieve the items inside the `collection_name` as a JSON array of objects. Each object is a record of the underlying table (in case of relational DB) or document (in case of NoSQL collection). By default the first 50 items are returned. See below the description of filtering system in the [query section](#) to change this behaviour.

### Example

```
GET /v2/repos/gridcore/tracciati HTTP/1.1
```

## Item detail

```
GET /v2/repos/<repo_name>/<collection_name>/<item_id> HTTP/1.1
```

Retrieve the detail of an item with a `given_id`. It will return a JSON object with the attributes mapping the schema of the given `collection_name`.

### Example

```
GET /v2/repos/infno/articles/22
```

## Item deletion

```
DELETE /v2/repos/<repo_name>/<collection_name>/<item_id> HTTP/1.1
```

Delete the given `item_id` of the the collection `collection_name`. Delete will be successfull only if the given item has no replica. You can force the deletion of item with replicas setting:

```
{
 "force": true
}
```

in the request body.

### Item update

```
PUT /v2/repos/<repo_name>/<collection_name>/<item_id> HTTP/1.1
```

Update one of more attributes of the given `item_id`. The request body has to contain a JSON object with the attribute-value pair to be updated with the new values.

### Queries with filters

```
GET /v2/repos/<repo_name>/<collection_name>?filter[<filterType>]=<spec>&filter[...]=<spec>... HTTP/1
```

where `filterType` is one of the following:

- `where`
- `include`
- `order`
- `limit`
- `skip`
- `fields`

and `spec` is the specification of the used filter.

Additional info on the full query syntax can be found [here](#)

### Example

## 6.2.8 Replicas

Each item can have one or more attachments, generally the same file stored in different locations, such as Cloud storage servers (Swift based) or Grid Storage Elements (DPM based). So we call them also replicas.

### Replica creation

```
POST /v2/repos/<repo_name>/<collection_name>/<item_id>/_replicas/ HTTP/1.1
```

name	description
uri	(optional) provides the full storage path of where the replica will be saved
type	(optional) specifies the type of storage backend. Currently “swift” or “grid”
filename	The filename of the given replica

The first two parameters (`uri` and `type`) are optionals if a `default_storage` attribute has been set for the given repository. If not, they need to be specified, otherwise the request to the API will fail.



Please note that this API will just create a replica entry for the item, but no actual file will be uploaded from the client. Once the replica has been created you need to use the **Upload** API to transfer the actual file payload.

### Retrieve all the replicas of the given `item_id`

```
GET /v2/repos/<repo_name>/<collection_name>/<item_id>/_replicas/ HTTP/1.1
```

### Download a given replica

```
GET /v2/repos/<repo_name>/<collection_name>/<item_id>/_replicas/<rep_id> HTTP/1.1
```

### Upload a replica

Upload the file payload to the destination storage. This requires two subsequent API request.

First, ask for the destination endpoint for the upload with:

```
PUT /v2/repos/<repo_name>/<collection_name>/<item_id>/_replicas/<rep_id> HTTP/1.1
```

This will return a **temporaryURL** valid a few seconds (example):

```
{
 "uploadURI": "http://stack-server-01.ct.infn.it:8080/v2/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/gridcore/ananas."
}
```

Then use the URL returned by the previous API to upload the actual file, using the PUT verb again (example):

```
PUT http://stack-server-01.ct.infn.it:8080/v2/AUTH_51b2f4e508144fa5b0c28f02b1618bfd/gridcore/ananas.
```

It will return a 201 status code, if the upload will complete successfully

### Delete a replica

```
DELETE /v2/repos/<repo_name>/<collection_name>/<item_id>/_replicas/<rep_id> HTTP/1.1
```

### Example

## 6.2.9 Relations

One to many relations can be created between collections of the same repository, properly setting a foreign key.

To set the relation among two collections, issue the following request to the collection in the “one” side of the one-to-many relation:

```
POST /v2/repos/<repo_name>/<collection_name>/_relation HTTP/1.1
```

The body of the request needs to provide two attributes:

<b>name</b>	<b>description</b>
<i>relatedCollection</i>	the “many” side of the one-to-many relation
<i>fk</i>	the <i>foreign key</i> of <i>relatedCollection</i> that match the <i>id</i> of <i>&lt;collection_name&gt;</i>

In practice, you should set the *fk* in such a way `collection_name.id == relatedCollection.fk`

### Retrieve related items

```
GET /v2/repos/<repo_name>/<collection_name>/<item_id>/<related_collection>
```

Retrieve all the items from `related_collection` of the given `item_id`.

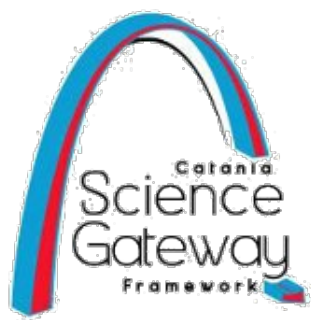
## 6.2.10 Contacts

---

## GRID & CLOUD ENGINE

---

### 7.1 About



The **Catania Grid & Cloud Engine** is a standard based middleware independent JAVA library that provides several APIs to submit and manage jobs on *Distributed Computer Infrastructures* (DCI). It is compliant with the **Open Grid Forum (OGF) Simple API for Grid Applications (SAGA)** standard.

The Catania Grid & Cloud Engine provides a standard way to interact with different DCI middlewares, so the developers can develop their applications without worrying about the details of the infrastructures where those applications will be performed.

The *Figure 1* shows the Catania Grid & Cloud Engine architecture, that consists of:

- **two interfaces:**
  - **Science GW interface:** towards the applications;
  - **DCI interface:** towards the DCI middlewares based on the SAGA Standard;
- **three modules:**
  - **Job Engine Module:** to manage jobs;
  - **Data Engine Module:** to move data towards/from the DCIs
  - **User Track & Monitoring Module:** to store information about the users interactions in the user-tracking database.

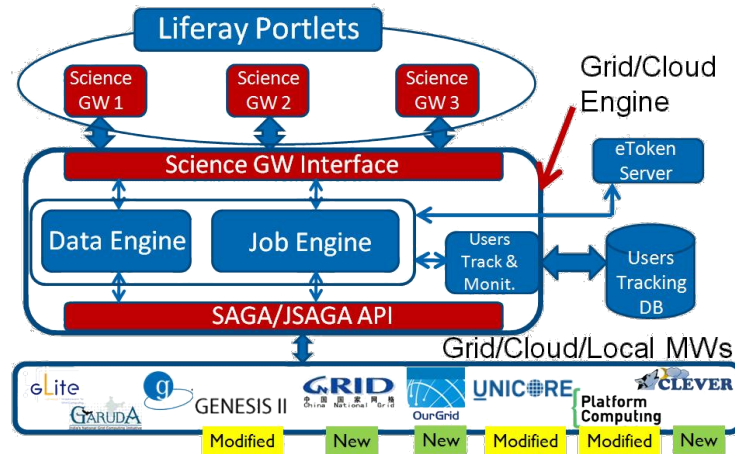


Fig. 7.1: Catania Grid & Cloud Engine Architecture

### 7.1.1 Job Engine

The *Job Engine*, one of the core component of the Catania Grid & Cloud Engine, is made of a set of libraries to develop applications able to submit and manage jobs on DCI. As said before, it is compliant with the OGF SAGA standard, and in particular adopts the JSAGA, an JAVA implementation of the SAGA standard, made by the CCIN2P3. It is optimized to be used in a Web Portal running an application server (e.g. Glassfish, Tomcat,...) based on J2EE, but can be also used also in stand-alone mode.

The Job Engine main features are:

- **Easiness:** the Job Engine allows to develop applications able to submit jobs on a DCI in a very short time, exposing a set of
  - the Job Engine periodically checks the job status;
  - when the job ends, the output will be automatically downloaded and (if set) an email will sent to notify the user.
- **Scalability:** the Job Engine is able to manage a huge number of parallel job submissions fully exploiting the HW of the machine where it is installed. A burst of parallel job submission will be enqueued and served according the hardware capabilities.
- **Performance:** delays due to grid interactions are hidden to the final users, because the Job Engine provide asynchronous functions for each job management actions.
- **Accounting:** the Job Engine provides an accounting system fully compliant with EGI VO Portal Policy and EGI Grid Security Traceability and Logging Policy.
- **Fault tollerance:** the Job Engine implements an advanced mechanism to ensure job submission and automatic re-submission mechanisms when a job fails for infrastructure related issues.

## 7.2 Installation

To install the Catania Grid & Cloud Engine first of all you have to create the Usertracking database where users interactions will be stored.

1. Create the usertracking database using the following commnad:

```
mysql -u root -p
```

```
mysql> CREATE DATABASE userstracking;
```

2. Download the SQL scripts from [here](#) and performs the following command to create the empty schema:

```
mysql -u root -p < UsersTrackingDB.sql
```

Then you need to download and configure the Catania Grid & Cloud Engine dependencies.

1. Download the GridEngine\_v1.5.10.zip from [this link](#)
2. Unzip the GridEngine\_v1.5.9.zip

```
unzip GridEngine_v1.5.9.zip
```

3. Copy the extracted lib folder under the application server /lib folder:

```
cp -r lib /opt/glassfish3/glassfish/domains/liferay/lib/
```

4. Download the attached GridEngineLogConfig.xml ([link](#)), and move this file to the Liferay config folder:

```
mv GridEngineLogConfig.xml \
/opt/glassfish3/glassfish/domains/liferay/config
```

5. Restart Glassfish server

```
/opt/glassfish3/bin asadmin-stop liferay
/opt/glassfish3/bin asadmin-start liferay
```

When the start process ends load the Glassfish Web Administration Console: <http://sg-server:4848>, fill with username liferayadmin and the password you set for the glassfish administrator and create the required resources.

## 7.2.1 JNDI Resources

Select *Resources* -> *JNDI* -> *Custom Resources* from left panel. Then on the right panel you can create the resources by clicking the **New...** button.

1. Create GridEngine-CheckStatusPool with the following parameters (*Figure 2*):

- **JNDI Name:** GridEngine-CheckStatusPool;
- **Resource Type:** it.infn.ct.ThreadPool.CheckJobStatusThreadPoolExecutor
- **Factory Class:** it.infn.ct.ThreadPool.CheckJobStatusThreadPoolExecutorFactory
- **Additional Properties:**
  - **corePoolSize:** 50
  - **maximumPoolSize:** 100
  - **keepAliveTime:** 4
  - **timeUnit:** MINUTES

- `allowCoreThreadTimeOut`: true
- `prestartAllCoreThreads`: true

**Edit Custom Resource**  
Edit an existing JNDI custom resource.  
[Load Defaults](#)

**JNDI Name:** GridEngine-CheckStatusPool

**Resource Type:** ☐ java.util.Properties ☒ it.infn.ct.ThreadPool.CheckJobStatusThreadPoolExecutor  
Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

**Factory Class:**   
Factory class for resource; implements javax.naming.spi.ObjectFactory

**Description:**

**Status:** ☒ Enabled

**Additional Properties (6)**

Name	Value
timeUnit	MINUTES
allowCoreThreadTimeOut	true
prestartAllCoreThreads	true
corePoolSize	50
maximumPoolSize	100
keepAliveTime	4

Fig. 7.2: GridEngine-CheckStatusPool JNDI Resource

2. Create GridEngine-Pool with the following parameters *Figure 3*):

- **JNDI Name:** GridEngine-Pool;
- **Resource Type:** it.infn.ct.ThreadPool.ThreadPoolExecutor
- **Factory Class:** it.infn.ct.ThreadPool.ThreadPoolExecutorFactory
- **Additional Properties:**
  - `corePoolSize`: 50
  - `maximumPoolSize`: 100
  - `keepAliveTime`: 4
  - `timeUnit`: MINUTES
  - `allowCoreThreadTimeOut`: true
  - `prestartAllCoreThreads`: true

3. Create JobCheckStatusService with the following parameters (*Figure 4*):

- **JNDI Name:** JobCheckStatusService;
- **Resource Type:** it.infn.ct.GridEngine.JobService.JobCheckStatusService
- **Factory Class:** it.infn.ct.GridEngine.JobService.JobCheckStatusServiceFactory
- **Additional Properties:**
  - `jobsupdatinginterval`: 900

4. Create JobServices-Dispatcher with the following parameters:

- **JNDI Name:** JobServices-Dispatcher;

## Edit Custom Resource

Edit an existing JNDI custom resource.

[Load Defaults](#)

JNDI Name: GridEngine-Pool

Resource Type: ☐ java.util.Properties

☒ it.infn.ct.ThreadPool.ThreadPoolExecutor

Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

Factory Class:

Factory class for resource; implements javax.naming.spi.ObjectFactory

Description:

Status: ☒ Enabled

Additional Properties (6)		
<input checked="" type="checkbox"/>	<input type="button" value="Add Property"/>	<input type="button" value="Delete Properties"/>
Name	Value	
<input type="checkbox"/> prestartAllCoreThreads	<input type="text" value="true"/>	
<input type="checkbox"/> allowCoreThreadTimeOut	<input type="text" value="true"/>	
<input type="checkbox"/> timeUnit	<input type="text" value="MINUTES"/>	
<input type="checkbox"/> keepAliveTime	<input type="text" value="4"/>	
<input type="checkbox"/> maximumPoolSize	<input type="text" value="100"/>	
<input type="checkbox"/> corePoolSize	<input type="text" value="50"/>	

Fig. 7.3: GridEngine-Pool JNDI Resource

## Edit Custom Resource

Edit an existing JNDI custom resource.

[Load Defaults](#)

JNDI Name: JobCheckStatusService

Resource Type: ☐ java.util.Properties

☒ it.infn.ct.GridEngine.JobService.JobCheckStatusService

Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

Factory Class:

Factory class for resource; implements javax.naming.spi.ObjectFactory

Description:

Status: ☒ Enabled

Additional Properties (1)		
<input checked="" type="checkbox"/>	<input type="button" value="Add Property"/>	<input type="button" value="Delete Properties"/>
Name	Value	
<input type="checkbox"/> jobupdatinginterval	<input type="text" value="900"/>	

Fig. 7.4: JobCheckStatusService JNDI Resource

- **Resource Type:** `it.infn.ct.GridEngine.JobService.JobServicesDispatcher`
- **Factory Class:** `it.infn.ct.GridEngine.JobService.JobServicesDispatcherFactory`
- **Additional Properties:**
  - **retrycount:** 3;
  - **resubnumber:** 10;
  - **myproxyservers:** `gridit=myproxy.ct.infn.it; prod.vo.eu-eela.eu=myproxy.ct.infn.it; cometa=myproxy.ct.infn.it; eumed=myproxy.ct.infn.it; vo.eu-decide.eu=myproxy.ct.infn.it; sagrid=myproxy.ct.infn.it; euindia=myproxy.ct.infn.it; see=myproxy.ct.infn.it;`

### Edit Custom Resource

Edit an existing JNDI custom resource.

[Load Defaults](#)

**JNDI Name:** `JobServices-Dispatcher`

**Resource Type:** ☐ `java.util.Properties` ☒ `it.infn.ct.GridEngine.JobService.JobServicesDispatcher`  
Enter a fully qualified type following the format xxx.xxx (for example, javax.jms.Topic)

**Factory Class:** `it.infn.ct.GridEngine.JobService.JobServicesDispatcherFactory`  
Factory class for resource; implements javax.naming.spi.ObjectFactory

**Description:**

**Status:** ☒ Enabled

Additional Properties (2)	
Name	Value
retrycount	3
resubnumber	10

Fig. 7.5: JobServices-Dispatcher JNDI Resource

Now you have to create the required JDBC Connection Pools. Select *Resources -> JDBC -> JDBC Connection Pools* from left panel. On the right panel you can create the resources by clicking the **New...** button.

- **Create UserTrackingPool with the following parameters:**
  - **General Settings (Step 1/2)** see [Figure 6](#):
    - \* **Pool Name:** `usertrackingPool`
    - \* **Resource Type:** select `javax.sql.DataSource`
    - \* **Database Driver Vendor:** select `MySQL`
    - \* Click Next
  - **Advanced Settings (Step 2/2)** [Figure 7](#):
    - \* **Edit the default parameters in Pool Settings using the following values:**
      - **Initial and Minimum Pool Size:** 64
      - **Maximum Pool Size:** 256



\* **Select all default Additional properties and delete them**

- Add the following properties:

Name	Value
Url	jdbc:mysql://sg-database:3306/userstracking
User	tracking_user
False	usertracking

\* Click Save

Please pay attention to the Url property, **sg-database** should be replaced with the correct Url of your database machine. You can check if you have correctly configured the Connection Pool by clicking on Ping button, you should see the message **Ping Succeeded**, otherwise please check your configuration.

### Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

[Load Defaults](#) [Flush](#) [Ping](#)

#### General Settings

**Pool Name:** UserTrackingPool

**Resource Type:**  Must be specified if the datasource class implements more than 1 of the interface.

**Datasource Classname:**  Vendor-specific classname that implements the DataSource and/or XADataSource APIs

**Driver Classname:**  Vendor-specific classname that implements the java.sql.Driver interface.

**Ping:** ☒ **Enabled**  
When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes

**Description:**

#### Pool Settings

**Initial and Minimum Pool Size:**  **Connections**  
Minimum and initial number of connections maintained in the pool

**Maximum Pool Size:**  **Connections**  
Maximum number of connections that can be created to satisfy client requests

**Pool Resize Quantity:**  **Connections**  
Number of connections to be removed when pool idle timeout expires

**Idle Timeout:**  **Seconds**  
Maximum time that connection can remain idle in the pool

**Max Wait Time:**  **Milliseconds**  
Amount of time caller waits before connection timeout is sent

#### Transaction

**Non Transactional Connections:** ☒ **Enabled**  
Returns non-transactional connections

**Transaction Isolation:**   
If unspecified, use default level for JDBC Driver

**Isolation Level:** ☒ **Guaranteed**  
All connections use same isolation level; requires Transaction Isolation

Fig. 7.6: UsersTrackingPool JDBC General settings

Finally, you have to create the required JDBC Resources. Select *Resources -> JDBC -> JDBC Resources* from left panel. On the right panel you can create the resources by clicking the **New...** button.

- Create jdbc/UserTrackingPool with the following parameter (*Figure 8*):
  - **JNDI Name:** jdbc/UserTrackingPool;
  - **Pool name:** select usertrackingPool.
- Create jdbc/hibernatepool with the following parameter *Figure 9*:
  - **JNDI Name:** jdbc/hibernatepool;
  - **Pool name:** select usertrackingPool.

General	Advanced	Additional Properties
---------	----------	-----------------------

### Edit JDBC Connection Pool Properties

Modify properties of an existing JDBC connection pool.

Additional Properties (3)	
Name	Value
Password	usertracking
User	tracking_user
Url	jdbc:mysql://10.10.10.10:3306/usertracking

Fig. 7.7: UsersTrackingPool JDBC Advanced settings

## Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

**JNDI Name:** jdbc/UserTrackingPool

**Pool Name:**

Use the [JDBC Connection Pools](#) page to create new pools

**Description:**

**Status:** ☒ **Enabled**

Additional Properties (0)	
Name	Value
No items found.	

Fig. 7.8: jdbcUsersTrackingPool JDBC Resource

## Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

**JNDI Name:** jdbc/gehibernatepool

**Pool Name:**  ▼

Use the [JDBC Connection Pools](#) page to create new pools

**Description:**

**Status:** ☒ **Enabled**

Additional Properties (0)	
<a href="#">Add Property</a>	<a href="#">Delete Properties</a>
Name	Value
No items found.	

Fig. 7.9: jdbcgehibernatepool JDBC Resource

Now, restart glassfish to save the resources.

## 7.3 Usage

Once you have successfully installed and configured the Catania Grid & Cloud Engine, to exploit all its features you could download and deploy one of ours portlets available on the [GitHub csgf repository](#). As an example you could referer to the [mi-hostname-portlet](#) to get info in how to install and use this portlet.

## 7.4 Contributors

Diego SCARDACI

Mario TORRISI



---

## JSAGA ADAPTOR FOR ROCCI-BASED CLOUDS

---

### 8.1 About



---

The Simple API for Grid Applications (SAGA) is a family of related standards specified by the Open Grid Forum [1] to define an application programming interface (API) for common distributed computing functionality.

These APIs does not strive to replace Globus or similar grid computing middleware systems, and does not target middleware developers, but application developers with no background on grid computing. Such developers typically wish to devote their time to their own goals and minimize the time spent coding infrastructure functionality. The API insulates application developers from middleware.

The specification of services, and the protocols to interact with them, is out of the scope of SAGA. Rather, the API seeks to hide the detail of any service infrastructures that may or may not be used to implement the functionality that the application developer needs. The API aligns, however, with all middleware standards within Open Grid Forum (OGF).

JSAGA [2] is a Java implementation of the Simple API for Grid Applications (SAGA) specification from the Open Grid Forum (OGF) [1]. It permits seamless data and execution management between heterogeneous grid infrastructures.

The current stable release is available at:

- <http://software.in2p3.fr/jsaga/>

The current development SNAPSHOT is available at:

- <http://software.in2p3.fr/jsaga/dev/>

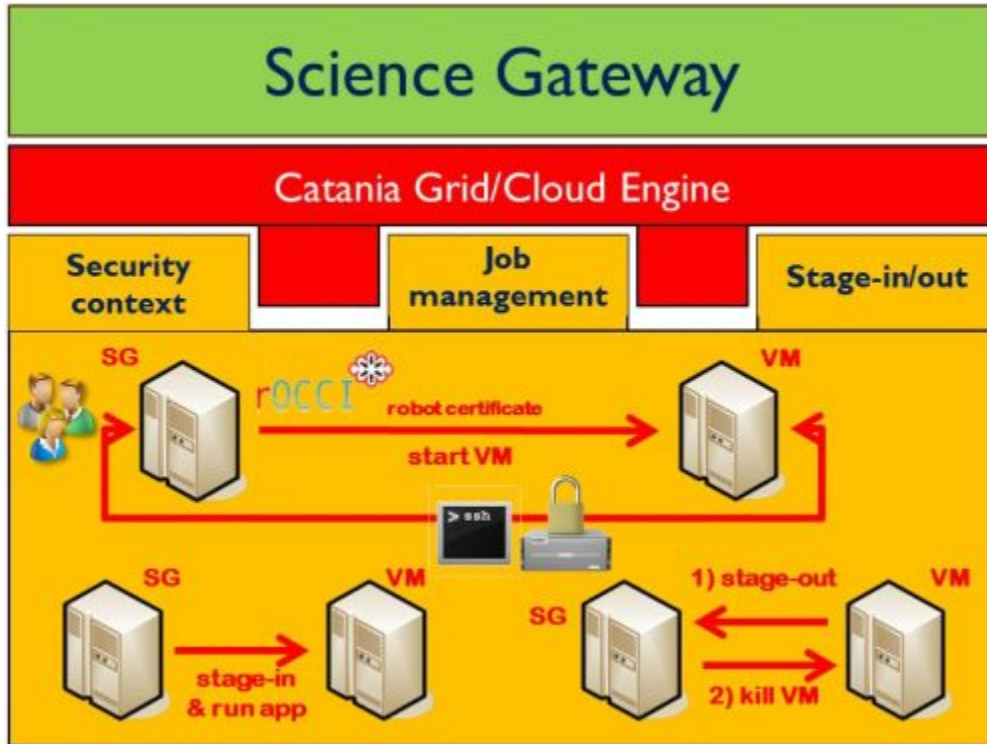
In the contest of the **CHAIN\_REDS** project a new JSAGA adaptor for OCCI-complaint [3] cloud middleware has been developed to demonstrate the interoperability between different open-source cloud providers.

Using the rOCCI implementation of the OCCI standard, the adaptor takes care of:

1. switching-on the VM pre-installed with the required application,
2. establishing a secure connection to it signed using a digital “robot” certificate,
3. staging the input file(s) in the VM,

4. executing the application,
5. retrieving the output file(s) at the end of the computation and
6. killing the VM.

The high-level architecture of the JSAGA adaptor for OCCI-complaint cloud middleware is shown in the below figure:



## 8.2 Installation

- Import this Java application into your preferred IDE (e.g. Netbeans).
- Configure the application with the needed JSAGA jars files.
- Configure the `src/test/RunTest.java` with your settings:

```
// Setting the CHAIN-REDS Contextualisation options
OCCI_PROXY_PATH = System.getProperty("user.home") +
 System.getProperty("file.separator") +
 "jsaga-adaptor-rocci" +
 System.getProperty("file.separator") +
 "x509up_u501";

// === OCCI SETTINGS for the INFN-STACK CLOUD RESOURCE === //
OCCI_ENDPOINT_HOST = "rocci://stack-server-01.ct.infn.it";
OCCI_ENDPOINT_PORT = "8787";
// Possible OCCI_OS values: 'generic_vm', 'octave' and 'r'
// f36b8eb8-8247-4b4f-a101-18c7834009e0 ==> generic_vm
// bb623e1c-e693-4c7d-a90f-4e5bf96b4787 ==> octave
// 91632086-39ef-4e52-a6d1-0e4f1bf95a7b ==> r
// 6ee0e31b-e066-4d39-86fd-059b1de8c52f ==> WRF
```

```

OCCI_OS = "f36b8eb8-8247-4b4f-a101-18c7834009e0";
OCCI_FLAVOR = "small";

OCCI_VM_TITLE = "rOCCI";
OCCI_ACTION = "create";

[...]
```

```

desc.setVectorAttribute(
 JobDescription.FILETRANSFER,
 new String[]{
 System.getProperty("user.home") +
 System.getProperty("file.separator") +
 "jsaga-adaptor-rocci" +
 System.getProperty("file.separator") +
 "job-generic.sh>job-generic.sh",

 System.getProperty("user.home") +
 System.getProperty("file.separator") +
 "jsaga-adaptor-rocci" +
 System.getProperty("file.separator") +
 "output.txt<output.txt",

 System.getProperty("user.home") +
 System.getProperty("file.separator") +
 "jsaga-adaptor-rocci" +
 System.getProperty("file.separator") +
 "error.txt<error.txt"
 }
);

```

- Create a simple bash script:

```

]$ cat job-generic.sh
#!/bin/sh
sleep 15
echo "General Info ...> This is a CHAIN-REDS test VM. See below server details "
echo "-----"
echo "Running host ...> " `hostname -f`
echo "IP address> " `/sbin/ifconfig | grep "inet addr:" \
 | head -1 | awk '{print $2}' | awk -F':' '{print $2}'`

echo "Kernel> " `uname -r`
echo "Distribution ...> " `head -n1 /etc/issue`
echo "Arch> " `uname -a | awk '{print $12}'`
echo "CPU> " `cat /proc/cpuinfo | grep -i "model name" \
 | head -1 | awk -F ':' '{print $2}'`

echo "Memory> " `cat /proc/meminfo | grep MemTotal | awk '{print $2}'` KB
echo "Partitions> " `cat /proc/partitions`
echo "Uptime host ...> " `uptime | sed 's/.*up ([^,]*), .*/1/'`
echo "Timestamp> " `date`
echo "-----"
echo "http://www.chain-project.eu/"
echo "Copyright © 2015"

```

- Compile the application with your IDE.

In case of successful compilation you should get the following output message:

```

init:
deps-clean:
 Updating property file: /home/larocca/jsaga-adaptor-rocci/build/built-clean.properties
 Deleting directory /home/larocca/jsaga-adaptor-rocci/build
clean:
init:
deps-jar:
 Created dir: /home/larocca/jsaga-adaptor-rocci/build
 Updating property file: /home/larocca/jsaga-adaptor-rocci/build/built-jar.properties
 Created dir: /home/larocca/jsaga-adaptor-rocci/build/classes
 Created dir: /home/larocca/jsaga-adaptor-rocci/build/empty
 Created dir: /home/larocca/jsaga-adaptor-rocci/build/generated-sources/ap-source-output
 Compiling 7 source files to /home/larocca/jsaga-adaptor-rocci/build/classes
 warning: [options] bootstrap class path not set in conjunction with -source 1.6
 1 warning
 Copying 4 files to /home/larocca/jsaga-adaptor-rocci/build/classes
compile:
 Created dir: /home/larocca/jsaga-adaptor-rocci/dist
 Copying 1 file to /home/larocca/jsaga-adaptor-rocci/build
 Copy libraries to /home/larocca/jsaga-adaptor-rocci/dist/lib.
 Building jar: /home/larocca/jsaga-adaptor-rocci/dist/jsaga-adaptor-rocci.jar
 To run this application from the command line without Ant, try:
 java -jar "/home/larocca/jsaga-adaptor-rocci/dist/jsaga-adaptor-rocci.jar"
jar:
 BUILD SUCCESSFUL (total time: 10 seconds)

```

## 8.3 Usage

- Create a RFC proxy certificate for your given VO:

```

]$ voms-proxy-init --voms vo.chain-project.eu -rfc
Enter GRID pass phrase for this identity:
Contacting voms.ct.infn.it:15011
[/C=IT/O=INFN/OU=Host/L=Catania/CN=voms.ct.infn.it] "vo.chain-project.eu".
Remote VOMS server contacted succesfully.

Created proxy in /tmp/x509up_u501.
Your proxy is valid until Wed Jun 03 22:38:16 CEST 2015

```

- Check if your RFC proxy certificate is valid:

```

]$ voms-proxy-info --all
subject : /C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Giuseppe La Rocca/CN=1660223179
issuer : /C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Giuseppe La Rocca
identity : /C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Giuseppe La Rocca
type : RFC3820 compliant impersonation proxy
strength : 1024
path : /tmp/x509up_u501
timeleft : 11:59:53
key usage : Digital Signature, Key Encipherment, Data Encipherment
=== VO vo.chain-project.eu extension information ===
VO : vo.chain-project.eu
subject : /C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Giuseppe La Rocca
issuer : /C=IT/O=INFN/OU=Host/L=Catania/CN=voms.ct.infn.it

```



```
attribute : /vo.chain-project.eu/Role=NULL/Capability=NULL
timeleft : 11:59:53
uri : voms.ct.infn.it:15011
```

- To test the JSAGA adaptor for OCCI-complaint cloud middleware without Ant, try:

```
]$ java -jar "/home/larocca/jsaga-adaptor-rocci/dist/jsaga-adaptor-rocci.jar"

init:
 Deleting: /home/larocca/jsaga-adaptor-rocci/build/built-jar.properties

deps-jar:
 Updating property file: /home/larocca/jsaga-adaptor-rocci/build/built-jar.properties
 Compiling 1 source file to /home/larocca/jsaga-adaptor-rocci/build/classes

warning: [options] bootstrap class path not set in conjunction with -source 1.6
1 warning

compile-single:

run-single:

10:58:02 INFO [RunTest:152]
Initialize the security context for the rOCCI JSAGA adaptor
10:58:02 Failed to load engine properties, using defaults \
 [./etc/jsaga-config.properties (No such file or directory)]

10:58:05
10:58:05 Initializing the security context for the rOCCI JSAGA adaptor [SUCCESS]
10:58:05 See below security context details...
10:58:05 User DN = /C=IT/O=INFN/OU=Personal Certificate/L=Catania/CN=Giuseppe La Rocca
10:58:05 Proxy = /home/larocca/jsaga-adaptor-rocci/x509up_u501
10:58:05 Lifetime = 11h.
10:58:05 CA Repos = /etc/grid-security/certificates
10:58:05 Type = rocci
10:58:05 VO name = vo.chain-project.eu
10:58:05
10:58:05 Initialize the JobService context...
10:58:05 serviceURL = \
 rocci://stack-server-01.ct.infn.it:8787/?prefix=&attributes_title=rOCCI&\
 mixin_os_tpl=f36b8eb8-8247-4b4f-a101-18c7834009e0&\
 mixin_resource_tpl=small&\
 user_data=&\
 proxy_path=/home/larocca/jsaga-adaptor-rocci/x509up_u501

10:58:05
10:58:05 Trying to connect to the cloud host [stack-server-01.ct.infn.it]
10:58:05
10:58:05 See below the details:
10:58:05
10:58:05 PREFIX =
10:58:05 ACTION = create
10:58:05 RESOURCE = compute
10:58:05
10:58:05 AUTH = x509
10:58:05 PROXY_PATH = /home/larocca/jsaga-adaptor-rocci/x509up_u501
10:58:05 CA_PATH = /etc/grid-security/certificates
10:58:05
```

```

10:58:05 HOST = stack-server-01.ct.infn.it
10:58:05 PORT = 8787
10:58:05 ENDPOINT = https://stack-server-01.ct.infn.it:8787/
10:58:05 PUBLIC KEY = /home/larocca/.ssh/id_rsa.pub
10:58:05 PRIVATE KEY = /home/larocca/.ssh/id_rsa
10:58:05
10:58:05 EGI FedCloud Contextualisation options:
10:58:05 USER DATA =
10:58:05
10:58:07 Creating a new OCCI computeID. Please wait!
10:58:07 VM Title = rOCCI
10:58:07 OS = f36b8eb8-8247-4b4f-a101-18c7834009e0
10:58:07 Flavour = small
10:58:07
10:58:07 occi --endpoint https://stack-server-01.ct.infn.it:8787/ \
--action create --resource compute \
--attribute occi.core.title=rOCCI \
--mixin os_tpl#f36b8eb8-8247-4b4f-a101-18c7834009e0 \
--mixin resource_tpl#small \
--auth x509 --user-cred /home/larocca/jsaga-adaptor-rocci/x509up_u501 \
--voms --ca-path /etc/grid-security/certificates

10:58:13 EXIT CODE = 0
10:58:13
10:58:13 A new OCCI computeID has been created:
https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2
10:58:23
10:58:23 See below the details of the VM
10:58:23
[https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2]
10:58:23
10:58:23 occi --endpoint https://stack-server-01.ct.infn.it:8787/ \
--action describe \
--resource compute \
--resource \
https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2 \
--auth x509 --user-cred /home/larocca/jsaga-adaptor-rocci/x509up_u501 \
--voms --ca-path /etc/grid-security/certificates \
--output-format json_extended_pretty

10:58:28 EXIT CODE = 0
10:58:28
10:58:28 [
10:58:28 {
10:58:28 "kind": "http://schemas.ogf.org/occi/infrastructure#compute",
10:58:28 "mixins": [
10:58:28 "http://schemas.openstack.org/compute/instance#os_vms",
10:58:28 "http://schemas.openstack.org/template/os#f36b8eb8-8247-4b4f-a101-18c7834009e0",
10:58:28],
10:58:28 "actions": [
10:58:28 "http://schemas.ogf.org/occi/infrastructure/compute/action#stop",
10:58:28 "http://schemas.ogf.org/occi/infrastructure/compute/action#suspend",
10:58:28 "http://schemas.ogf.org/occi/infrastructure/compute/action#restart",
10:58:28 "http://schemas.openstack.org/instance/action#create_image",
10:58:28 "http://schemas.openstack.org/instance/action#chg_pwd"
10:58:28],
10:58:28 "attributes": {
10:58:28 "occi": {

```

```

10:58:28 "core": {
10:58:28 "id": "845593b9-2e31-4f6e-9fa0-7386476373f2"
10:58:28 },
10:58:28 "compute": {
10:58:28 "architecture": "x86",
10:58:28 "cores": "1",
10:58:28 "hostname": "rocci",
10:58:28 "memory": "1.0",
10:58:28 "speed": "0.0",
10:58:28 "state": "active"
10:58:28 },
10:58:28 },
10:58:28 "org": {
10:58:28 "openstack": {
10:58:28 "compute": {
10:58:28 "console": {
10:58:28 "vnc": \
 "http://212.189.145.95:6080/vnc_auto.html?token=7cdfb12e-96d3-4e4c-9881-7fd0fe363110"
10:58:28 },
10:58:28 "state": "active"
10:58:28 }
10:58:28 }
10:58:28 },
10:58:28 "id": "845593b9-2e31-4f6e-9fa0-7386476373f2",
10:58:28 "links": [
10:58:28 {
10:58:28 "kind": "http://schemas.ogf.org/occi/infrastructure#networkinterface",
10:58:28 "mixins": [
10:58:28 "http://schemas.ogf.org/occi/infrastructure/networkinterface#ipnetworkinterface"
10:58:28],
10:58:28 "attributes": {
10:58:28 "occi": {
10:58:28 "networkinterface": {
10:58:28 "gateway": "0.0.0.0",
10:58:28 "mac": "aa:bb:cc:dd:ee:ff",
10:58:28 "interface": "eth0",
10:58:28 "state": "active",
10:58:28 "allocation": "static",
10:58:28 "address": "90.147.16.130"
10:58:28 },
10:58:28 "core": {
10:58:28 "source": "/compute/845593b9-2e31-4f6e-9fa0-7386476373f2",
10:58:28 "target": "/network/public",
10:58:28 "id": "/network/interface/03fc1144-b136-4876-9682-d1f5647aa281"
10:58:28 }
10:58:28 },
10:58:28 },
10:58:28 "id": "/network/interface/03fc1144-b136-4876-9682-d1f5647aa281",
10:58:28 "rel": "http://schemas.ogf.org/occi/infrastructure#network",
10:58:28 "source": "/compute/845593b9-2e31-4f6e-9fa0-7386476373f2",
10:58:28 "target": "/network/public"
10:58:28 },
10:58:28 {
10:58:28 "kind": "http://schemas.ogf.org/occi/infrastructure#networkinterface",
10:58:28 "mixins": [
10:58:28 "http://schemas.ogf.org/occi/infrastructure/networkinterface#ipnetworkinterface"
10:58:28],

```

```
10:58:28 "attributes": {
10:58:28 "occi": {
10:58:28 "networkinterface": {
10:58:28 "gateway": "192.168.100.1",
10:58:28 "mac": "fa:16:3e:2f:23:35",
10:58:28 "interface": "eth0",
10:58:28 "state": "active",
10:58:28 "allocation": "static",
10:58:28 "address": "192.168.100.4"
10:58:28 },
10:58:28 "core": {
10:58:28 "source": "/compute/845593b9-2e31-4f6e-9fa0-7386476373f2",
10:58:28 "target": "/network/admin",
10:58:28 "id": "/network/interface/c313ca29-0e86-4162-8994-54dfd45756a2"
10:58:28 }
10:58:28 },
10:58:28 "id": "/network/interface/c313ca29-0e86-4162-8994-54dfd45756a2",
10:58:28 "rel": "http://schemas.opengroup.org/occi/infrastructure#network",
10:58:28 "source": "/compute/845593b9-2e31-4f6e-9fa0-7386476373f2",
10:58:28 "target": "/network/admin"
10:58:28 }
10:58:28]
10:58:28 }
10:58:28 }
10:58:28
10:58:28 Starting VM [90.147.16.130] in progress...
10:58:28
10:58:28 Waiting the remote VM finishes the boot! Sleeping for a while...
10:58:28 Wed 2015.06.03 at 10:58:28 AM CEST
10:59:32 [SUCCESS]
10:59:32 Wed 2015.06.03 at 10:59:32 AM CEST
10:59:36
10:59:36 Job instance created:
10:59:36 [rocci://stack-server-01.ct.infn.it:8787/?prefix=&\
 attributes_title=rOCCI&\
 mixin_os_tpl=f36b8eb8-8247-4b4f-a101-18c7834009e0&\
 mixin_resource_tpl=small&\
 user_data=&\
 proxy_path=/home/larocca/jsaga-adaptor-rocci/x509up_u501]-\
 [a991707d-3c4b-4a2f-9427-7bf19ded17b5@90.147.16.130#\
 https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2]
10:59:36
10:59:36 Closing session...
10:59:36
10:59:36 Re-initialize the security context for the rOCCI JSAGA adaptor
10:59:37
10:59:37 Trying to connect to the cloud host [stack-server-01.ct.infn.it]
10:59:37
10:59:37 See below the details:
10:59:37
10:59:37 PREFIX =
10:59:37 ACTION = create
10:59:37 RESOURCE = compute
10:59:37
10:59:37 AUTH = x509
10:59:37 PROXY_PATH = /home/larocca/jsaga-adaptor-rocci/x509up_u501
```

```

10:59:37 CA_PATH = /etc/grid-security/certificates
10:59:37
10:59:37 HOST = stack-server-01.ct.infn.it
10:59:37 PORT = 8787
10:59:37 ENDPOINT = https://stack-server-01.ct.infn.it:8787/
10:59:37 PUBLIC KEY = /home/larocca/.ssh/id_rsa.pub
10:59:37 PRIVATE KEY = /home/larocca/.ssh/id_rsa
10:59:37
10:59:37 EGI FedCloud Contextualisation options:
10:59:37 USER DATA =
10:59:37
10:59:37
10:59:37 Fetching the status of the job
10:59:37 [a991707d-3c4b-4a2f-9427-7bf19ded17b5@90.147.16.130#\
 https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2]
10:59:37
10:59:37 JobID [
 [rocci://stack-server-01.ct.infn.it:8787/?prefix=&\
 attributes_title=rOCCI&\
 mixin_os_tpl=f36b8eb8-8247-4b4f-a101-18c7834009e0&\
 mixin_resource_tpl=small&\
 user_data=&\
 proxy_path=/home/larocca/jsaga-adaptor-rocci/x509up_u501]-\
 [a991707d-3c4b-4a2f-9427-7bf19ded17b5@90.147.16.130#\
 https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2]
]
10:59:37
10:59:37 Calling the getStatus() method
10:59:37 Current Status = RUNNING
10:59:37 Execution Host = 90.147.16.130
10:59:37
10:59:37 Unexpected job status: RUNNING
10:59:48
10:59:48 Calling the getStatus() method
10:59:48 Current Status = RUNNING
10:59:48 Execution Host = 90.147.16.130
10:59:48
10:59:48 Unexpected job status: RUNNING
10:59:58
10:59:58 Calling the getStatus() method
10:59:58 Current Status = DONE
10:59:58 Execution Host = 90.147.16.130
10:59:58 Calling the getExitCode() method
10:59:58
10:59:58 Final Job Status = DONE
10:59:58 Exit Code (0) [SUCCESS]
10:59:58
10:59:58 Retrieving job results.
10:59:58 This operation may take a few minutes to complete...
11:00:03 Calling the getCreated() method
11:00:04 Calling the getStarted() method
11:00:04 Calling the getFinished() method
11:00:04 Calling the getExitCode() method
11:00:04
11:00:04 Stopping the VM [90.147.16.130] in progress...
11:00:04 occi --endpoint https://stack-server-01.ct.infn.it:8787/ \
 --action delete \
 --resource compute \

```

```
--resource \
https://stack-server-01.ct.infn.it:8787/compute/845593b9-2e31-4f6e-9fa0-7386476373f2 \
--auth x509 \
--user-cred /home/larocca/jsaga-adaptor-rocci/x509up_u501 \
--voms \
--ca-path /etc/grid-security/certificates

11:00:08 EXIT CODE = 0
11:00:08

11:00:08 Job outputs retrieved [SUCCESS]
11:00:08
11:00:08 Initialize the JobService context [SUCCESS]
BUILD SUCCESSFUL (total time: 2 minutes 7 seconds)
```

- Check results:

```
]$ cat output.txt
General Info ...> This is a CHAIN-REDS test VM. See below server details

Running host ...>
IP address> 192.168.100.4
Kernel> 2.6.32-504.3.3.el6.i686
Distribution ...> CentOS release 6.6 (Final)
Arch> i686
CPU> AMD Opteron 62xx class CPU
Memory> 1030588 KB
Partitions> major minor #blocks name 253 0 10485760 vda 253 1 204800 vda1 ...
Uptime host> 11:13:48 up 1 min, 0 users, load average: 0.15, 0.06, 0.02
Timestamp> Wed Jun 3 11:13:48 CEST 2015

http://www.chain-project.eu/
Copyright © 2015
```

## 8.4 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Diego SCARDACI - Italian National Institute of Nuclear Physics ([INFN](#))

## 9.1 About

Portlet for managing jobs.

## 9.2 Installation

Deploy MyJobs.war with: `cp MyJobs $LIFERAY_HOME/deploy/`

Watch the Liferay' server.log file till: 'MyJobs successfully deployed'

### 9.2.1 Requirements

Application registration in the GridOperations table is mandatory for the MyJobs portlet

- ▶ Each user action which involves the distributed infrastructure will be tracked by the **UsersTracking** Database
- ▶ The GridEngine uses the **GridOperations** table to register applications and services accessing the distributed infrastructure

```
mysql -u tracking_user -pusertracking usertracking
desc GridOperations;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
portal	varchar(120)	NO		NULL	
description	varchar(200)	NO		NULL	

- ▶ We can register and then track the activity of our developed portlet with:

```
insert into GridOperations values (9,'test','testPortlet');
```

*How to get GridOperations values*

- id – Just a numeric value; ‘9’ historically used by Tester Apps
- portal – Value added in the preference - liferay control panel
- description – Use any human readable application description

GridOperations values will be carefully selected for production portals

To check the job status and to retrieve the output when job is done, you should install our MyJob portlet, in order to do this you have to make some configuration in your liferay environment.

- Open the Glassfish Administration Console (<http://localhost:4848>).
- Create a new JDBC connection pool for MyJob portlet:

**\*\* On the left menu select Resources > JDBC > JDBC Connection Pools**

Click New... to create a new pool with the following settings:

Pool Name: usertrackingPool

ResourceType: javax.sql.DataSource

Database Driver Vendor: select MySql

Click Next and left the default parameters;

Select and remove all the properties from the "Additional Properties" table (bottom page);

Click on "Add" and create the following three properties:

Name: Url, Value: jdbc:mysql://localhost:3306/usertracking

Name: User, Value: tracking\_user

Name: Password, Value: usertracking

Click on "Finish" button to save configuration.

- Click on the ‘Ping’ button to test the connection pool. If everything is working fine the “Ping Succeeded” message should appear on top.
- Create a new JDBC Resource:

On the left menu select Resources > JDBC > JDBC Resources

Click New... to create a new JDBC Resource with the following settings:

JNDI Name: jdbc/UserTrackingPool

Pool Name: select usertrackingPool

Click on “Finish” button to save configuration.

- Restart Glassfish

When restart procedure has completed you can proceed with the installation of the MyJob portlet.



## 9.3 Usage

**Active Jobs**

MyJobs

Active Jobs List Done Jobs List

The table below shows the status of your jobs.

Statuses are automatically updated every 15 minutes so there is no need to reload this page more frequently. However, if you don't see your jobs in the table within a reasonable amount of time (a couple of hours at most), click on Help in the MyWorkspace portlet and notify us the problem.

Once your jobs have finished, you have 96 hours to retrieve their output. Beyond that time, the output of your jobs will automatically be deleted from the Science Gateway in order not to fill its storage with undesired stuff.

Copy Print Save Download Job output Search: [ ]

Show 10 entries First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	testjob	multi-infrastructure demo job description1	2013-07-17 13:36:52.0	

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Copy Print Save Download Job output Search: [ ]

Show 10 entries First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	mi-parallel-app	Demo Collection	2013-07-27 15:23:02.0	RUNNING

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Copy Print Save Download Job output Search: [ ]

Show 10 entries First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	Job Collections	mi-parallel-app	Demo Collection	2013-07-27 15:23:02.0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

**Collection status**

**Jobs current status**

**Collection DONE**

## 9.4 Contributors

- Rita Ricceri



---

## OPENID CONNECT FOR LIFERAY

---

### 10.1 About

OpenId Connect for Liferay is a very rough but effective implementation of the OpenId connect protocol for Liferay. Using this class it is possible to authenticate with any OpenId provider specified in the code.

### 10.2 Installation

Before to start you must have a Liferay instance already deployed and executing properly.

Edit the file

```
src/main/java/it/infnt/security/liferay/openidconnect/Utils/Authenticator.java
```

to modify the client-id, the secret and the callback using the information provided by the OpenId Connect server you want to use. The other values reference to the *EGI access portal authentication service*. If you plan to use a different *OpenID Connect provider* the urls to the service need to be modified with the values provided by your provider (this version does not use service description so all the urls should be modified).

Create the package with **maven** executing the command:

```
$ mvn clean install
```

Maven will create two jar files Inside the directory `target`, one including all dependencies (with *with-dependencies* suffix) and the other without. Copy the one with dependencies inside the `lib` directory of Liferay (locate Liferay inside your application server, this will contain the directory `WEB-INF/lib` where copy the jar).

Edit the Liferay file `portal-ext.properties` (if you have not create a new one in `WEB-INF/classes`) and add the new `AutoLogin` class:

```
auto.login.hooks=\
 it.infnt.security.liferay.openidconnect.OpenIdConnectAutoLogin,\
 com.liferay.portal.security.auth.CASAutoLogin,\
 com.liferay.portal.security.auth.FacebookAutoLogin,\
 com.liferay.portal.security.auth.NtlmAutoLogin,\
 com.liferay.portal.security.auth.OpenIdAutoLogin,\
 com.liferay.portal.security.auth.OpenSSOAutoLogin,\
 com.liferay.portal.security.auth.RememberMeAutoLogin,\
 com.liferay.portal.security.auth.SiteMinderAutoLogin
```

Finally, edit the sign-in link in your theme in order to redirect the user to the URL:

```
/c/portal/login?openIdLogin=true
```

This allow to authente users using the sign-in link in the page. If you access a protected page or open the login portlet the login form still is available. It is suggested to disable the portlet if you plan to use only OpenId Connect.

## 10.3 Usage

Users have to sign-in to the portal using the provided link *Sign-in* as explained in the section *Installation*. The only difference is that the other sign-in procedure must be disabled so the user cannot see the login for sh/she is used to.

## 10.4 Contributors

- [Marco Fargetta](#)

### 10.4.1 Contribution

A revised version of this repository will be merged with the [federated-login-ext repository](#) therefore new contribution should go to that.

---

## SG-MON - INSTALLATION AND CONFIGURATION

---

### 11.1 About

This document covers installation and configuration of SG-Mon. A brief overview of SG-Mon is also provided.

### 11.2 Introduction to SG-Mon

SG-Mon is a collection of python scripts, developed with the purpose of monitoring availability and reliability of web services and portlets running on Catania Science Gateway Framework. SG-Mon scripts are intended as plugin for network monitoring tools : this guide covers Nagios, but with minor modifications they can be adapted to Zabbix. Currently, it is composed by the following modules :

- SGApp : run test instances of a given Scientific Gateway application
- eTokenServer : verify the eTokenserver instance are up and properly working
- Open Access Repository Login : verify login to an Open Access Repository
- Virtuoso : verify that Virtuoso store instances are up and properly responding to queries

### 11.3 Requirements

- A working installation of Nagios (v3 or above)
- Java (v 1.7 or above)
- Apache jMeter
- Python v. 2.7 (2.6 should be working as well).

Depending on the checks being actually activated, there could be further dependencies, which are generally mentioned in the preamble of each probe.

### 11.4 Installation

The easiest way to install SG-Mon is to clone this repository (or download the ZIP archive). Copy the content of AppChecks folder in a directory able to execute plugins (eg. /usr/local/nagios/myplugins). Create the directory if

needed. A part from NagiosCheck.py, which exports some functions imported by other modules, all the other SG-Mon modules are each other independent; see Configuration section in order to find out how to setup properly each of the modules.

## 11.5 Configuration

Conforming with Nagios good practices, all SG-Mon modules return 0 if the service status is OK, 1 if WARNING and 2 if CRITICAL. In any case, the module returns a message with the output of the metric used for the probe.

### 11.5.1 SGApp

This module is the most complex within SGMon, handling two separate interactions : the first with Apache jMeter, for the execution of a portlet instance on a CSGF portal, and the second with a CSGF's User Tracking Database, in order to check that the application has been really submitted to CSGF Engine. It's worth noting that this module is completely transparent to the portlet being submitted, which is defined by the jMeter input file (*.jmx*). Here a possible definition of the Nagios command

```
define command {

command_name check_sg-hostname-seq
command_line $USER2$/NagiosCheckSGApp.py --critical 75 --warning 25
--outfile $_SERVICEWEBLOG$ --jmx $_SERVICEJMX$
--jmx-log $_SERVICEJMXLOG$ --number-of-jobs 1 --utdb-param $_SERVICEDBCONNPARAMS$
--utdb-classes-prefix $_SERVICELIBRARYPATH$
}
```

as can be seen, many inputs are actually defined as service macros. Outfile is a log file for the check, which is eventually exposed by Nagios as extra note. With `--number-of-jobs` (shortly `-n`) it can be specified how many time to submit the request; in order to compute critical and warning rates, will be counted number of successful submissions over total submissions. Here two different service definitions : some values are common across different service instance, for instance path to User Tracking DB client lib, while others vary slightly according with the actual service instance being monitored.

```
define service{

use generic-service
host_name recasgateway.cloud.ba.infn.it
service_description Hostname Sequential
check_interval 120
notification_interval 240
check_command check_sg-hostname-seq
servicegroups Science Gateway Applications
_WEBLOG /usr/local/nagios/share/results/SG-RECAS-BA-hostname-seq.txt
_JMX /usr/local/nagios/myplugins/SG-App-Checks/jmx/SG-Bari-HostnameSeq.jmx
_LIBRARYPATH /usr/local/nagios/myplugins/SG-App-Checks/javajlibs
_DBCONNPARAMS /usr/local/nagios/myplugins/SG-App-Checks/etc/SG-Bari-UsersTrackingDBClient.cfg
_JMXLOG /usr/local/nagios/myplugins/SG-App-Checks/logs/SG-Bari-HostnameSeq.txt
notes_url https://sg-mon.ct.infn.it/nagios/results/SG-RECAS-BA-hostname-seq.txt
}

define service{
```

```

use generic-service
host_name sgw.africa-grid.org
service_description Hostname Sequential
check_interval 120
notification_interval 240
check_command check_sg-hostname-seq
servicegroups Science Gateway Applications
_WEBLOG /usr/local/nagios/share/results/SG-AfricaGrid-hostname-seq.txt
_JMX /usr/local/nagios/myplugins/SG-App-Checks/jmx/SG-AfricaGrid-HostnameSeq.jmx
_LIBRARYPATH /usr/local/nagios/myplugins/SG-App-Checks/javalibs
_DBCONNPARAMS /usr/local/nagios/myplugins/SG-App-Checks/etc/SG-AfricaGrid-UsersTrackingDBC
_JMXLOG /usr/local/nagios/myplugins/SG-App-Checks/logs/SG-AfricaGrid-HostnameSeq.txt
notes_url https://sg-mon.ct.infn.it/nagios/results/SG-AfricaGrid-hostname-seq.txt
}

```

## 11.5.2 eTokenServer

This module takes as input:

- a list of eToken urls
- file where to stream check's output
- warning and critical thresholds, computed as rate of successes contacting given urls.

A possible way to define the command for Nagios:

```

define command {

command_name check_etokenserver
command_line $USER2$/NagiosCheckeTokenServer.py
--urlfile /usr/local/nagios/var/check_sandbox/check_etokenserver/etokenserverurls.txt
--outputfile /usr/local/nagios/share/results/etokenserver.txt
--warning 10 --critical 20

}

```

## 11.5.3 OAR Login

This module is used to simulate login to an Open Access Repository. In order to simulate the interaction with the web site, it is used Apache jMeter; login information, as username, password and endpoint are inserted in the jmx file given in input to the module. The other input parameters accepted by the module are

- path to the output file (which is eventually exposed by Nagios supporting troubleshooting )
- property file for jMeter
- jMeter log file
- size of the test (number of attempts)
- critical and warning thresholds (expressed as a fraction of successful attempts over number of attempts)

The path to the jMeter binary, is set within the module to */usr/local/apache-jmeter-2.9/bin*, and can be changed replacing assigning a value to *jMeterPrefix* variable in *runJMeter* call. Here an example of the Nagios command for this check:

```
define command {

command_name check_oar-login
command_line $USER2$/NagiosCheckOARLogin.py
--critical 50 --warning 25
--outfile $_SERVICEWEBLOG$
--jmx $_SERVICEJMX$
--jmx-log $_SERVICEJMXLOG$
--number-of-users 2
}
```

in this case, several parameters are defined as service macros:

```
define service{

 use generic-service
 host_name www.openaccessrepository.it
 service_description Login
 check_interval 15
 notification_interval 240
 check_command check_oar-login
 servicegroups Semantic and Open Data
 _WEBLOG /usr/local/nagios/share/results/openaccessrepository-login.txt
 _JMX /usr/local/nagios/myplugins/OpenAccessRepo/jmx/openaccessrepo-login.jmx
 _JMXLOG /usr/local/nagios/myplugins/OpenAccessRepo/logs/openaccessrepo-login.log
 notes_url https://sg-mon.ct.infn.it/nagios/results/openaccessrepository-login.txt
}
```

## 11.5.4 Virtuoso

Beside built-in plugins, two modules have been developed for Virtuoso, checking service availability either submitting explicitly a SPARQL query, or contacting the REST interface with proper keyword parameters. Endpoint change slightly

```
define command {

command_name check_virtuoso_db
command_line $USER2$/NagiosCheckVirtuoso.py
--query $_SERVICEQUERYCOUNT$
--endpoint $_SERVICEENDPOINT$
--outputfile $_SERVICEWEBLOG$
--warning 0 --critical 15000000
}

define command {

command_name check_virtuoso_apiREST
command_line $USER2$/NagiosCheckVirtuosoREST.py
--keyword $_SERVICEKEYWORD$
--endpoint $_SERVICEENDPOINT$
--outputfile $_SERVICEWEBLOG$
--limit 10
--warning 5
}
```



```
--critical 0
}
```

as with OAR, several parameters are defined as service macros

```
define service{

use generic-service
host_name virtuoso
service_description Number of records in the semantic DB
check_interval 10
notification_interval 720
check_command check_virtuoso_db
_QUERYCOUNT "select count(?s) where {?s rdf:type <http://semanticweb.org/ontologies/2013/2/7/Rep
_ENDPOINT "http://virtuoso.ct.infn.it:8896/chain-reds-kb/sparql"
_WEBLOG "/usr/local/nagios/share/results/virtuosoDB.txt"
servicegroups Semantic and Open Data
}

define service{

use generic-service
host_name virtuoso
service_description API REST functionality
check_interval 10
notification_interval 720
check_command check_virtuoso_apiREST
_KEYWORD "eye"
_ENDPOINT "http://www.chain-project.eu/virtuoso/api/simpleResources"
_WEBLOG /usr/local/nagios/share/results/virtuosoAPI.txt
notes_url https://sg-mon.ct.infn.it/nagios/results/virtuosoAPI.txt
servicegroups Semantic and Open Data
}
```



---

## ABINIT WITH DATA MANAGEMENT

---

### 12.1 About



ABINIT is a package whose main program allows one to find the total energy, charge density and electronic structure of systems made of electrons and nuclei (molecules and periodic solids) within Density Functional Theory (DFT), using pseudopotentials and a planewave or wavelet basis.

ABINIT also includes options to optimize the geometry according to the DFT forces and stresses, or to perform molecular dynamics simulations using these forces, or to generate dynamical matrices, Born effective charges, and dielectric tensors, based on Density-Functional Perturbation Theory, and many more properties.

Excited states can be computed within the Many-Body Perturbation Theory (the GW approximation and the Bethe-Salpeter equation), and Time-Dependent Density Functional Theory (for molecules). In addition to the main ABINIT code, different utility programs are provided.

ABINIT is a project that favours development and collaboration ([short presentation of the ABINIT project](#)).

### 12.2 Installation

To install the abinitDM portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

**WebDAV:** The EMI-3 DPM Grid Storage Element, with WebDAV interface, to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRID-Support e-Infrastructure.

**Portlet Settings**

? Enabled \* ☒

? Infrastructure \* EUMED

? VName \* eumed

? TopBDII \* ldap://top-bdii.c4.csir.co.za:2170

? WMS Endpoint \* wms://wms.ulakbim.gov.tr:7443/glite\_wms\_wmproxy\_serv

? WMS Endpoint \* N/A

? MyProxyServer \* myproxy.ct.infn.it

? eTokenServer \* etokenserver2.ct.infn.it

? Port \* 8082

? Serial Number \* bc681e2bd4c3ace2a4c54907ea0c379b

? WebDAV \* prod-se-03.ct.infn.it

? Role

? Proxy Renewal ☒

? Disable VOMS ☐

« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Log Level:** The log level for the application (e.g.: *INFO* or *VERBOSE*);

**Metadata Host:** The Metadata hostname where download/upload digital-assets (e.g. *glibrary.ct.infn.it*);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the [CHAIN-REDS](#) Science Gateway.

Portlet Settings

?

 AppID \*

134

?

 Log Level \*

VERBOSE

?

 Metadata Host \*

glibrary.ct.infn.it

?

 Output Path \*

/tmp

?

 SoftwareTAG

ABINIT-7.6.4-sl5-x86\_64-gccany,ABINIT-7.6.4-sl6-x86\_64

?

 Proxy Path

?

 HostName

?

 UserName

?

 Password

?

 SMTP

smtp.gmail.com

?

 Sender

credentials-admin@ct.infn.it



« Prev | 1 | 2 | 3 | 4 | 5 | 6 | Z | Next »

## 12.3 Usage

The run abinit simulation the user has to click on the third accordion, select the type of job to run (e.g. ‘*Sequential*’ or ‘*Parallel*’) and upload the input files.

The **ABINIT** input files consist of:

- An input file (.in);
- A list of Pseudo Potential files;
- A file of files (.files).

For demonstrative use cases, the user can also click on ‘*Run demo*’ check-box and execute **ABINIT** with some pre-configured inputs.

Each run will produce:

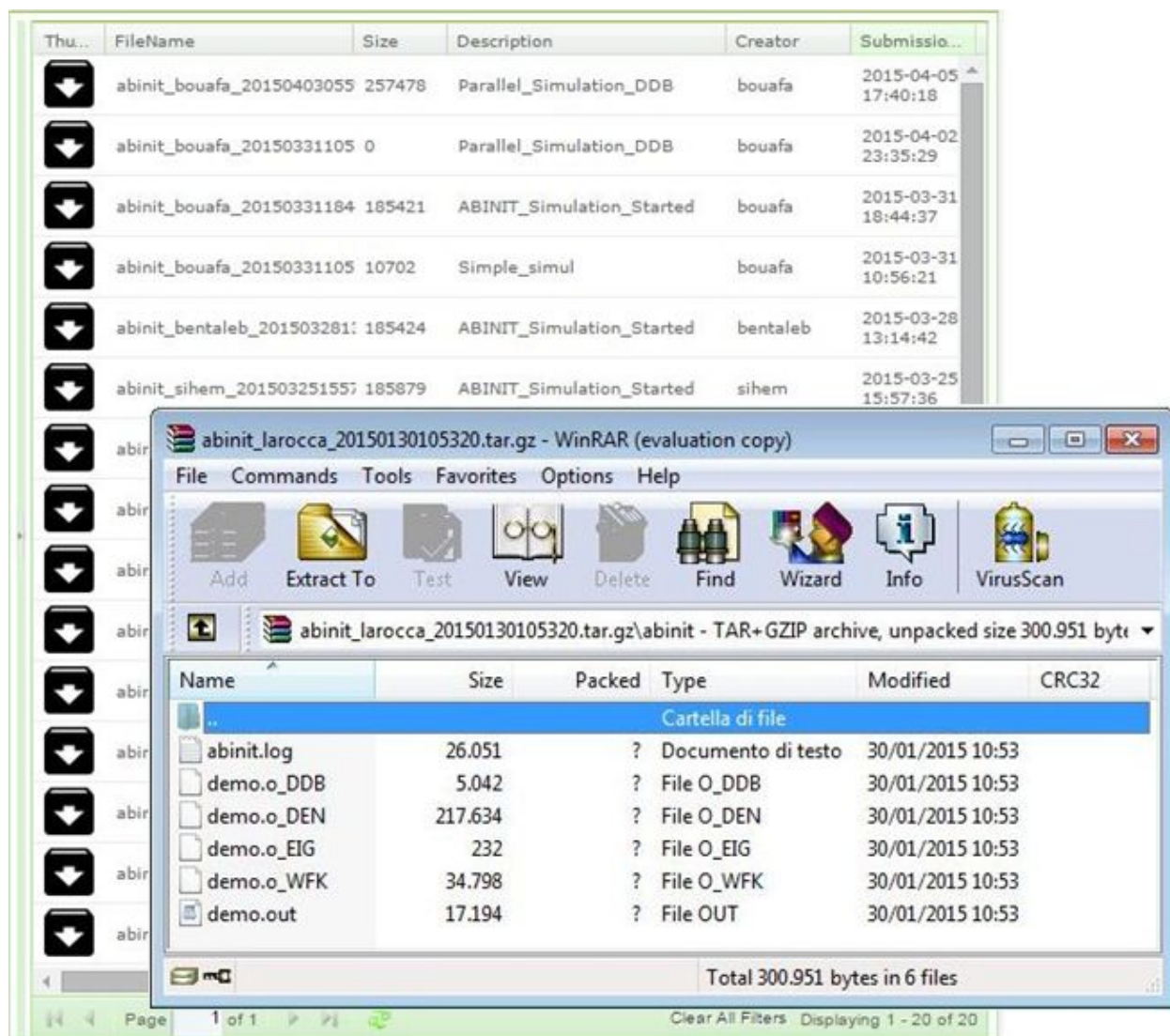
- std.txt: the standard output file;
- std.err: the standard error file;
- abinit.log: the application log file;
- some additional log files. By default, only the std OUT/ERR files will be provided;
- .tar.gz: the application results available through the **gLibrary** Metadata Server.

A typical simulation produces, at the end, the following files:

```
]$ tree ABINITSimulationStarted_147780/
ABINITSimulationStarted_147780/
-- abinit.log
-- curl.log
-- env.log
-- std.err
-- std.txt
```

To inspect **ABINIT** log files:

- *navigate* the digital repository for the application clicking [ [here](#) ];
- *select* the digital assets of any interest for downloading as shown in the figure below:



## 12.4 References

- e-AGE 2014 - “*International Connectivity of the Pan Arab Network*” - December 10-11, 2014 – Muscat, Oman [1];
- CHAIN-REDS Conference: “*Open Science at the Global Scale: Sharing e-Infrastructures, Sharing Knowledge, Sharing Progress*” – March 31, 2015 – Brussels, Belgium [2];

## 12.5 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics (INFN),

Mario TORRISI - University of Catania (DFA),

Brahim LAGOUN,

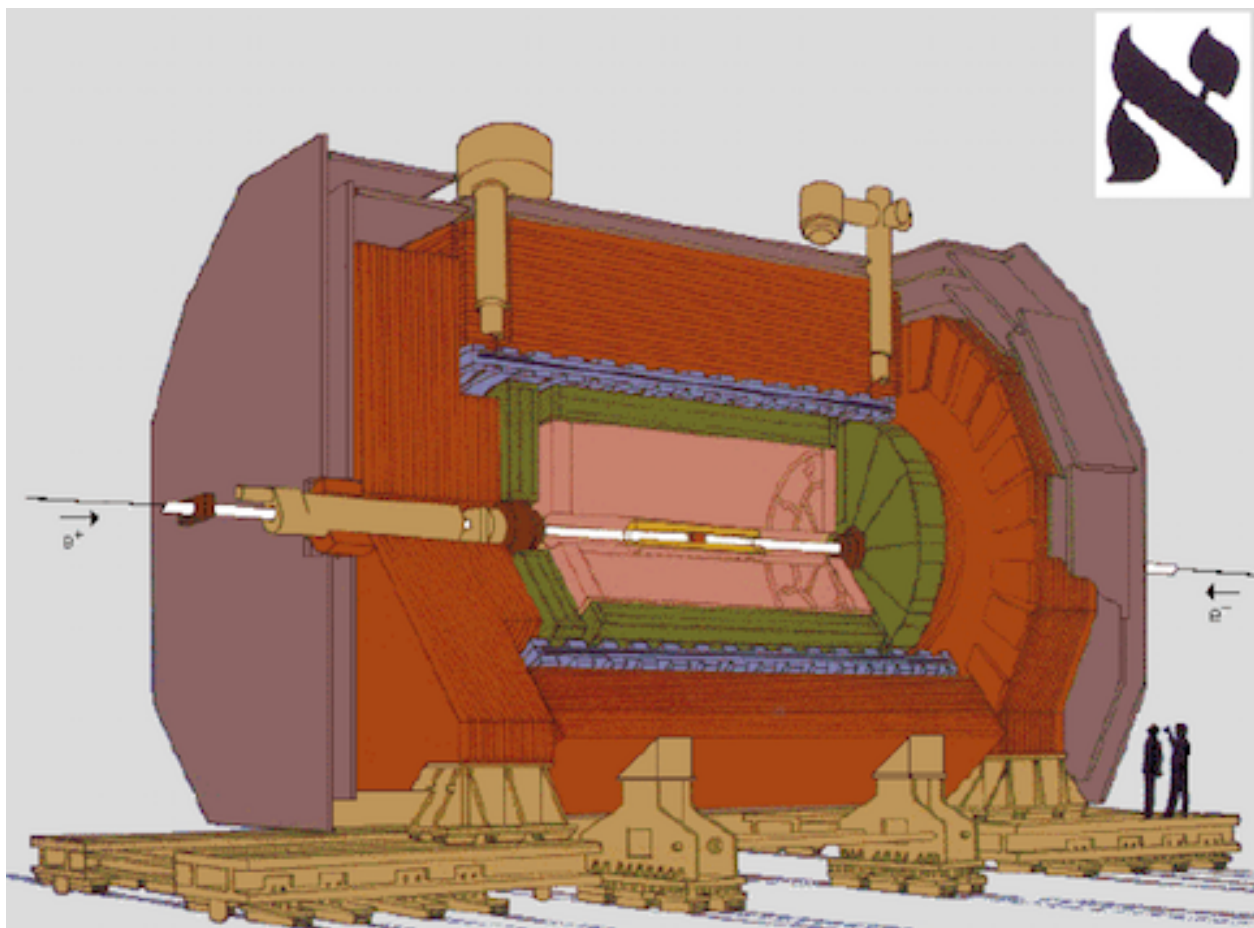
Ouafa BENTALB - Algerian Research Network ([ARN](#))



---

**ALEPH ANALYSIS**

---

**13.1 About**

ALEPH was a particle physics experiment installed on the Large Electron-Positron collider (LEP) at the CERN laboratory in Geneva/Switzerland. It was designed to explore the physics predicted by the Standard Model and to search for physics beyond it. ALEPH first measured events in LEP in July 1989. LEP operated at around 91 GeV – the predicted optimum energy for the formation of the Z particle. From 1995 to 2000 the accelerator operated at energies up to 200 GeV, above the threshold for producing pairs of W particles. The data taken, consisted of millions of events recorded

by the ALEPH detector, allowed precision tests of the electro-weak Standard Model (SM) to be undertaken. The group here concentrated our analysis efforts mainly in Heavy Flavour (beauty and charm) physics, in searches for the Higgs boson, the particles postulated to generate particle mass, and for physics beyond the SM, e.g. Supersymmetry, and in W physics. This application perform the search for the production and non-standard decay of a scalar Higgs boson into four tau leptons through the intermediation of the neutral pseudo-scalars Higgs particle. The analysis was conducted by the ALEPH collaboration with the data collected at centre-of-mass energies from 183 to 209 GeV.

## 13.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure several settings into the portlet preferences pane. Preferences have the form of a set of couples (key,value). Preferences are also grouped accordingly to the service configured. The meaning of each preference key will be described below, grouping them as well:

### aleph portlet configuration

Pref. Name	Pref. Value
GridOperation	136
cloudMgrHost	null
proxyFile	null

#### 13.2.1 General settings

**Grid Operation** Value used by the GridEngine to register user activity on the DCI

**cloudMgrHost** Unused

**proxyFile** Unused

#### 13.2.2 eTokenServer

Following settings are related to the eTokenServer service wich is the responsible to deliver proxy certificates from robot Certificates

**eTokenHost** Server hostname that issues Robot proxy certificates

**eTokenPort** Server port that issues Robot proxy certificates

**eTokenMd5Sum** The MD5 Sum specifies which specific Robot Certificate will be used to create the proxy certificate

**eTokenVO** VO name for the proxy certificate (VOMS) extension

**eTokenVOGroup** VOMS Group requested

**eTokenProxyRenewal** proxy certificate proxy renewal flag

**alephGroupName** unused

### 13.2.3 Guacamole

Aleph uses Guacamole service to obtain VNC and SSH connections available from the portal

**guacamole\_dir** Guacamole service server path

**guacamole\_noauthxml** path to the Guacamole noauthxml file

**guacamole\_page** base page for Guacamole

### 13.2.4 iServices

iServices is a new GridEngine helper service that manages the interactive services, its allocation status, lifetime, etc.

**iservices\_dbname** iservices database name

**iservices\_dbhost** iservices database host

**iservices\_dbport** iservices database port

**iservices\_dbuser** iservices database user

**iservices\_dbpass** iservices database password

**iservices\_srvname** iservices interactive service name

### 13.2.5 cloudProvider

cloudProvider is a new GridEngine helper service that maintains the necessary configuration to allocate new services on the cloud

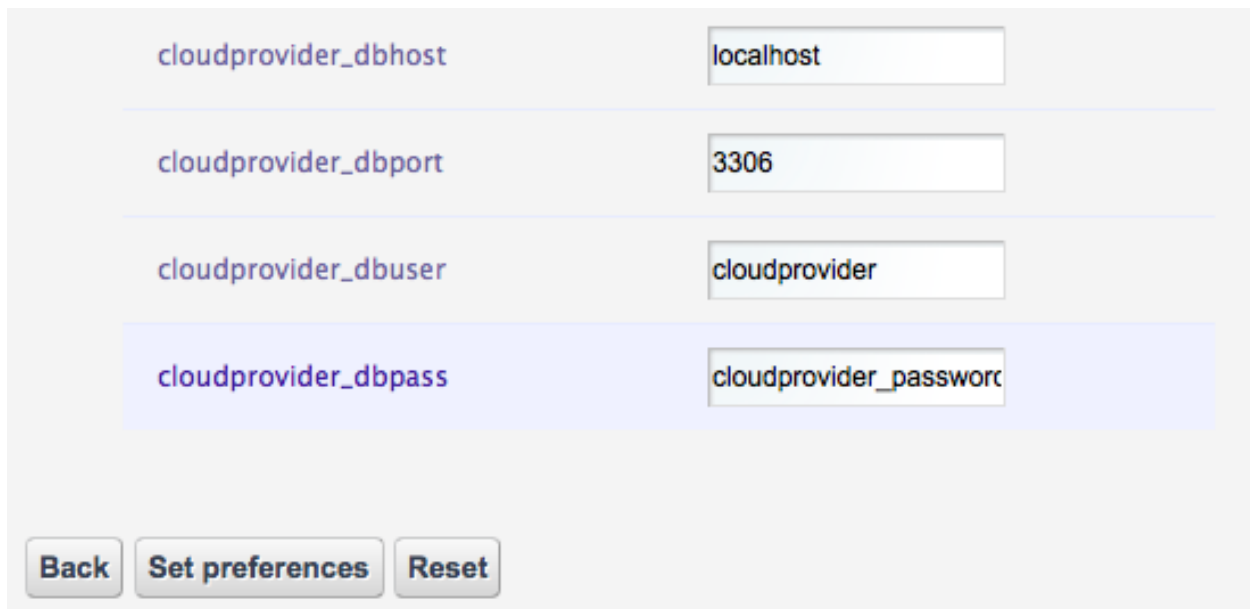
**cloudprovider\_dbname** cloud provider database name

**cloudprovider\_dbhost** cloud provider database host

**cloudprovider\_dbport** cloud provider database port

**cloudprovider\_dbuser** cloudprovider database user

**cloudprovider\_dbpass** cloudprovider database password



A screenshot of a web form for configuring database settings. It contains four rows, each with a label on the left and a text input field on the right. The labels are 'cloudprovider\_dbhost', 'cloudprovider\_dbport', 'cloudprovider\_dbuser', and 'cloudprovider\_dbpass'. The input fields contain 'localhost', '3306', 'cloudprovider', and 'cloudprovider\_password' respectively. The last row is highlighted with a light blue background. Below the form are three buttons: 'Back', 'Set preferences', and 'Reset'.

cloudprovider_dbhost	localhost
cloudprovider_dbport	3306
cloudprovider_dbuser	cloudprovider
cloudprovider_dbpass	cloudprovider_password

**Back** **Set preferences** **Reset**

The buttons represented by the picture above are representing

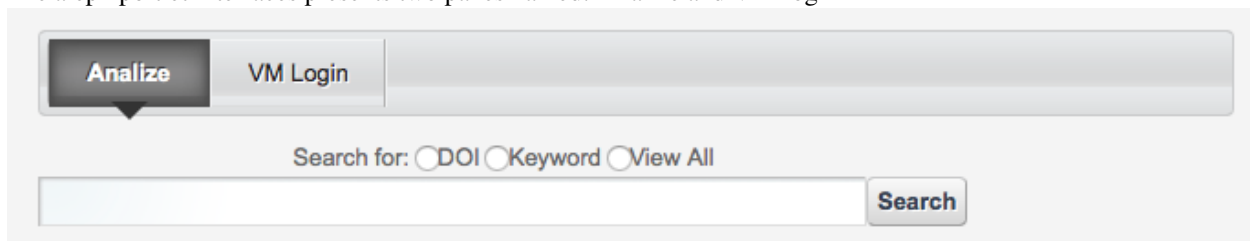
**Back** Return to the portlet

**Set Preferences** Apply changes to the preferences

**Reset** Reset default portlet settings as configured inside the portlet.xml file

## 13.3 Usage

The aleph portlet interfaces presents two panes named: Analyze and VMLogin



A screenshot of the Aleph portlet interface. It features a top navigation bar with two buttons: 'Analyze' (which is highlighted with a dark background and a downward-pointing triangle) and 'VM Login'. Below the navigation bar is a search section with the text 'Search for:' followed by three radio buttons labeled 'DOI', 'Keyword', and 'View All'. Below the radio buttons is a text input field and a 'Search' button.

**Analyze** VM Login

Search for: ☐ DOI ☐ Keyword ☐ View All

**Search**

### 13.3.1 Analyze

In the Analyze section the user can retrieve one of the available experiment file selecting a DOI number, a keyword or listing the whole content. Once obtained the list of the available files, the user can replicate the analysis just selecting the algorithm from a list e pressing the 'Analyze' button

Analyze

VM Login

Search for: ☐ DOI ☐ Keyword ☒ View All

Search

Show

10

entries

Search:

	Title	Type	Analisis	Action
	LEP Data from The ALEPH Collaboration year 2000 datafile ZD4000.1.AL	6LEP		Analyze
	LEP Data from The ALEPH Collaboration year 2000 datafile ZD4000.11.AL	6LEP		Analyze

### 13.3.2 VM Login

In this section the user can obtain the access to a Virtual Machine hosting the whole environment in order to extend the analysis introducing new algorithms or new analisis files.

Analyze

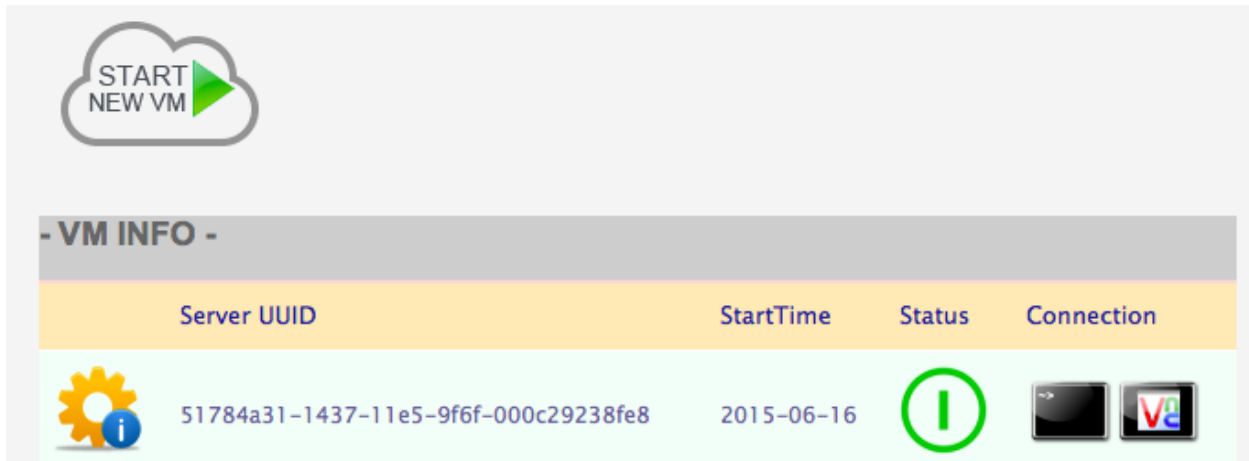
VM Login





ALEPH was a particle physics experiment installed on the Large Electron-Positron collider (LEP) at the CERN laboratory in Geneva

By clicking on 'Start new VM' button you will create a new ALEPH virtual machine that you can access for about 72 hours. You will be notified via e-mail about the necessary credentials to access the newly instatiated VM. You cannot start more than one ALEPH instance until the ALEPH VM expires.

START NEW VM

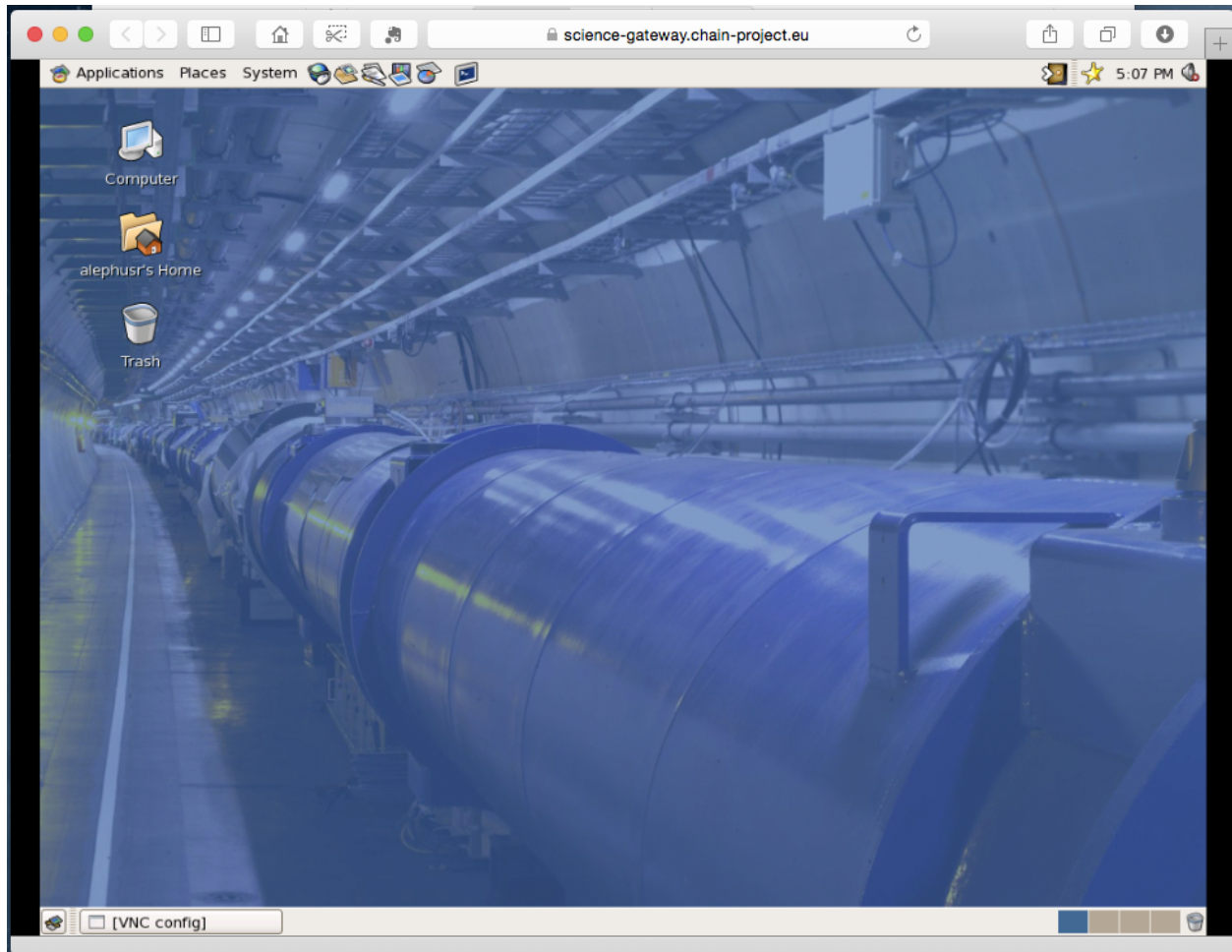
Pressing the 'Start VM' button, a new virtual machine will be started and associated to the user.



Server UUID	StartTime	Status	Connection
 51784a31-1437-11e5-9f6f-000c29238fe8	2015-06-16		 

Once available the VM, two image buttons representing a console and the VNC logo inside a monitor, allow respectively to connect the VM to an SSH console or into a VNC session from the portal. In any case the information about how to connect the VM will be sent to the user via email including the necessary credentials.

```
Last login: Fri Nov 21 17:59:22 2014 from gridsrv-03.roma3.infn.it
#
ALEPH2000-2014-001
#
[localhost] ~ $
```



## 13.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),  
Riccardo BRUNO - Italian National Institute of Nuclear Physics (INFN),  
Rita RICCERI - Italian National Institute of Nuclear Physics (INFN),  
Carla CARRUBBA - Italian National Institute of Nuclear Physics (INFN),  
Giuseppina INSERRA - Italian National Institute of Nuclear Physics (INFN),

### 13.4.1 WARNING for developers

Aleph portlet represents a new way to develop portlets for science gateways, deprecating the classic template mi-hostname-portlet.





---

**ALICE ANALYSIS**

---

**14.1 About**

---

ALICE (A Large Ion Collider Experiment) is a heavy-ion detector designed to study the physics of strongly interacting matter at extreme energy densities, where a phase of matter called quark-gluon plasma forms. The ALICE collabora-

---

tion uses the 10,000-tons ALICE detector – 26 m long, 16 m high, and 16 m wide – to study quark-gluon plasma. The detector sits in a vast cavern 56 m below ground close to the village of St Genis-Pouilly in France, receiving beams from the LHC. As of today, the ALICE Collaboration counts 1,550 members from 151 institutes of 37 countries. In this demonstrative application, you are able to select some ALICE datasets and run some analyses of the transverse momentum of charged particles in Pb-Pb collisions.

## 14.2 Installation

To install this portlet the WAR file has to be deployed into the application server. In the application server must be installed Guacamole service to obtain VNC and SSH connections available from the portal.

## 14.3 Usage

The alice portlet interfaces presents two panes named: Analyze and VMLogin

The screenshot displays the 'Analyze' pane of the ALICE portlet. At the top, there are two tabs: 'Analyze' (which is active and highlighted with a dark background) and 'VM Login'. Below the tabs, there are two sub-tabs: 'RAA2' (active) and 'PT ANALYSIS'. The main content area of the 'RAA2' sub-tab contains a welcome message: 'Welcome to RAA Analysis!' followed by the instruction: 'In this frame you should set the centrality range for RAA analysis, then start the analysis using the start button below.' Below this text, there are two input fields: 'Minimum Centrality: 0' and 'Maximum Centrality: 5', both with up and down arrows for adjustment. At the bottom center of this section is a button labeled 'Start Analysis'.

### 14.3.1 Analyze

In the Analyze section the user can do or a RAA Experiment or a PT Analysis. In the “RAA2” the user can choose the minimum and maximum centrality and replicate the analysis of the a dataset ALICE pressing the ‘Start Analysis’ Button. In the “PT Analysis” the user can choose the dataset to use (pp or PbPb) and the number of files to be processed, then start the analysis pressing the ‘Start Analysis’ Button.

Analyze

VM Login

RAA2

PT ANALYSIS

Welcome to PT Analysis!

In this frame you should select the dataset to use (pp or PbPb) and the number of files to be processed, then start the analysis using the start button below.

---

Type Analysis

pp 3.5 TeV

Number of files

1

Start Analysis

### 14.3.2 VM Login

In this section the user can obtain the access to an ALICE Virtual Machine hosting the whole environment in order to extend the analysis introducing new algorithms or new analysis files.

Analyze

VM Login

ALICE is the acronym for A Large Ion Collider Experiment, the experiment at the Large Hadron Collider

By clicking on 'Start new VM' button you will create a new ALICE virtual machine that you can access for about 72 hours. You will be notified via e-mail about the necessary credentials to access the newly instantiated VM. You cannot start more than one ALICE instance until the ALICE VM expires.

START  
NEW VM

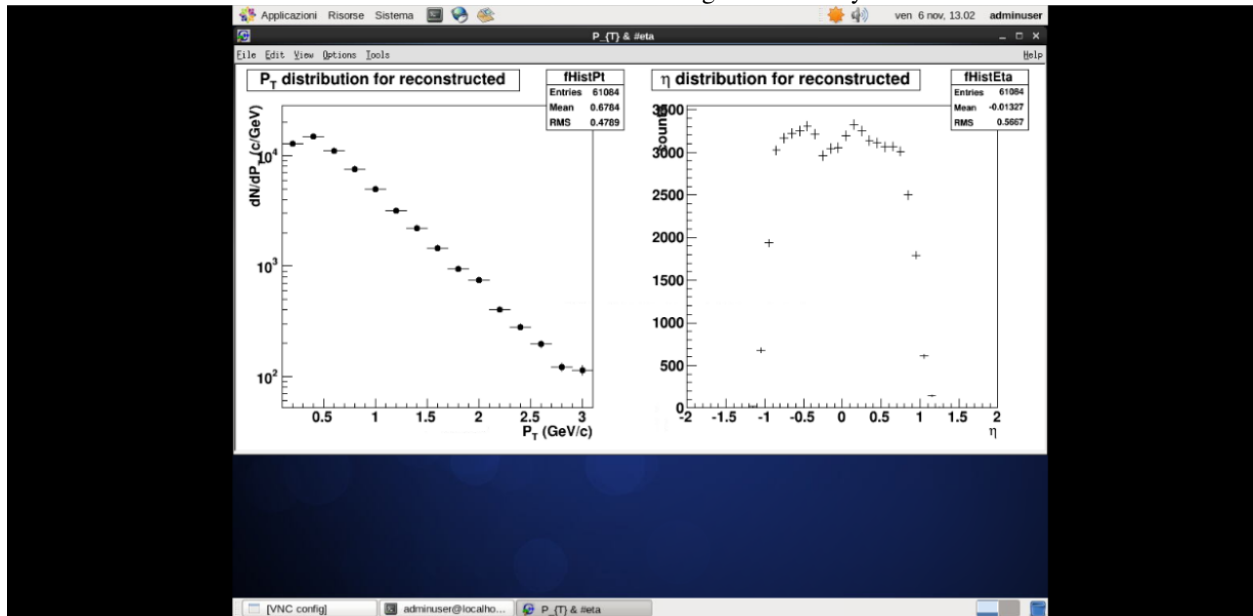
- VM INFO -

Server UUID	StartTime	Status	Connection
-------------	-----------	--------	------------

Pressing the 'Start VM' button, a new virtual machine will be started and associated to the user.



Once available the VM, two image buttons representing a console and the VNC logo inside a monitor, allow respectively to connect the VM to an SSH console or into a VNC session from the portal. In any case the information about how to connect the VM will be sent to the user via email including the necessary credentials.



## 14.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),  
 Riccardo BRUNO - Italian National Institute of Nuclear Physics (INFN),  
 Rita RICCERI - Italian National Institute of Nuclear Physics (INFN),  
 Carla CARRUBBA - Italian National Institute of Nuclear Physics (INFN),  
 Giuseppina INSERRA - Italian National Institute of Nuclear Physics (INFN),

## 15.1 About



---

The **ASTRA** project aims to reconstruct the sound or timbre of ancient instruments (not existing anymore) using archaeological data as fragments from excavations, written descriptions, pictures, etc.

The technique used is the Physical Modeling Synthesis (**PMS**), a complex digital audio rendering technique which allows modeling the time-domain physics of the instrument. In other words the basic idea is to recreate a model of the musical instrument and produce the sound by simulating its behavior as a mechanical system.

The application would produce one or more sounds corresponding to different configurations of the instrument (i.e. the different notes). The project runs since 2006 thanks to the GEANT backbone and to computing resources located in Europe and in other regions of the world, allowing researchers, musicians and historians to collaborate, communicate and share experiences on lost instruments and sounds ASTRA brings again to life.

## 15.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRID-Support e-Infrastructure.

**Portlet Settings**

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	EUMED
VName *	eumed
TopBDII *	ldap://bdii.eumedgrid.eu:2170
WMS Endpoint *	wms://wms.ulakbim.gov.tr:7443/glite_wms_wmproxy_server
	N/A
MyProxyServer *	myproxy.ct.infn.it
eTokenServer *	etokenserver.ct.infn.it
Port *	8082
Serial Number *	bc681e2bd4c3ace2a4c54907ea0c379b
Role	
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the [CHAIN-REDS](#) Science Gateway.

Portlet Settings

?

AppID\*

114

?

Output Path\*

/tmp

?

SoftwareTAG

POVRAY-3.6

?

HostName

?

UserName

?

Password

?

SMTP

smtp.gmail.com

?

Sender

credentials-admin@ct.infn.it



« Prev

1

2

3

4

5

6

7

8

Next »

## 15.3 Usage

To run a simulation with [ASTRA](#) the user has to:

- click on the *second* accordion of the portlet and,
- select some settings for the generation of the digital libraries as shown in the below figure:

2

Specify the Input Settings

Please, select the ASTRA input settings for your run

?

Min Param.\*

7

?

Max Param.\*

71

?

Min Velocity\*

10

?

Max Velocity\*

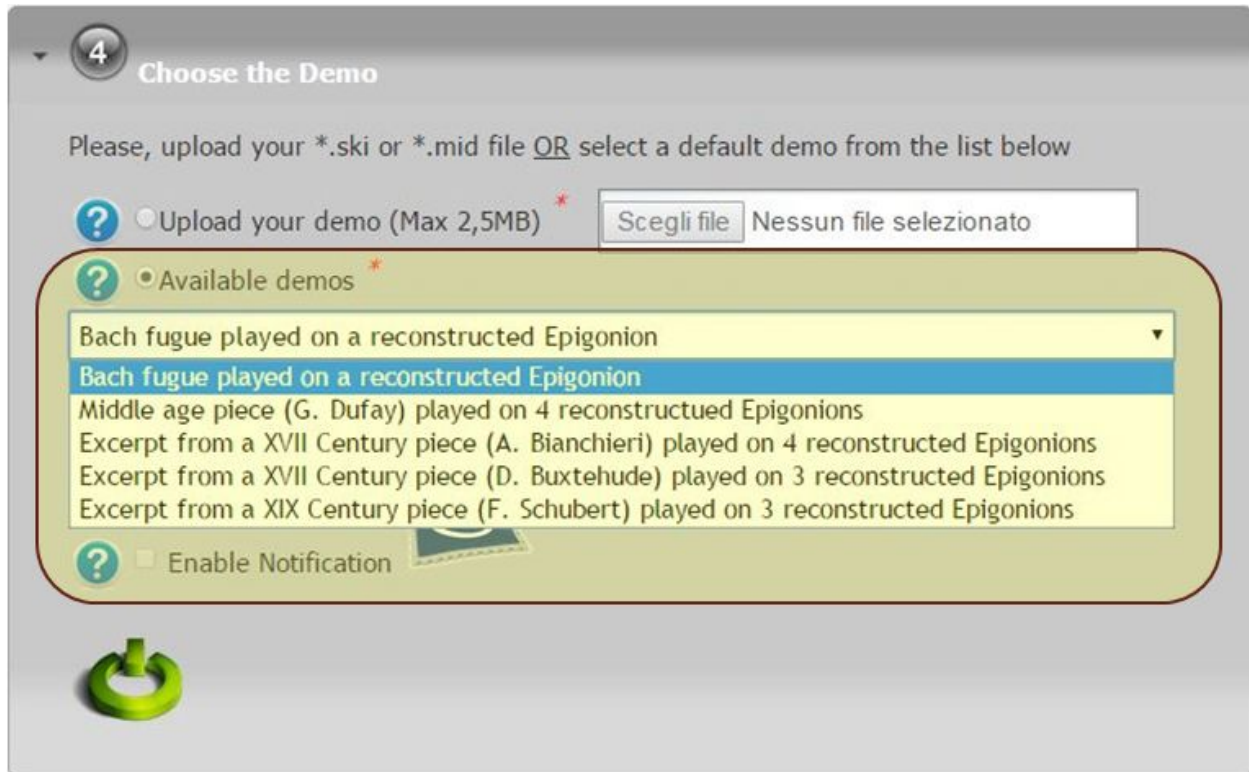
104

?

Step Velocity\*

3

- click on the *third* accordion of the portlet and,
- select the input file (e.g. *.ski* or *.mid* files) OR select a demo from the list as shown in the below figure:



Each simulation will produce:

- *std.txt*: the standard output file;
- *std.err*: the standard error file;
- *.wav*: a MIDI file about an opera played using the [Epigonion](#);
- *.tar.gz*: containing the sound libraries of each singular string of the [Epigonion](#).

A typical simulation produces, at the end, the following files:

```
]$ tree ASTRASoundTimbreReconstructionSimulationStarted_148681/
ASTRASoundTimbreReconstructionSimulationStarted_148681/
-- AstraStk.err
-- AstraStk.out
-- bachfugue.wav (8.7M)
-- output.README
-- samples.tar.gz (589M)
```

## 15.4 References

- Final workshop of Grid Projects “Pon Ricerca 2000-2006, Avviso 1575”: “*ASTRA Project Achievements: The reconstructed Greek Epigonion with GILDA/ASTRA brings history to life. It takes archaeological findings of extinct musical instruments, and lets us play them again thanks to a virtual digital model running on the GRID.EUMEDGRID on GEANT2/EUMEDCONNECT*” – February 10-12, 2009 Catania, Italy [1];
- Conferenza GARR: “*ASTRA Project: un ponte fra Arte/Musica e Scienza/Tecnologia - Conferenza GARR*” – September 2009, Napoli, Italy [2];



- International Symposium on Grid Computing 2009: “*The ASTRA (Ancient instruments Sound/Timbre Reconstruction Application) Project brings history to life*” – March 2010, Taipei, Taiwan [3];
- Proceedings of the International Conference on Computational Science, ICCS2010, doi:10.1016/j.procs.2010.04.043: “*Data sonification of volcano seismograms and Sound/Timbre reconstruction of ancient musical instruments with Grid infrastructures*” – May, 2010 Amsterdam, The Netherlands [4];

## 15.5 Contributors

Please feel free to contact us any time if you have any questions or comments.

**Authors** Salvatore AVANZO - *Responsible for Development Activities*,

Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),

Francesco DE MATTIA - Universidad de Málaga ([MALAGA](#)),

Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#)),

Mariapaola SORRENTINO - Conservatory of Music of Avellino ([\[5\]](#)),

Domenico VICINANZA - DANTE ([DANTE](#))



## 16.1 About

With this service it is possible to execute scientific applications on Virtual Machines (VMs) deployed on standard-based federated cloud.

The present service is based on the following standards and software frameworks:



## 16.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the CHAIN\_REDS Cloud Testbed from the project Science Gateway [1].

**Portlet Settings**

- Enabled \*** ☒
- Infrastructure \*** CHAIN-REDS
- VName \*** vo.chain-project.eu
- TopBDII**
- Cloud Resource \***
  - rocci://nebula-server-01.ct.infn.it:9000/
  - rocci://stack-server-01.ct.infn.it:8787/
  - rocci://carach5.ics.muni.cz:11443/
  - rocci://prisma-cloud.ba.infn.it:8787/
  - rocci://one01.ciemat.es:11443/
  - rocci://rocci.grid.arn.dz:11443/
  - N/A
- MyProxyServer \*** myproxy.ct.infn.it
- eTokenServer \*** etokenserver2.ct.infn.it
- Port \*** 8082
- Serial Number \*** bc681e2bd4c3ace2a4c54907ea0c379b
- Role** vo.chain-project.eu
- Proxy Renewal** ☒
- Disable VOMS** ☐

« Prev | 1 | 2 | **3** | 4 | 5 | 6 | 7 | 8 | Next »

In the following figure is shown how the portlet has been configured to run simulation on the EGI Federated Cloud Infrastructure [2] from the project Science Gateway [1].

**Portlet Settings**

? Enabled \* ☒

? Infrastructure \* FEDCLOUD

? VOname \* fedcloud.egi.eu

? TopBDII ldap://egee-bdii.cnaf.infn.it:2170

rocci://carach5.ics.muni.cz:11443/ -

rocci://cloud.cesga.es:3202/ -

rocci://occi.cloud.gwdg.de:3100/ -

rocci://oceanos-occi2.hellasgrid.gr:9000/ -

N/A + -

? Cloud Resource \*

? MyProxyServer \* etokenserver.ct.infn.it

? eTokenServer \* etokenserver.ct.infn.it

? Port \* 8082

? Serial Number \* bc779e33367eaad7882b9dfaa83a432c

? Role fedcloud.egi.eu

? Proxy Renewal ☒

? Disable VOMS ☐

« Prev | 1 | 2 | 3 | **4** | 5 | 6 | 7 | 8 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;


**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the [CHAIN\\_REDS](#) Science Gateway [1].

**Portlet Settings**

? AppID *	130
? Log Level *	VERBOSE
? Output Path *	/tmp
? SoftwareTAG	
? HostName	
? UserName	
? Password	
? SMTP	smtp.gmail.com
? Sender	credentials-admin@ct.infn.it





« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | Next »

## 16.3 Usage


To run the simulations the user has to:


- click on the *third* accordion of the portlet,
- select the VM to run on the available Cloud Testbed as shown in the below figure:


**3 Specify the VM Settings** 


 Please, use the drop-down list to choose the VM profile you wish to deploy



**[ VM Settings ]**


 Virtual Server \*


 VM Flavor \*


 VM Name \*

 Description


 ☐ Notification 

 Advanced Settings ►

 Configuring advanced settings (if any) for the selected VM


 ☐ Upload your macro ASCII file (\*.r) for R (Max 2,5MB) \*

Nessun file selezionato

 ☒ Insert here the macro to be processed \*

```
This is a simple R macro
Create a bivariate normal distribution

x <- rnorm(n = 10000, mean = 10, sd = 4)
y <- rnorm(n = 10000, mean = 10, sd = 4)
```



Each simulation will produce:

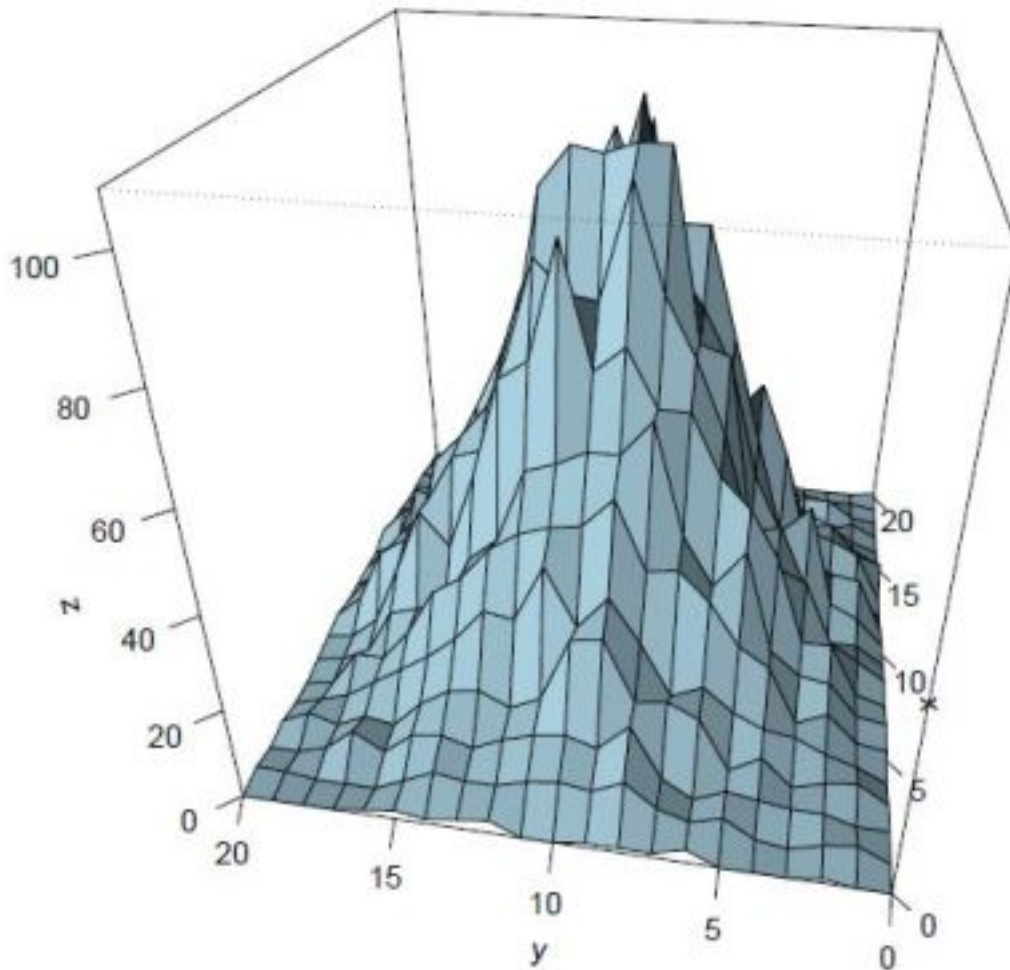
- *std.out*: the standard output file;
- *std.err*: the standard error file;
- *.tar.gz*: the output results.

A typical simulation produces, at the end, the following files:

```
]$ tree Pleaseinsertthereyourdescription_148684/
Pleaseinsertthereyourdescription_148684/
-- results.tar.gz
-- std.err
-- std.out
```

The list of files produced during the run are the following:

```
]$ tar ztvf results.tar.gz
output.README
Rplots.pdf
```



## 16.4 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#))



---

## CLUSTALW

---



---

### 17.1 About

ClustalW2 [1] is a widely used multiple program for multiple alignment of nucleic acid and protein sequences. sequence alignment computer program.

The program accepts a wide range on input formats including: NBRF/PIR, FASTA, EMBL/Swissprot, Clustal, GCC/MSF, GCG9 RSF, and GDE, and executes the following workflow:

- Pairwise alignment;
- Creation of a phylogenetic tree (or use a user-defined tree);
- Use of the phylogenetic tree to carry out a multiple alignment

Users can align the sequences using the default setting but occasionally it may be useful to customize one's own parameters. The main parameters are the gap opening penalty and the gap extension penalty.

For more information:

- Consult the official documentation [2];
- Clustal W and Clustal X version 2.0 [3];
- The MP4 file is a video [4] showing how to use the ClustalW from the Africa Grid Science Gateway [5]

### 17.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRIDSupport e-Infrastructure [6].

**Portlet Settings**

? Enabled *	<input checked="" type="checkbox"/>
? Infrastructure *	EUMED
? VOName *	eumed
? TopBDII *	ldap://bdii.eumedgrid.eu:2170
? WMS Endpoint *	wms://wms.ulakbim.gov.tr:7443/glite_wms_wmproxy_server
? MyProxyServer *	etokenserver.ct.infn.it
? eTokenServer *	etokenserver.ct.infn.it
? Port *	8082
? Serial Number *	bc681e2bd4c3ace2a4c54907ea0c379b
? Role	
? Proxy Renewal	<input checked="" type="checkbox"/>
? Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;


**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the Africa Grid Science Gateway [5].

Portlet Settings

AppID *	164
Log Level *	VERBOSE
Output Path *	/tmp
SoftwareTAG	CLUSTALW2
HostName	
UserName	
Password	
SMTP	smtp.gmail.com
Sender	credentials-admin@ct.infn.it



« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | Next »

## 17.3 Usage

To perform the Multi Sequence Alignment for DNA or protein the user has to:

- click on the *third* accordion of the portlet,
- choose the sequence type (e.g.: *DNA* or *protein*),
- upload the sequence as ASCII file OR use the default one pre-configured clicking in the textarea below,
- configured additional settings if needed as shown in the below figure:

**3 Sequences & Options**

Please, enter the sequence you want to analyze in the text-area below, OR upload a file

Choose the type of your sequences, between Protein

Upload your input file (Max 2,5MB)

Scegli file Nessun file selezionato

Insert here the sequence to be analyzed

```
>seq0
FQTWEEFSRAAEKLYLADPMKVRVVLKYRHVDGNLCIKVTDDLVCVLYRTDQAQDVKK
IEKF
>seq1
KYRTWEEFTRAEEKLYQADPMKVRVVLKYRHCDGNLCIKVTDDVCLLYRTDQAQDVK
```

Alignment type ☒ Slow, but accurate ☐ Fast, but approximate

Description

Please, insert here a description for your job

☐ Notification

**Step 2 - Slow Pairwise Alignment Options**

The default settings will fulfill the needs of most users.  
Configure advanced settings for the selected alignment options (if needed).

Protein Weight Matrix Gonnet

GAP OPEN 10

GAP EXTENTION 0.1

**Set 3 - Set up your Multiple Sequence Alignment Options**

**Set 4 - Set up your Advanced Output Options**

Each simulation will produce:

- *std.out*: the standard output file;
- *std.err*: the standard error file;
- *.tar.gz*: containing the results of the Monte Carlo simulation.

A typical simulation produces, at the end, the following files:

```
]$ tree SequenceAlignmentSimulationStarted_126163/
SequenceAlignmentSimulationStarted_126163/
-- std.err
-- std.out
-- output.README
-- outputs.tar.gz

]$ tar zxvf outputs.tar.gz
20150601120928_larocca.aln
20150601120928_larocca.dnd
```

## 17.4 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),

Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#))



---

## CORSIKA-LAGO

---

### 18.1 About



LAGO is an international project from more than 80 scientists of 8 latinamerican countries started in 2005 (complete list of the collaboration members and their institutions). The LAGO project aims at observing Gamma Ray Bursts (GRBs) by the single particle technique using water Cherenkov detectors (WCD). It consists of various sites at high altitude, in order to reach a good sensitivity to the faint signal expected from the high energy photons from GRBs.

CORSIKA (COsmic Ray SIMulations for KAScade) is a program for detailed simulation of extensive air showers initiated by high energy cosmic ray particles. Protons, light nuclei up to iron, photons, and many other particles may be treated as primaries.

### 18.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure the preference settings into the portlet preferences pane.

Preferences are splitted in three separate parts: Generic, Infrastructures and the application execution setting. The generic part contains the **Log level** which contains one of following values, sorted by decreasing level: info, debug, warning and error.

The **Application Identifier** refers to theId field value of the GridEngine 'UsersTracking'database table: GridInteractions. The infrastructure part consists of different settings related to the destination of users job execution. The fields belonging to this category are:

**Enable infrastructure:** A true/false flag which enables or disable the current infrastructure;

**Infrastructure Name:** The infrastructure name for these settings;

**Infrastructure Acronym:** A short name representing the infrastructure;

**BDII host:** The Infrastructure information system endpoint (URL). Infrastructure preferences have been thought initially for the elite Grid based infrastructures;

**WMS host:** It is possible to specify which is the brokering service endpoint (URL);

**Robot Proxy values:** This is a collection of several values which configures the robot proxy settings (Host, Port, proxyID, VO, Role, proxy renewal);

**Job requirements:** This field contains the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc.

Infrastructure preferences	
Enable infrastructure	yes
Infrastructure number	2
Infrastructure name	The EUMEDGrid Infrastructure
Infrastructure acronym	EUMEDGRID
BDII Host	ldap://bdii.eumedgrid.eu:2170
WMS Hosts	
Proxy Robot host server	etokenserver.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	true
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	eumed
Proxy Robot VO Role	eumed
Proxy Robot Renewal Flag	true
Local Proxy	null
Software Tags	

< - + >

Application job requirements

Application pilot job pilot\_script.sh

UsersTrackingDB Settings	
UserTrackingDB Hostname	localhost
UserTrackingDB Username	tracking_user
UserTrackingDB Password	usertracking
UserTrackingDB Database	usertracking

Set preferences Reset Back to application

Actually, depending on the infrastructure, some of the fields above have an overloaded meaning. Please contact the support for further information or watch existing production portlet settings.

## 18.3 Usage

To run the Corsika simulation the user has to:




- Select the type of simulation to perform, corresponding to one of the three Corsika versions: single, array or epos-thining-qsj2.
- Choose whether the input file corresponds to a single simulation or to a group of them. In the later, files must be compressed on tar.gz format.
- Choose whether to receive a confirmation email at the end of the execution-
- Choose what to do with the output files. If a Storage Element is desired, files will be uploaded and remain there for a long time. If not, they will be stored in the Science Gateway, although their lifetime will probably be shorter.
- Last but not least, choose an input file or insert a PID where the input file is stored.
- The system will automatically create a name for the run. If a different one is desired, it can be freely modified.

Each run will produce:

- std.txt: the standard output file;
- std.err: the standard error file;
- abinit.log: the application log file;
- some additional log files. By default, only the std OUT/ERR files will be provided;
- .tar.gz: the application results available through the **gLibrary** Metadata Server.

**CORSIKA-LAGO**



**CORSIKA, LAGO VERSION**  
 this portlet is devoted to execute Corsika in remote Grid infrastructures.  
 Pressing the '**Reset**' Button all input fields will be initialized.  
 Pressing the '**About**' Button information about the application will be shown

**Corsika version**

LAGO single ▼

**Is it a parametric job?**  
 If so, remember that input file must be a .tar.gz with a single folder containing all the inputs, and that folder must ONLY contain inputs  
☒ No ☐ Yes

**Do you want to receive an email alert when the execution has finished?**  
☒ No ☐ Yes

**Do you want to store the results in a remote Storage Element?**  
☒ No ☐ Yes

**Input data**  
 There are two options: local input file, or PID of a remote input file.

Local input file:  No file chosen

PID of remote input file:

**job identifier**  
 Choose a name to identify your job in this page (by default it is set to the name of the input file).

A typical simulation produces, at the end, the following files:

```

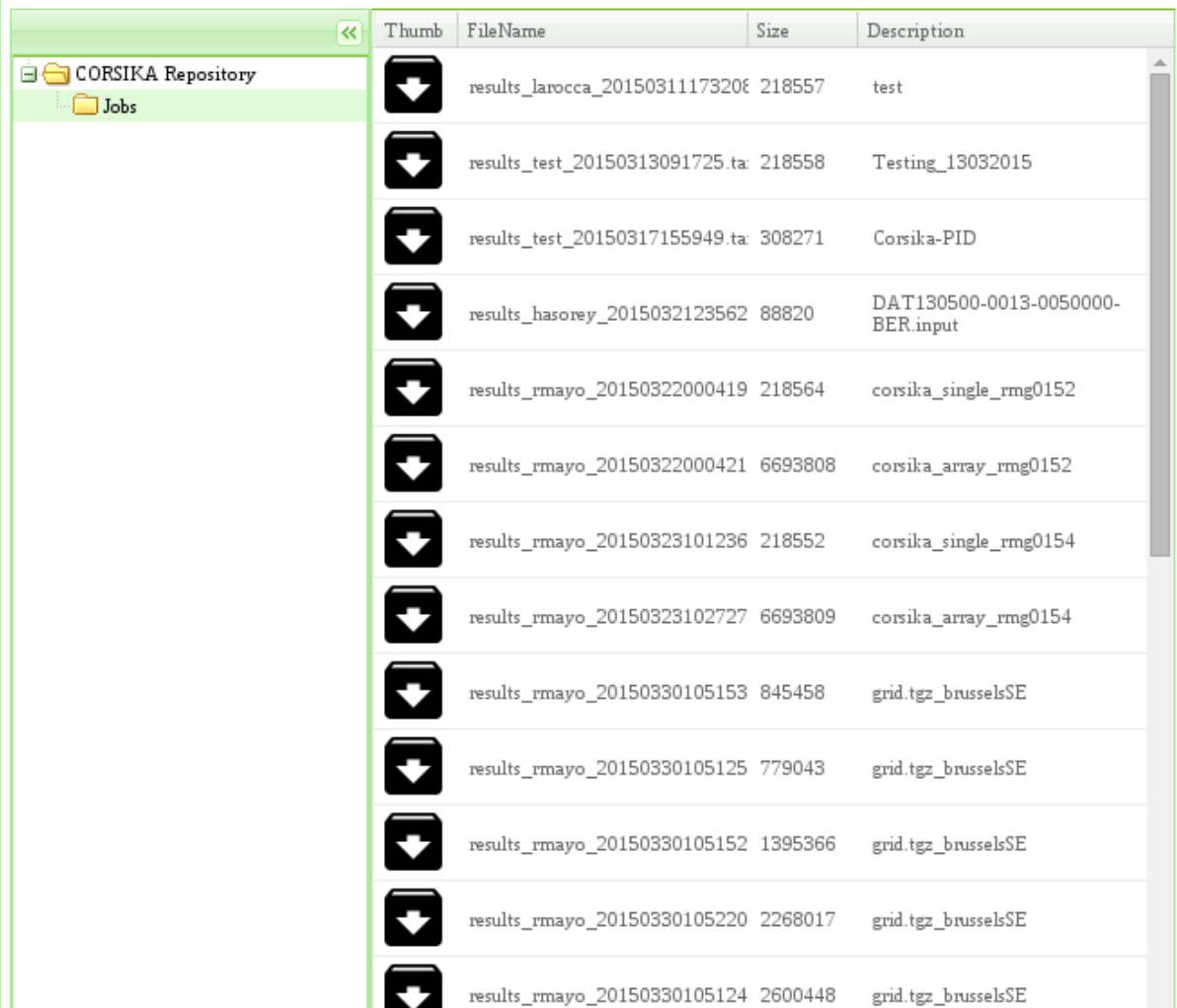
]$ tree 26_144903
26_144903
|-- corsika-Error.txt
|-- corsika-Output.txt
`-- results
 |-- DAT030014
 |-- DAT030014-0014-0527176.input
 `-- DAT030014.dbase

```

To inspect Corsika log files:

- *navigate* the digital repository for the application clicking [ [here](#) ];
- *select* the digital assets of any interest for downloading as shown in the figure below:

**CORSIKA Repository**



The screenshot displays the CORSIKA Repository interface. On the left, a sidebar shows a tree view with 'CORSIKA Repository' and 'Jobs' folders. The main area is a table with columns: Thumb, FileName, Size, and Description. Each row represents a digital asset with a download icon in the 'Thumb' column.

Thumb	FileName	Size	Description
	results_larocca_20150311173208	218557	test
	results_test_20150313091725.ta	218558	Testing_13032015
	results_test_20150317155949.ta	308271	Corsika-PID
	results_hasorey_2015032123562	88820	DAT130500-0013-0050000-BER.input
	results_rmayo_20150322000419	218564	corsika_single_rmg0152
	results_rmayo_20150322000421	6693808	corsika_array_rmg0152
	results_rmayo_20150323101236	218552	corsika_single_rmg0154
	results_rmayo_20150323102727	6693809	corsika_array_rmg0154
	results_rmayo_20150330105153	845458	grid.tgz_brusselsSE
	results_rmayo_20150330105125	779043	grid.tgz_brusselsSE
	results_rmayo_20150330105152	1395366	grid.tgz_brusselsSE
	results_rmayo_20150330105220	2268017	grid.tgz_brusselsSE
	results_rmayo_20150330105124	2600448	grid.tgz_brusselsSE

## 18.4 References

- CHAIN-REDS Conference: “*Open Science at the Global Scale: Sharing e-Infrastructures, Sharing Knowledge, Sharing Progress*” – March 31, 2015 – Brussels, Belgium [1];

## 18.5 Contributors

Please feel free to contact us any time if you have any questions or comments.

**Authors** Manuel RODRIGUEZ-PASCUAL - CIEMAT Sci-Track

Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics (INFN)



## GLIBRARY-REPO-BROWSER-PORTLET

This portlet allow to browse digital repositories using [gLibrary](#), the Digital Repository System developed by [INFN](#).

### 19.1 Installation

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure some settings:

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to navigate EUMEDGrid-Support digital repositories.

**Portlet Settings**

Enabled*	<input checked="" type="checkbox"/>
Infrastructure*	EUMED
VOName*	eumed
eTokenServer*	etokenserver.ct.infn.it
Port*	8082
Serial Number*	bc681e2bd4c3ace2a4c54907ea0c379b
Role	

« Prev | **1** | 2 | Next »

2.) To configure the application, the following settings have to be provided:

**Proxy:** The proxy used to contact [gLibrary](#);

**Repository:** The digital repository to browse;

**LAT:** The default latitude of the EMI-3 DPM Storage Element;

**LONG:** The default longitude of the EMI-3 DPM Storage Element.

In the figure below is shown how the portlet has been configured to browse the **ESArep** digital repository.

**Portlet Settings**

 Proxy*	<input type="text" value="glibrary"/>
 Repository*	<input type="text" value="ESArep"/>
 LAT*	<input type="text" value="37.525809"/>
 LONG*	<input type="text" value="15.073628"/>



« Prev | 1 | **2** | Next »







## 20.1 About



**GROMACS** is versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles.

It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the non-bonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

## 20.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

**WebDAV:** The EMI-3 DPM Grid Storage Element, with WebDAV interface, to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRID-Support e-Infrastructure.

**Portlet Settings**

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	EUMED
VName *	eumed
TopBDII *	ldap://top-bdii.c4.csir.co.za:2170
WMS Endpoint *	wms://wms.ulakbim.gov.tr:7443/glite_wms_wmproxy_server
MyProxyServer *	myproxy.ct.infn.it
eTokenServer *	etokenserver2.ct.infn.it
Port *	8082
Serial Number *	bc681e2bd4c3ace2a4c54907ea0c379b
WebDAV *	prod-se-03.ct.infn.it
Role	
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | **3** | 4 | 5 | 6 | 7 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Log Level:** The log level for the application (e.g.: *INFO* or *VERBOSE*);

**Metadata Host:** The Metadata hostname where download/upload digital-assets (e.g. *glibrary.ct.infn.it*);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the **CHAIN-REDS** Science Gateway.

Portlet Settings

?

 AppID \*

126

?

 Log Level \*

VERBOSE

?

 Metadata Host \*

glibrary.ct.infn.it

?

 Output Path \*

/tmp

?

 SoftwareTAG

GROMACS-4.6.5-sl5-x86\_64-gccany,GROMACS-4.6.5-sl6-x8

?

 Proxy Path

?

 HostName

?

 UserName

?

 Password

?

 SMTP

smtp.gmail.com

?

 Sender

credentials-admin@ct.infn.it



« Prev | 1 | 2 | 3 | 4 | 5 | 6 | Z | Next »

## 20.3 Usage

To run a molecular dynamic simulation with **GROMACS** the user has to:

- click on the third accordion of the portlet,
- select the **GROMACS** release to use (e.g. *v4.6.5* or *v5.0.4*),
- upload the input *macro* file (*.tpr*).

For demonstrative use cases, the user can also click on ‘*Run demo*’ check-box and execute a simulation with some pre-configured inputs.

Each molecular dynamic simulation will produce:

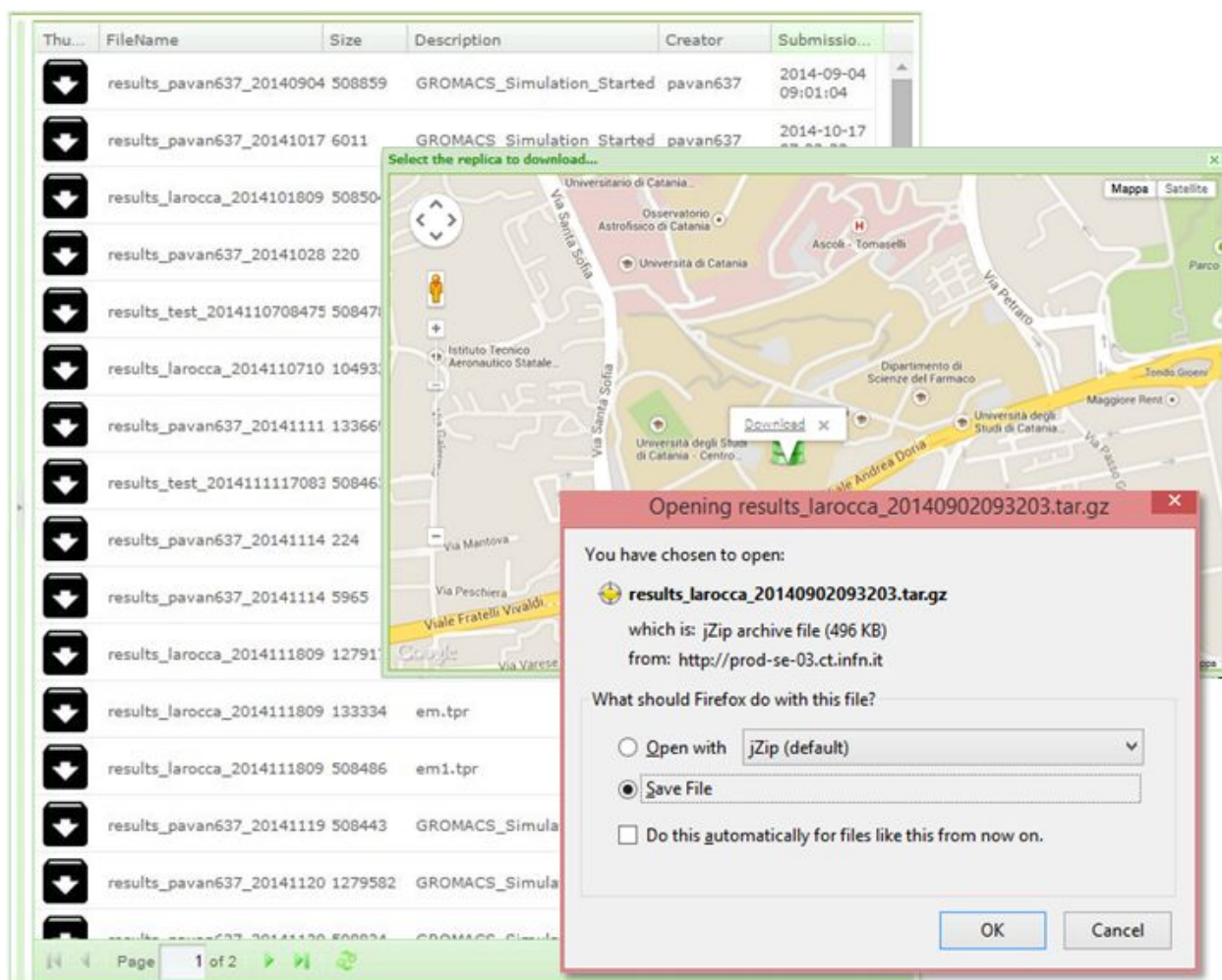
- *std.txt*: the standard output file;
- *std.err*: the standard error file;
- *gromacs.log*: the application log file;
- some additional log files;
- *.tar.gz*: the application results available through the **gLibrary** Metadata Server.

A typical simulation produces, at the end, the following files:

```
]$ tree GROMACSSimulationStarted_147118/
GROMACSSimulationStarted_147118/
-- curl.log
-- env.log
-- gromacs.log
-- output.README
-- std.err
-- std.txt
```

To inspect **GROMACS** log files:

- *navigate* the digital repository for the application clicking [ [here](#) ];
- *select* the digital assets of any interest for downloading as shown in the figure below:



## 20.4 References

- CHAIN-REDS Conference: “*Open Science at the Global Scale: Sharing e-Infrastructures, Sharing Knowledge, Sharing Progress*” – March 31, 2015 – Brussels, Belgium [1];

## 20.5 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics (INFN),

Mario TORRISI - University of Catania (DFA),

Sarath Kumar BASKARAN - Centre for Biotechnology, Anna University, Chennai (AUC)



---

## INFECTION MODEL PORTLET

---

### 21.1 About



The infection model is an example of an Agent-Based Simulation Infection Model implemented in the well-known **Repast Symphony** ([repast.sourceforge.net](http://repast.sourceforge.net)) agent-based simulation toolkit. Agent-based simulation is a highly useful technique that allows individuals and their behaviours to be represented as they interact over time. This means, with appropriate data, agent-based simulation can be used to study various socio-medical phenomena such as the spread of disease and infection in a population.

The aim of this demonstration model is to show how a science gateway could support the study of the spread of disease or infection in a population. As well as having direct healthcare application, it can also be used in the field of health economics to study the cost effectiveness of various infection preventive strategies.

Within the science gateway, the Repast Infection Model has been deployed in a portlet named `infectionModel-portlet`. This has been developed to enable users to submit experiments with different input parameters and to obtain results. As well as the results output file, the application also has a demonstration graph tool that allows users to see the graphical visualisation of the results.

This shows that science gateways can be developed to support online complex simulations in an extremely easy to use manner. See the [Sci-GaIA project](#) web pages and the [educational modules](#) to get information on how to implement these applications as well as how science gateways and data repositories can be used to support Open Science.

### 21.2 Installation

This section explains how to deploy and configure the `infectionModel-portlet` into a Science gateway to submit some preconfigures experiments towards Distributed Computing infrastructures.

1. Move into your Liferay plugin SDK portlets folder and clone the `infectionModel-portlet` source through the following git command:

```
git clone https://github.com/csgf/infectionModel-portlet.git
```

2. Now, move into the just created `infectionModel-portlet` directory and execute the deploy command:

```
ant deploy
```



When the previous command has completed, verify that the portlet is “*Successfully autodeployed*”, look for a string like this in the Liferay log file under `$LIFERAY_HOME/glassfish-3.1.2/domains/domain1/logs/server.log`.

3. Then, open your browser and point at your Science Gateway instance and from there click Add > More in the Brunel University category, click on Add button to add this new portlet. Following picture shows the correctly result:



The aim of this demonstration model is to show how a science gateway could support the study of the spread of disease or infections in a population. The **Repast-infection-model** is implemented in **Repast Symphony**. The aim of this example model is to study the behaviour of infections with an annual outbreak according to several input parameters.

The parameters are:

**Simulation Period** - specifies how many years the simulation will run for.

**Recovered Count** - specifies the initial healthy population. Healthy population have immunity and cannot be infected immediately. However, after a number of contacts with infected population, they lose their immunity and become susceptible to infection.

**Infected Count** - specifies the initial infected population. Infected population can infect susceptible population upon contacting them. They recover after a period of time and become healthy.

**Susceptible Count** - specifies initial susceptible population. Susceptible population can be infected when contacted by infected population. We assume that if more than one susceptible agent is in the proximity of an infected agent, only one will be infected.

The output of the simulation is the amount of each population (Recovered, Infected and Susceptible) over time.

#### How to use the Infection Model

1. Select an experiment to run from the drop down box (in a full application users would be able to enter their own parameters).
2. Note the Simulation Identifier. You will need this to identify the results from the experiment that you are about to run.
3. Press **Submit**. This will submit the experiment to the computers of the e-Infrastructure that you are using.
4. Go to **MyWorkSpace** (menu bar above). Select **MyJobs**. Your current experiments will be listed here. When the job is complete then you can download by clicking on the icon on the right of the job.
5. Unzip your results file. The download is in tgz zipped format. Use a tool like 7-zip ([www.7-zip.com](http://www.7-zip.com)) to unzip the file.
6. Each results file will contain **output(timestamp).txt**. This contains how each population varies over time. To visualise it, select **Visualise** from the left hand side menu and upload the text file to produce a graph of population variance over time.

Please fill the following form and then press the ‘**SUBMIT**’ button to launch this application.

Requested inputs are:

Select experiment	Please select an experiment ▼
Simulation Parameters	
Simulation Period (years)	<input type="text"/>
Recovered Count	<input type="text"/>
Infected Count	<input type="text"/>
Susceptible Count	<input type="text"/>
Simulation identifier	Repast simulation...
<input type="button" value="SUBMIT"/> <input type="button" value="RESET"/>	

As soon as the portlet has been successfully deployed you have to configure:


1. the list of e-Infrastructures where the application can be executed;
2. some additional application settings.

To configure the e-Infrastructure, go to the portlet preferences and provide the the right values for the following parameters:



- **Enable infrastructure:** A yes/no flag which enables or disable the generic e-Infrastructure;
- **Infrastructure name:** A label representing the e-Infrastructure;
- **Infrastructure acronym:** The acronym to reference the e-Infrastructure;
- **BDII:** The Top BDII for this e-Infrastructure;
- **WMS Hosts:** A separated ; list of WMS endpoint for this e-Infrastructure;
- **Proxy Robot host server:** The eTokenServer for this e-Infrastructure;
- **Proxy Robot host port:** The eTokenServer port for this e-Infrastructure;
- **Proxy Robot secure connection:** A true/false flag if the eTokenServer require a secure connection;
- **Proxy Robot Identifier:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;
- **Proxy Robot Virtual Organization:** The VO for this e-Infrastructure;
- **Proxy Robot VO Role:** The VO role for this e-Infrastructure;
- **Proxy Robot Renewal Flag:** A true/false Flag to require proxy renewal before it expires;
- **Local Proxy:** The path to the proxy if you are using a local proxy;
- **Software Tags:** The list of software tags requested by the application.

The following figure shown how the portlet has been configured to run simulation on a cloud based system.



**Application preferences**  
 Edit the application preferences:  
☐ Click here to get help on configuration values

Pressing the **'Set preferences'** Save the new preferences  
 Pressing the **'Cancel'** No changes will be done to the current preferences  
 Pressing the **'Reset'** Button all input fields will be initialized.

**Generic preferences**

Log Level	info
Grid operation identifier	175
Infrastructure numbers	2
Grid operation description	myRepass-infection description

**Infrastructure preferences**

Enable infrastructure	yes
Infrastructure number	2
Infrastructure name	DFA OCCI
Infrastructure acronym	rocci
BDII Host	REPASS
WMS Hosts	rocci://stack-server-01.cloud.dfa.unict.it:8787
Proxy Robot host server	etokenserver2.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	true
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	vo.africa-grid.org
Proxy Robot VO Role	vo.africa-grid.org
Proxy Robot Renewal Flag	true
Local Proxy	null
Software Tags	2fe6035a-cdc8-4bba-bfc6-82cfa890ffc;repass/network/publ

< - + >

**Application job requirements**

Application pilot job	pilot_script.sh
-----------------------	-----------------

**UsersTrackingDB Settings**

UserTrackingDB Hostname	
UserTrackingDB Username	
UserTrackingDB Password	
UserTrackingDB Database	

Set preferences Reset Back to application

Another important step to have infectionModel-portlet ready to be used is: to create a new entry in GridOperations table of the UsersTracking database, as shown below.

```
INSERT INTO GridOperation VALUES ('<portal name>' , 'Infection Model portlet');

-- portal name: is a label representing the portal name, you can get the
-- right value from your Science Gateway instance.
```

## 21.3 Usage

The infectionModel-portlet, has been developed in the contest of the Sci-GaIA project, and it is currently available on the [Africa Grid Science Gateway](#). You can read more information on how to use this application, after sign in, on its dedicated [run page](#).

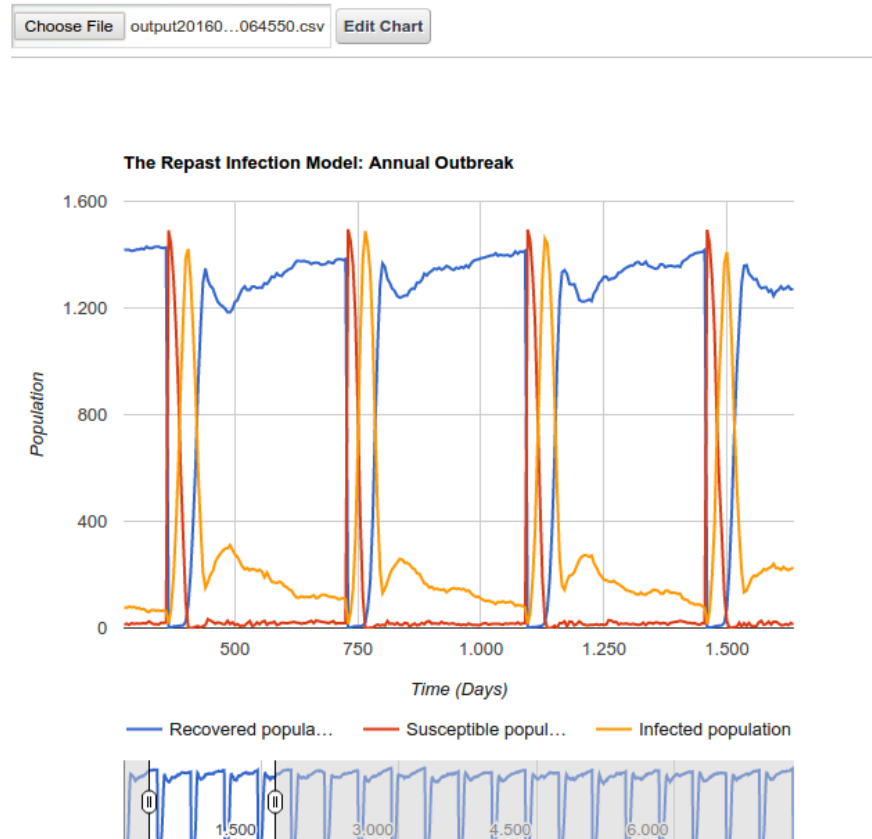
As soon as your submitted interaction complete its execution you can exploit the [Visualize infection Model result](#) portlet, to see the simulation outputs in a graphical way, like shown in the picture below.

When an authorised user successfully log on, they are presented with the portlet, i.e the infection model portlet, where

they can specify all the necessary input parameters of the infection model. After a user has finished specifying the parameters and clicked on the submit button, the jobs can then be submitted to the different Distributed Computing Infrastructures. However, due to limitation of resources, this portlet presents a version where a number of experiments have been fixed and users can only choose from within a predefined set of experiments. After submitting a job, users would be notified that their jobs have been successfully submitted and then advised to check the MyJobs portlet, a dedicated portlet where the status of all running jobs can be found. A done job status would be represented by a small folder icon and users can download the output of the infection model for analysis.

The analysis of the infection model result output file, using the visualize portlet, can be seen below:

This is an example of how a portlet can be used to visualise the output of the infection model. Select the appropriate `output(timestamp).csv` from your experiments and you will see the results.



## 21.4 Contributor(s)

If you have any questions or comments, please feel free to contact us using the Sci-GaIA project discussion forum ([discourse.sci-gaia.eu](http://discourse.sci-gaia.eu))

**Authors** Roberto BARBERA - University of Catania ([DFA](#)),  
 Adedeji FABIYI - Brunel University London ([BRUNEL](#)),  
 Simon TAYLOR - Brunel University London ([BRUNEL](#)),  
 Mario TORRISI - University of Catania ([DFA](#))



---

## INFECTION MODEL PARALLEL PORTLET

---

### 22.1 About



While the first version of the Infection Model portlet was executed sequentially, this version, the Infection Model parallel portlet, would be executed in parallel. This portlet will be used to investigate how an Agent Based modelling simulation experiments can be executed in parallel by making use of high performance computing facilities.

Similar to the Infection model portlet, it makes use of different input parameters to help users submit experiments and obtain results. These parameters include: input parameters for the model include the simulation period (specifies how many years the simulation will run), recovered count (specifies the initial healthy population), infected count (specifies the initial infected population) and susceptible count (specifies the initial susceptible population). When an infected agent approaches a susceptible agent, it becomes infected and if there are more than one susceptible agent in the cell, only one, randomly selected agent, is infected. Infected agents recover after a period and become healthy with a level of immunity. Recovered agents immunity decreases every time they are approached by an infected agent and when immunity becomes zero, the recovered agent becomes susceptible and can be infected again, thereby, forming a host of infection networks.

However, rather than running jobs sequentially, with single core machines, this version will run jobs with machines that have many cores running at different cloud sites.

### 22.2 Installation

This section explains how to deploy and configure the **infectionModel-parallel-portlet**.

1. Move into your Liferay plugin SDK portlets folder and clone the template-portlet source code through the git clone command:

```
git clone https://github.com/csgf/infectionModel-parallel.git
```

2. Now, move into the just created portlet directory and execute the deploy command:

```
ant deploy
```

When the previous command has completed, verify that the portlet was “*Successfully autodeployed*”, look for a string like this in the Liferay log file under `$LIFERAY_HOME/glassfish-3.1.2/domains/domain1/logs/server.log`.

3. Then, open your browser and point at your Science Gateway instance and from there click Add > More in the BRUNEL category, click on Add button to add this new portlet. Following picture shows the correctly result:

While our initial version of the Repast-infection-model was executed **sequentially**, this version is the **parallel** approach to executing the model, an **Agent-Based Simulation Infection Model** implemented in the well-known **Repast Symphony** agent-based simulation toolkit, on distributed systems.

Please insert the number of experiments you like to run in parallel.

**Application input**

Insert number of parallel jobs: (Required) 3

	Simulation Period	Recovered Count	Infected Count	Susceptible Count
Job 1:	20	0	20	1500
Job 2:	20	0	20	2000
Job 3:	20	0	10	2000

Job label

test\_20160606141929

As soon as the portlet has been successfully deployed you have to configure it using the portlet configuration menu. Portlet configuration is splitted in two parts: *Generic application preferences*, *Infrastructures preferences*.

### 22.2.1 Generic application preferences

The generic part contains:

- **Application Identifier** the identifier assigned to the application in the GridInteractions database table.
- **Application label** (*Required*) a short meaningful label for the application.
- **Production environment** a boolean flag that specifies if the portlet will be used in a production or development environment.
  - if *true* the development environment preferences will be shown
    - \* **UserTrackingDB hostname** hostname of the Grid and Cloud Engine Usertracking database. Usually *localhost*
    - \* **UserTrackingDB username** username of the Grid and Cloud Engine Usertracking database user. Usually *user\_tracking*
    - \* **UserTrackingDB password** password specified for the Usertracking database user. Usually *usertracking*
    - \* **UserTrackingDB database** Grid and Cloud Engine Usertracking database name. Usually *user-tracking*
- **Application requirements** the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc. defined using JDL format.

**Infection Model Parallel Portlet - Configuration**

Setup Permissions Sharing

Archive/Restore Setup

**Generic application preferences**

Save

**Application Identifier**  
82

**Application label (Required)**  
Infection Model Parallel

**Infrastructure number**  
3

☒ Production environment

**Application requirements**

Edit pilot script

**Available Infrastructures**

Add New

Infrastructure status successfully changed

Name	Infrastructure Middleware/Acronym	Status	
SSH@DFA	ssh		Actions
ROCCI@DFA	rocci		Actions
ROCCI@100%it	rocci		Actions

Showing 3 results.

**Note:** You can get the *Application Identifier* inserting a new entry into the **GridOperations** table:

```
INSERT INTO GridOperation VALUES ('<portal name>' , 'Template Portlet');
-- portal name: is a label representing the portal name, you can get the
-- right value from your Science Gateway instance.
```

## 22.2.2 Infrastructure preferences

The infrastructure preferences section shows the e-Infrastructures configured. Using the actions menu on the right side of the table, you can:

- Activate / Deactivate
- Edit
- Delete

an available infrastructure. The *Add New* button is meant to add a new infrastructure available to the application. When you click this button a new panel, will be shown with several fields where you can specify the Infrastructure details.

The fields belonging to this panel are:

- **Enabled** A boolean which enable or disable the current infrastructure.
- **Infrastructure Name** (*Required*) The infrastructure name for these settings.
- **Middleware** (*Required*) The middleware used by the current infrastructure. Here you can specify 3 different values.
  - **an acronym** for gLite based middleware.
  - **ssh** for HPC Cluster.
  - **rocci** for cloud based middleware.

Following fields will be translated in the relevant infrastructure parameters based on the value specified in this field.

- **BDII host:** The Infrastructure information system endpoint (URL).
  - If Middleware is **ssh** here you can specify a ";" separated string with ssh authentications parameters (username;password or username for key based authentication).
  - If Middleware is **rocci** here you can specify the name of the compute resource that will be created.
- **WMS host:** is the service endpoint (URL).
- **Robot Proxy host server:** the robot proxy server hostname.
- **Robot Proxy host port:** the robot proxy server port.
- **Proxy Robot secure connection:** a boolean to specify if robot proxy server needed a SSL connection.
- **Robot Proxy identifier:** the robot proxy identifier.
- **Proxy Robot Virtual Organization:** the virtual organization configured.
- **Proxy Robot VO Role:** the role virtual organization configured.
- **Proxy Robot Renewal Flag:** a boolean to specify if robot proxy can be renewed before its expiration.
- **RFC Proxy Robot:** a boolean to specify if robot proxy must be RFC.
  - If Middleware is **rocci** this field must be checked.
- **Local Proxy:** the path of a local proxy if you want use this type of authentication.
- **Software Tags:** infrastructure specific information.
  - If Middleware is **rocci** here you can specify a ";" separated string with  
`<image_id>;<flavor>;<link_resource>`



**Infection Model Parallel Portlet - Configuration**

Setup Permissions Sharing

[Archive/Restore Setup](#)

**Infrastructure preferences** [Back](#)

☒ Enabled

**Infrastructure name (Required)**  
ROCCI@DFA

**Middleware (Required)**  
rocci

**BDII Host**  
REPASt

**WMS Hosts**  
rocci://stack-server-01.cloud.dfa.unict.it:8787

**Proxy Robot host server**  
etokenserver2.ct.infn.it

**Proxy Robot host port**  
8082

☐ Proxy Robot secure connection

**Proxy Robot Identifier**  
bc681e2bd4c3ace2a4c54907ea0c379b

**Proxy Robot Virtual Organization**  
vo.africa-grid.org

**Proxy Robot VO Role**  
vo.africa-grid.org

☐ Proxy Robot Renewal Flag

☒ RFC Proxy Robot

**Local Proxy**

**Software Tags**  
3fb9e799-0d4f-41ab-ac7a-b6939f5b81c4;repast/network/public

Save Reset

## 22.3 Usage

Similar to the infection Model portlet, When an authorised user successfully log on, they are presented with the portlet, i.e the infection model-parallel portlet. However, this portlet only present an interface where users can specify the number of experiments they will like to execute in parallel. This is done by inserting the number of jobs in the “insert number of parallel jobs” field. After specifying the number of jobs, users can then click on the ok button and this will automatically generate and display the input fields for the different parameters of the infection model (i.e the recovered, susceptible and the infected population). Users can then specify their input parameters by using these fields. After a user has finished specifying the parameters and clicked on the submit button, the jobs can then be submitted to the different Distributed Computing Infrastructures. After submitting a job, users would be notified that their jobs have been successfully submitted and then advised to check the MyJobs portlet, a dedicated portlet where the status of all running jobs can be found. A job will be considered to be done when all the running jobs, which have been submitted in parallel, becomes done. A done job status would be represented by a small folder icon and users can download the output of the infection model for analysis.

## 22.4 Contributor(s)

If you have any questions or comments, please feel free to contact us using the Sci-GaIA project discussion forum ([discourse.sci-gaia.eu](https://discourse.sci-gaia.eu))

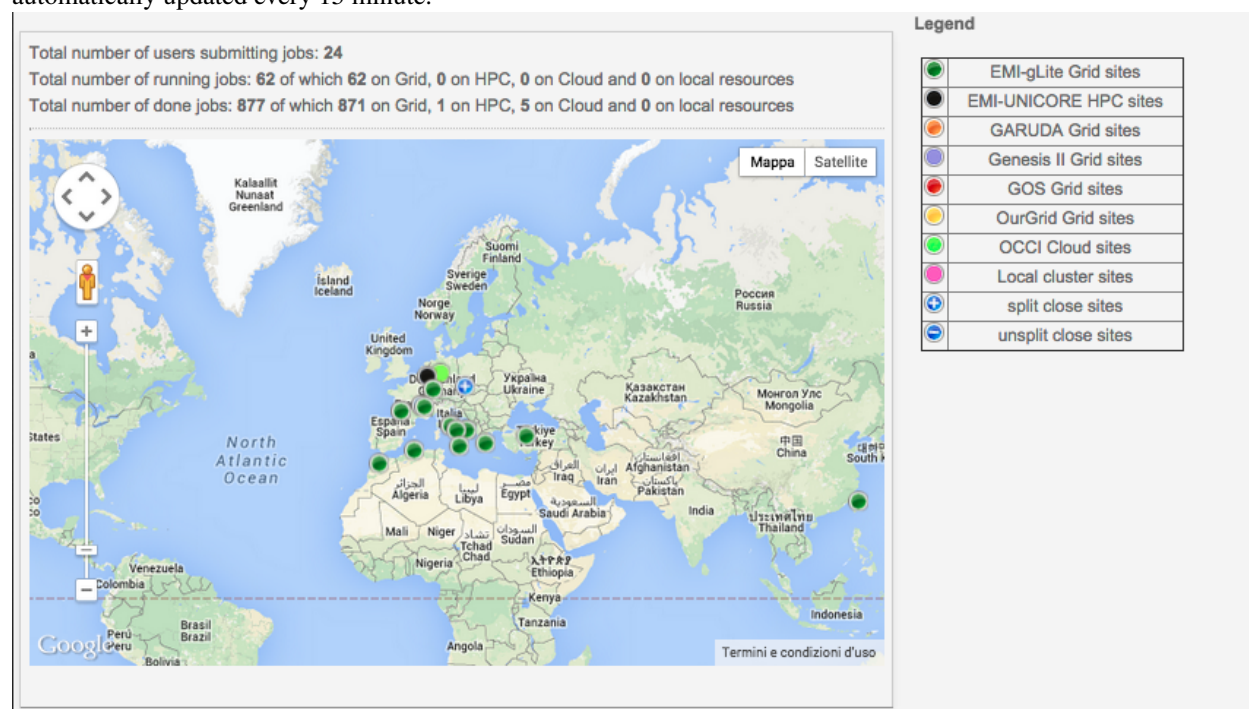
**Authors** Roberto BARBERA - University of Catania ([DFA](#)),  
 Adedeji FABIYI - Brunel University London ([BRUNEL](#)),  
 Simon TAYLOR - Brunel University London ([BRUNEL](#)),  
 Mario TORRISI - University of Catania ([DFA](#))



## INTEROPERABILITY DEMO

## 23.1 About

The map below shows cumulative information about users running the Demo Applications on the various Grid, HPC, Cloud and local resources supported by the CHAIN-REDS Science Gateway. The legend on the right shows the correspondence between marker colours and types of sites. Click on the markers which appear on the map to get more information about running and done (i.e., whose output has not yet been retrieved) jobs at a given site. Job statuses are automatically updated every 15 minute.



## 23.2 Contributors

- Rita Ricceri



## 24.1 About

IntraOperative Electron Radiotherapy (IOERT) [1] is a radiotherapy technique that delivers a single dose of radiation directly to the tumor bed, or to the exposed tumor, during surgery. The objective is to achieve a higher dose in the target volume while dose-limiting structures are surgically displaced. IOERT is used for limited-stage breast tumors treatment and also successfully for prostate, colon and sarcoma cancers. For this purpose, a new generation of mobile linear accelerators has been designed to deliver radiation therapy in the operating theater.

As in conventional radio-therapy techniques, the use of Monte Carlo simulations is mandatory to design the beam collimation system and to study radioprotection characteristics as the radiation leakages. In the clinical activities the simulations can be used to commissioning of the linac and in the optimization of the therapeutic dose and patient radioprotection.

## 24.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the Italian e-Infrastructure.

Portlet Settings

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	GRIDIT
VOname *	gridit
TopBDII *	ldap://egee-bdii.cnaf.infn.it:2170
WMS	wms://wms014.cnaf.infn.it:7443/glite_wms_wmproxy_server
Endpoint *	N/A
MyProxyServer *	myproxy.ct.infn.it
eTokenServer *	etokenserver.ct.infn.it
Port *	8082
Serial Number *	bc779e33367eaad7882b9dfaa83a432c
Role	
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | 5 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the [GARR Science Gateway](#).

Portlet Settings

?

AppID \*

84

?

Log Level \*

VERBOSE

?

Output Path \*

/tmp

?

SoftwareTAG

GEANT4-09-05-patch-01

?

HostName

?

UserName

?

Password

?

SMTP

smtp.gmail.com

?

Sender

sgwadmin@garr.it



« Prev | 1 | 2 | 3 | 4 | 5 | Next »

## 24.3 Usage

To run the Monte Carlo simulations the user has to:

- click on the *third* accordion of the portlet,
- upload the macro as ASCII file OR paste its content in the below textarea, and
- select the number of jobs to be executed as shown in the below figure:

**3 Specify your Input Settings**

Please, paste the macro you want to run in the textarea below, OR upload it in an ASCII file

☒ Upload macro Scegli file defaultMacro.mac

☐ Insert macro

Description Please, insert here a description for your job

☐ Notification

CPU Time Max CPU Time per simulation (in min.): 1440

Number of GEANT4 simulations to run: 20

Each simulation will produce:

- *std.txt*: the standard output file;
- *std.err*: the standard error file;
- *.tar.gz*: containing the results of the Monte Carlo simulation.

A typical simulation produces, at the end, the following files:

```
]$ tree IortTherapySimulationStarted_646/
IortTherapySimulationStarted_646/
-- std.err
-- std.txt
-- output.README
-- results.tar.gz
```

The list of files produced during the run are the following:

```
]$ tar ztvf results.tar.gz
currentEvent.rndm
currentRun.rndm
Dose.out
Energy_MeV.out
```



## 24.4 References

- Concurrency and Computation: Practice and Experience (2014). Published online in Wiley Online Library. DOI: 10.1002/cpe.3268: “A *GEANT4* web-based application to support Intra-Operative Electron Radiotherapy usign the European grid infrastructure” – 2014 [2];

## 24.5 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),

Carlo CASARINO - Laboratorio di Tecnologie Oncologiche ([LATO](#)),

Giuliana Carmela CANDIANO - Laboratorio di Tecnologie Oncologiche ([LATO](#)),

Giuseppe Antonio Pablo CIRRONE - Italian National Institute of Nuclear Physics (LNS)  
[INFN\\_LNS](#),

Susanna GUATELLI - Centre for Medical Radiation Physics, School of Engineering Physics, University of Wollongong, NSW 2522 Australia,

Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#))



## KNOWLEDGE BASE

### 25.1 About

The **CHAIN-REDS Knowledge Base** is one of the largest existing e-Infrastructure-related digital information systems. It currently contains information, gathered both from dedicated surveys and other web and documental sources, for largely more than half of the countries in the world. Information is presented to visitors through geographic maps and tables. Users can choose a continent in the map and, for each country where a marker is displayed, get the information about the Regional Research & Education Network(s) and the Grid Regional Operation Centre(s) (ROCs) the country belongs to as well as the National Research & Education Network, the National Grid Initiative, the Certification Authority, and the Identity Federation available in the country, down to the Grid site(s) running in the country and the scientific application(s) developed by researchers of the country and running on those sites.



Besides e-Infrastructure sites, services and applications, the CHAIN-REDS Knowledge Base publishes information about Open Access Document Repositories and Data Repositories.

- Open Access Document Repositories -



## 25.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

## 25.3 Support

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),

Rita RICCERI - Italian National Institute of Nuclear Physics (INFN),

Salvatore MONFORTE - Italian National Institute of Nuclear Physics (INFN)

**Date** June 4th, 2015



## 26.1 About



---

The **MPI Example** (mpi-portlet) consists of a portlet example able to submit a parallel MPI job into one or more distributed infrastructures (DCIs). This portlet contains almost all GUI elements to provide a complete input interface together with the necessary code to deal with DCIs settings, portlet preferences etc. The aim of this portlet is to use its code as a template that Science Gateway developers may customize to fit their own specific requirements. To facilitate the customization process, a `customize.sh` bash script is included inside the source code package.

## 26.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure the preference settings into the portlet preferences pane.

Preferences are splitted in three separate parts: Generic, Infrastructures and the application execution setting. The generic part contains the **Log level** which contains one of following values, sorted by decreasing level: info, debug, warning and error. The **Application Identifier** refers to the `theId` field value of the GridEngine 'UsersTracking' database table: `GridInteractions`. The infrastructure part consists of different settings related to the destination of users job execution. The fields belonging to this category are:

**Enable infrastructure:** A true/false flag which enables or disable the current infrastructure;

**Infrastructure Name:** The infrastructure name for these settings;

**Infrastructure Acronym:** A short name representing the infrastructure;

**BDII host:** The Infrastructure information system endpoint (URL). Infrastructure preferences have been thought initially for the elite Grid based infrastructures;

**WMS host:** It is possible to specify which is the brokering service endpoint (URL);

**Robot Proxy values:** This is a collection of several values which configures the robot proxy settings (Host, Port, proxyID, VO, Role, proxy renewal);

**Job requirements:** This field contains the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc.

Generic preferences	
Log Level	error
Application Identifier	113
Infrastructure numbers	6
Infrastructure preferences	
Enable infrastructure	yes
Infrastructure number	1
Infrastructure name	EUMEDGRID-Support infrastructure
Infrastructure acronym	EUMEDGRID
BDII Host	ldap://bdii.eumedgrid.eu:2170
WMS Hosts	
Proxy Robot host server	etokenserver.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	false
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	eumed
Proxy Robot VO Role	eumed
Proxy Robot Renewal Flag	true
Job Requirements	
<div> <div>&lt;</div> <div>-</div> <div>+</div> <div>&gt;</div> </div>	
Application job requirements	
Application pilot job	pilot_script.sh
Use the Local Proxy	
<div> <div>Set preferences</div> <div>Reset</div> <div>Back to application</div> </div>	

Actually, depending on the infrastructure, some of the fields above have an overloaded meaning. Please contact the support for further information or watch existing production portlet settings.



## 26.3 Usage

The usage of the portlet is simple; the user can select to upload a file, or insert inside the text field any input text. The **job identifier** text is a human readable value that users will use to keep track of any job execution on DCIs Following buttons: Demo, Submit, Reset values and About are respectively:

*Submit* - Executes the given macro on the distributed infrastructure

*Reset* - Resets the input form

*About* - Gives an overview of the portlet

## 26.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),

Riccardo BRUNO - Italian National Institute of Nuclear Physics (INFN),



---

## MULTI-INFRASTRUCTURE PARALLEL JOB

---

### 27.1 About



This portlet represents a template that allows you to develop your own portlet to submit and run special jobs.

You can choose the kind of parallel job you would like to run from a list containing the following elements:

1. **Job Collection:** is a simple parallel application that spawns N sub-jobs; when all these are successfully completed the whole collection becomes DONE.
2. **Workflow N1:** is a parallel application that spawns N sub-jobs, waits until all these are correctly completed and then submits a new job whose input files are the outputs of the N sub-jobs. When also this “*final job*” is successfully executed, the whole Workflow N1 becomes DONE.
3. **Job Parametric:** is a parallel application that spawns N sub-jobs with the same executable and with different arguments (i.e., input parametrers); when all these are successfully completed the whole parametric job becomes DONE.

### 27.2 Installation

This section explains how to deploy mi-parallel-portlet to submit parallel jobs towards a Distributed Computing infrastructure.

1. Move into your Liferay plugin SDK portlets folder and cloneget the mi-paralle-portlet through the following git command:

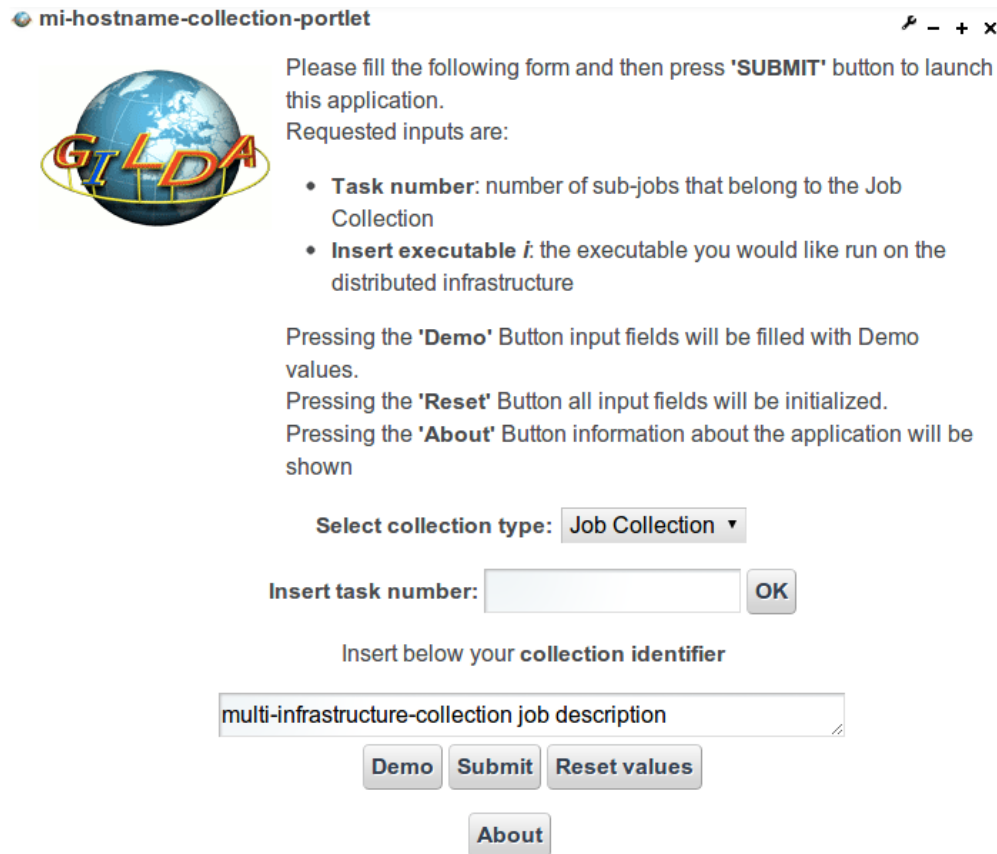
```
git clone https://github.com/csgf/mi-parallel-portlet.git
```

2. Now, move into the just created mi-parallel-potlet directory and execute the deploy command:

```
ant deploy
```

When the previous command has completed, verify that the portlet is “*Successfully autodeployed*”, look for a string like this in the Liferay log file under \$LIFERAY\_HOME/glassfish-3.1.2/domains/domain1/logs/server.log.

3. Then, open your browser at <http://localhost:8080> click Add > More in the GILDA menu, click on Add button to add this new portlet. following picture shows the correctly result:



**mi-hostname-collection-portlet**

Please fill the following form and then press 'SUBMIT' button to launch this application.  
Requested inputs are:

- **Task number:** number of sub-jobs that belong to the Job Collection
- **Insert executable *f*:** the executable you would like run on the distributed infrastructure

Pressing the 'Demo' Button input fields will be filled with Demo values.  
Pressing the 'Reset' Button all input fields will be initialized.  
Pressing the 'About' Button information about the application will be shown

Select collection type: Job Collection ▼

Insert task number:  OK

Insert below your collection identifier

Demo Submit Reset values

About

As soon as the portlet has been successfully deployed you have to configure:

1. the list of e-Infrastructures where the application can be executed;
2. some additional application settings.

Some e-Infrastructure has been already defined as default, in order to simplify the portlet usage.

1. To configure other e-Infrastructure, from the portlet preferences, you have to be provided the following parameters:
  - **Enable infrastructure:** A yes/no flag which enables or disable the generic e-Infrastructure;
  - **Infrastructure name:** A label representig the e-Infrastructure;
  - **Infrastructure acronym:** The acronym to reference the e-Infrastructure;
  - **BDII:** The Top BDII for this e-Infrastructure;
  - **WMS Hosts:** A separated ; list of WMS endpoint for this e-Infrastructure;
  - **Proxy Robot host server:** The eTokenServer for this e-Infrastructure;
  - **Proxy Robot host port:** The eTokenServer port for this e-Infrastructure;
  - **Proxy Robot secure connection:** A true/false flag if the eTokenServer require a secure connection;
  - **Proxy Robot Identifier:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

- **Proxy Robot Virtual Organization:** The VO for this e-Infrastructure;
- **Proxy Robot VO Role:** The VO role for this e-Infrastructure;
- **Proxy Robot Renewal Flag:** A true/false Flag to require proxy renewal before it expires;
- **Local Proxy:** The path to the proxy if you are using a local proxy;
- **Software Tags:** The list of software tags requested by the application.

The following figure shown how the portlet has been configured to run simulation on the EUMEDGRID-Support e-Infrastructure.

2. To configure the application, the following settings have to be provided:

- **Grid operation identifier:** The application identifier as registered in the UserTracking MySQL database (GridOperations table), the default value is 10 and in order to see the submitted special jobs status you should insert a new in usertracking database, if it doesn't already exist, using the following command:

```
INSERT INTO GridOperation VALUES (10, '<portal name>' , '<application description>');

--portal name: is a label representing the portal name;
--application description: is a label representing the application name.
```

- **Log Level:** The log level for the application (e.g.: *INFO* or *VERBOSE*).

## 27.3 Usage

The run special jobs you should:

1. select the kind of special job from the combobox;
2. provide the number of task;
3. the input required;
4. a label to identify yours collections;
5. finally, click on the Submit button to execute this collection.

Select the parallel application type that you want perform, then fill the following form and press **'SUBMIT'** button to launch the selected application.  
Pressing the **'Demo'** Button input fields will be filled with Demo values.  
Pressing the **'Reset'** Button all input fields will be initialized.  
Pressing the **'About'** Button information about the application will be shown

**1. Select the collection**  
Select collection type: Job Collection ▼

**2. Provide the tasks number**  
Insert task number: 2

**Sub Jobs**

**3. Specify the input required**

Executables	Arguments

**4. An identifier for this collection**  
Insert below your **collection identifier**

multi-infrastructure-collection job description

You can also select the collection type from the combo box, and press the Demo button that submits a demo that consists of 3 tasks.

Now move to the MyJob portlet and if all went well, this is the result that you should see:

**MyJobs** ✎ - + x

**Active Jobs List** Done Jobs List

The table below shows the status of your jobs.

Statuses are automatically updated every 15 minutes so there is no need to reload this page more frequently. However, if you don't see your jobs in the table within a reasonable amount of time (a couple of hours at most), click on Help in the the MyWorkspace portlet and notify us the problem.

Once your jobs have finished, you have 96 hours to retrieve their output. Beyond that time, the output of your jobs will automatically be deleted from the Science Gateway in order not to fill its storage with undesired stuff.

Copy Print Save Download Job output

Search:

Show **10** entries First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	mi-hostname	Demo Collection	2013-07-22 15:21:51.0	RUNNING

User Description	Started on (UTC)	Status
Demo Collection_1	2013-07-22 15:21:51.0	RUNNING
Demo Collection_0	2013-07-22 15:21:51.0	SUBMITTED
Demo Collection_2	2013-07-22 15:21:51.0	SUBMITTED

Showing 1 to 1 of 1 entries First Previous 1 Next Last

When all jobs are successfully completed the whole collection becomes **DONE** and you can download the output on your PC, as shown below.

Copy Print Save Download Job output

Search:

Show **10** entries First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
Job Collections	mi-parallel-portlet	Demo Collection: 16/6/2015 - 17:44:45	2015-06-16 15:44:47.0	
Job Collections	mi-parallel-portlet			

Showing 1 to 2 of 2 entries

Want Easy Social Collaboration? Get Social Office ►

**DemoCollection1662015174445\_25.tgz**

Archive Edit View Help

Open Add Files Extract

Back Location: /DemoCollection1662015174445\_25/

Name	Size	Type
DemoCollection1662015174445_0_246	26 bytes	Folder
DemoCollection1662015174445_1_247	29 bytes	Folder
DemoCollection1662015174445_2_248	51 bytes	Folder

3 objects (106 bytes)

## 27.4 Contributors

Mario TORRISI

Riccardo BRUNO





---

## MULTI-INFRASTRUCTURE “HELLO WORLD!”

---

### 28.1 About



---

The **MI-HOSTNAME** (mi-hostname-portlet) consists of a portlet example able to submit a job into one or more distributed infrastructures (DCIs). This portlet contains almost all GUI elements to provide a complete input interface together with the necessary code to deal with DCIs settings, portlet preferences etc. The aim of this portlet is to use its code as a template that Science Gateway developers may customize to fit their own specific requirements. To facilitate the customization process, a `customize.sh` bash script is included inside the source code package.

### 28.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure the preference settings into the portlet preferences pane.

Preferences are splitted in three separate parts: Generic, Infrastructures and the application execution setting. The generic part contains the **Log level** which contains one of following values, sorted by decreasing level: info, debug, warning and error. The **Application Identifier** refers to the `theId` field value of the GridEngine ‘UsersTracking’ database table: `GridInteractions`. The infrastructure part consists of different settings related to the destination of users job execution. The fields belonging to this category are:

**Enable infrastructure:** A true/false flag which enables or disable the current infrastructure;

**Infrastructure Name:** The infrastructure name for these settings;

**Infrastructure Acronym:** A short name representing the infrastructure;

**BDII host:** The Infrastructure information system endpoint (URL). Infrastructure preferences have been thought initially for the elite Grid based infrastructures;

**WMS host:** It is possible to specify which is the brokering service endpoint (URL);

**Robot Proxy values:** This is a collection of several values which configures the robot proxy settings (Host, Port, proxyID, VO, Role, proxy renewal);

**Job requirements:** This field contains the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc.

Generic preferences	
Log Level	error
Application Identifier	113
Infrastructure numbers	6
Infrastructure preferences	
Enable infrastructure	yes
Infrastructure number	1
Infrastructure name	EUMEDGRID-Support infrastructure
Infrastructure acronym	EUMEDGRID
BDII Host	ldap://bdii.eumedgrid.eu:2170
WMS Hosts	
Proxy Robot host server	etokenserver.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	false
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	eumed
Proxy Robot VO Role	eumed
Proxy Robot Renewal Flag	true
Job Requirements	
<div> <div>&lt;</div> <div>-</div> <div>+</div> <div>&gt;</div> </div>	
Application job requirements	
Application pilot job	pilot_script.sh
Use the Local Proxy	
<div> <div>Set preferences</div> <div>Reset</div> <div>Back to application</div> </div>	

Actually, depending on the infrastructure, some of the fields above have an overloaded meaning. Please contact the support for further information or watch existing production portlet settings.

## 28.3 Usage

The usage of the portlet is simple; the user can select to upload a file, or insert inside the text field any input text. The **job identifier** text is a human readable value that users will use to keep track of any job execution on DCIs Following buttons: Demo, Submit, Reset values and About are respectively:

*Submit* - Executes the given macro on the distributed infrastructure

*Reset* - Resets the input form

*About* - Gives an overview of the portlet

## 28.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),

Riccardo BRUNO - Italian National Institute of Nuclear Physics (INFN),



---

## NUCLEMD

---

### 29.1 About

NUCLEMD is a computer code based on the Constrained Molecular Dynamics model. The peculiarity of the algorithm consists on the isospin dependence of the nucleon-nucleon cross section and on the presence of the Majorana Exchange Operator in the nucleon-nucleon collision term.

The code will be devoted to the study of Single and Double Charge Exchange processes in nuclear reactions at low and intermediate energies.

The aim is to provide theoretical support to the experimental results of the DREAMS collaboration obtained by means

of the  spectrometer.

### 29.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the Italian e-Infrastructure.

**Portlet Settings**

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	GRIDIT
VOname *	gridit
TopBDII *	ldap://egee-bdii.cnaf.infn.it:2170
WMS Endpoint *	wms://wms004.cnaf.infn.it:7443/glite_wms_wmproxy_serv
MyProxyServer *	myproxy.ct.infn.it
eTokenServer *	etokenserver2.ct.infn.it
Port *	8082
Serial Number *	bc779e33367eaad7882b9dfaa83a432c
WebDAV	
Role	/gridit
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | 5 | **6** | 7 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the [CHAIN\\_REDS](#) Science Gateway [1].

Portlet Settings

?

 AppID \*

137

?

 Log Level \*

VERBOSE

?

 Metadata Server

?

 Output Path \*

/tmp

?

 SoftwareTAG

?

 HostName

?

 UserName

?

 Password

?

 SMTP

smtp.gmail.com

?

 Sender

credentials-admin@ct.infn.it



« Prev | 1 | 2 | 3 | 4 | 5 | 6 | Z | Next »

## 29.3 Usage

To run the simulations the user has to:

- click on the *third* accordion of the portlet,
- select the binary release
- upload the input files OR use the pre-configured demo ones, and
- select the Max Wall Clock Time (WCT) requested for the execution as shown in the below figure:

3 NUCLEMD Settings

Please, upload your input files, or choose a demo, to run the model

**Binary Release** ☒ v0.1 ☐ v0.2

**Upload INP file**  Nessun file selezionato. (.inp)

**Upload CONF (I)**  Nessun file selezionato. (.dat)

**Upload CONF (II)**  Nessun file selezionato. (.dat)


**Description**

**Wall Clock Time**

Max WallClock time per simulation (in min.): 2025

**Run demo** ☒ Demo input files selected for release v0.1:  
[ cmd\_coll1.inp ]  
[ conf\_ca40\_tr0165.dat ]  
[ conf\_180\_new\_1.dat ]

**Notification** ☒ SMTP settings:  
[ smtp.gmail.com, 25 ]



Each simulation will produce:

- *std.out*: the standard output file;
- *std.err*: the standard error file;
- *nuclemd.log*: the NUCLEMD log file;
- *.tar.gz*: containing the NUCLEMD output results.

A typical simulation produces, at the end, the following files:

```
]$ tree NUCLEMDSimulationStarted_1826/
NUCLEMDSimulationStarted_1826/
```



```
-- std.err
-- std.out
-- output.README
-- nuclemd.log
-- results.tar.gz
```

The list of files produced during the run are the following:

```
]$ tar ztvf results.tar.gz
18040Ca_out
18040Ca_t_out
fort.6
output.README
POT.DAT
rp_out_runco.conf
seed_dat_runco.conf
```

## 29.4 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Gianluca GIULIANI - Italian National Institute of Nuclear Physics - LNS ([INFN\\_LNS](#)),  
Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#))



## 30.1 About



---

GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and nonlinear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. The Octave language is quite similar to Matlab so that most programs are easily portable.

## 30.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure the preference settings into the portlet preferences pane.

Preferences are splitted in three separate parts: Generic, Infrastructures and the application execution setting. The generic part contains the **Log level** which contains one of following values, sorted by decreasing level: info, debug, warning and error. The **Application Identifier** refers to theId field value of the GridEngine 'UsersTracking'database table: GridInteractions. The infrastructure part consists of different settings related to the destination of users job execution. The fields belonging to this category are:

**Enable infrastructure:** A true/false flag which enables or disable the current infrastructure;

**Infrastructure Name:** The infrastructure name for these settings;

**Infrastructure Acronym:** A short name representing the infrastructure;

**BDII host:** The Infrastructure information system endpoint (URL). Infrastructure preferences have been thought initially for the elite Grid based infrastructures;

**WMS host:** It is possible to specify which is the brokering service endpoint (URL);

**Robot Proxy values:** This is a collection of several values which configures the robot proxy settings (Host, Port, proxyID, VO, Role, proxy renewal);

**Job requirements:** This field contains the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc.

Generic preferences	
Log Level	error
Application Identifier	113
Infrastructure numbers	6
Infrastructure preferences	
Enable infrastructure	yes
Infrastructure number	1
Infrastructure name	EUMEDGRID-Support infrastructure
Infrastructure acronym	EUMEDGRID
BDII Host	ldap://bdii.eumedgrid.eu:2170
WMS Hosts	
Proxy Robot host server	etokenserver.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	false
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	eumed
Proxy Robot VO Role	eumed
Proxy Robot Renewal Flag	true
Job Requirements	
<div> <div>&lt;</div> <div>-</div> <div>+</div> <div>&gt;</div> </div>	
Application job requirements	
Application pilot job	pilot_script.sh
Use the Local Proxy	
<div> <div>Set preferences</div> <div>Reset</div> <div>Back to application</div> </div>	

Actually, depending on the infrastructure, some of the fields above have an overloaded meaning. Please contact the support for further information or watch existing production portlet settings.

## 30.3 Usage

The usage of the portlet is simple; the user can select to upload a local Octave macro file selecting the Browse button in the **Application input file** section, or insert inside the text field the Octave macro text by pasting a text or editing directly on the larger text box below. The **job identifier** text is a human readable values that users will use to keep track of any job execution. Following buttons: Demo, Submit, Reset values and About are respectively:

*Demo* - Fills the Macro Text box with an Octave macro example

*Submit* - Executes the given macro on the distributed infrastructure

*Reset* - Resets the input form

*About* - Gives an overview of the portlet

The screenshot shows a web portlet interface. At the top, there is a section labeled "Application' input file" containing a "Browse..." button and the text "No file selected.". Below this is a large text area with the placeholder text "Insert here your text file, or upload a macro file". Underneath the text area is a label "Insert below your job identifier" followed by a text input field containing "Octave execution ...". At the bottom of the portlet, there are four buttons: "Demo", "Submit", "Reset values", and "About".

## 30.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** [Roberto BARBERA](#) - Italian National Institute of Nuclear Physics (INFN),

[Riccardo BRUNO](#) - Italian National Institute of Nuclear Physics (INFN),



---

## PARALLEL SEMANTIC SEARCH

---

### 31.1 About

The Parallel Semantic Search portlet aims to correlate enriched [Open Access Document Repositories](#) and [Data Repositories](#) to create linked data and discover new knowledge through semantic enrichment of metadata and to search in parallel across the millions of resources contained in many different repositories.

The user can search, in more than 100 languages, in parallel across [CHAIN-REDS Knowledge Base](#) and in:

- [ENGAGE](#)
- [EUROPEANA](#)
- [CULTURA ITALIA](#)
- [ISIDORE](#)
- [OPENAGRIS](#)
- [PUBMED](#)

### 31.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

Once the portlet is installed, through its preferences page is possible to select which SPARQL endpoints are wanted to be queried and the number of records to display per page, as shown in the following figure.

PARALLELSEMANTICSEARCH-PORTLET

Select Repositories for your semantic search engine

☐ OpenAgris 
☒ Europeana 
☒ Cultura Italia 
☒ Isidore 
☐ Pubmed 
☐ Engage

Choose record's number to view in the result page

(Default value is 10)

Set preferences

Back To Application Reset

### 31.3 Usage

The search can be performed:


- By typing one or more keywords in the textbox search. After entering the search string, clicking the “Search” button the system submit a SPARQL query that retrieve the results from the CHAIN-REDS repository.
- Using search filters. Inserting in textbox search string, for example, “author: G. Smith” and clicking “Search” will be launched a SPARQL query that will return only the resources of CHAIN-REDS that have at least one author who matches “Smith G.”. The possible filters (according to the Dublin Core Standard) are : dc: author, dc: subject dc: type and dc: publisher. Clicking the “Examples” button are show examples of search filters in 4 of the 110 languages supported by the system.

Examples			
ENGLISH	ARABIC	CHINESE	RUSSIAN
author:Smith G. subject:policy type:thesis format:image/jpeg publisher:elsevier	author:الحامد subject:الإدارة التربوية type:مصادر ومراجع publisher:مجلة جامعة الملك سعود	author:邓祥征 subject:台灣文學 type:期刊论文 publisher:信州大学人文学部	author:ИвановичСкупский subject:Христианство type:Монография format:электронная копия publisher:Тбилиси
close			



In both cases, assuming that the endpoints selected from the portlet preference are Europeana, Cultura Italia and Isidore, the system executes four query in parallel mode and the results for each repository are subdivide by tabs, as shown in the following figures

You will search in CHAIN-REDS-KB , Europeana , Culturalitalia , Isidore

 Examples  Search

Enter a keyword

CHAIN-REDS-KB Europeana Culturalitalia Isidore

Records found. Displaying 1 to 10

Title

[The Campanile, Giotto Florence](#) [Citations] [Linked Data]

Author

*Bradley James; Bradley James;*

Description


*Part of the Archives' Visual Materials collection*

Repository

[Scholarly Materials And Research @ Georgia Tech](#)

[More Info](#)

You will search in CHAIN-REDS-KB , Europeana , Culturalitalia , Isidore

 Examples  Search

Enter a keyword

CHAIN-REDS-KB Europeana Culturalitalia Isidore

Records found. Displaying 1 to 10

Title

[Giotto](#) [Linked Data]  
(TYPE: IMAGE)

Author


*DTZ.DTZG=sec. XIX;DTS.DTSI=1844;DTS.DTSF=1844;DTM=bibliografia; [Production]*

Description

*busto*

[More Info](#)

You will search in CHAIN-REDS-KB , Europeana , Culturaltalia , Isidore

 Examples  Search

Enter a keyword

CHAIN-REDS-KB Europeana **Culturaltalia** Isidore


Records found. Displaying 1 to 10

Title

Biblioteca comunale Giotto [\[Linked Data\]](#)  
(Type: Legal Body)

[More Info](#)

You will search in CHAIN-REDS-KB , Europeana , Culturaltalia , Isidore

 Examples  Search

Enter a keyword

CHAIN-REDS-KB Europeana Culturaltalia **Isidore**

Records found. Displaying 1 to 9

Title

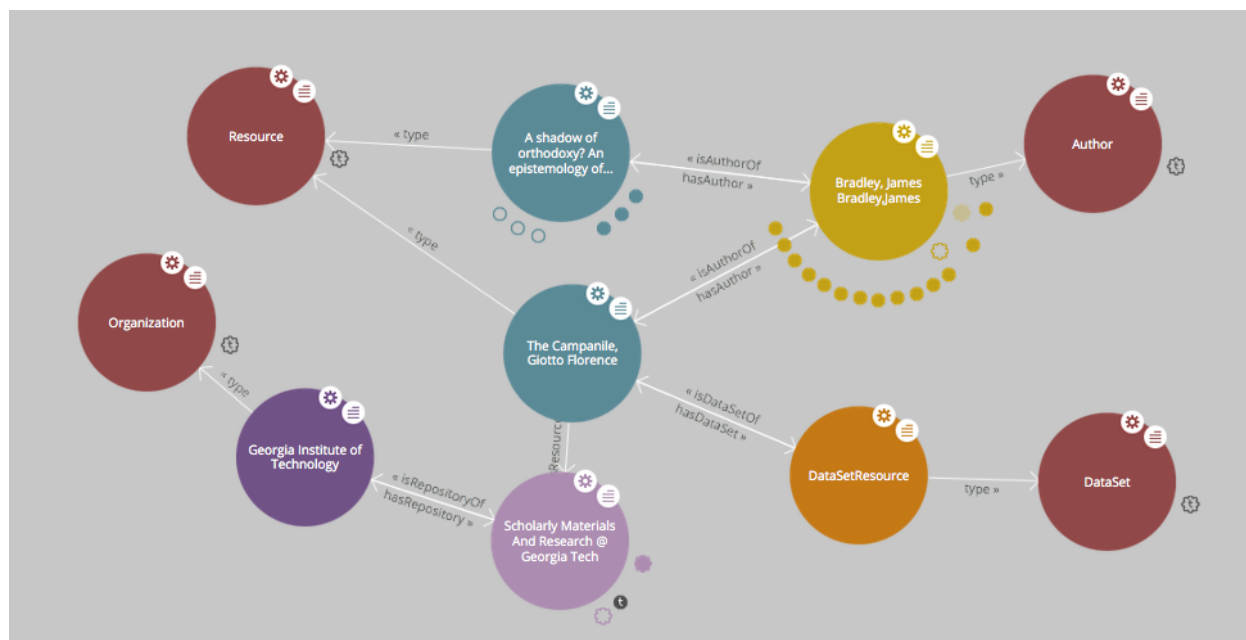
6ème centenaire de Giotto [\[Linked Data\]](#)  
6ème centenaire de [\[Linked Data\]](#)

[More Info](#)

Clicking on “More Info” link, you can access all the details of the resource.

Clicking on “[Citations]” link (only for CHAIN-REDS tab), if available, are displayed more information from Google Scholar about the versions and the quotations of the resource.

Clicking on “[Linked Data]” link will be open a new tab that shows a [LodLive](#) graph of the resource with all its metadata.



## 31.4 References

- ‘A CHAIN-REDS Perspective about Data Access and Metadata Management’. 3rd International Platform on Integrating Arab e-infrastructure in a Global Environment, e-AGE 2013. December 12-13, 2013. Tunisia. [1]
- The CHAIN-REDS Semantic Search Engine.[2]
- Science Gateways for Semantic-Web- Based Life Science Applications. Studies in Health Technology and Informatics, Volume 175: HealthGrid Applications and Technologies Meet Science gateways for Life Sciences, pp 119-130. DOI: 10.3233/978-1-61499-054-3-119 [3]
- The CHAIN-REDS Knowledge Base and Semantic Search Engine. e-Infrastructures for e-Sciences 2013. A CHAIN-REDS Workshop organised under the eegis of the European Commission (eleS2013). October 22, 2013 Beijing, P.R. of China. [4]
- A CHAIN-REDS solution for accessing computational services. Cuarta Conferencia de Directores de Tecnología de Información, TICAL2014 Gestión de las TICs para la Investigación y la Colaboración, Cancún, del 26 al 28 de mayo de 2014.[5]
- CHAIN-REDS application use cases on data management. “4th International Platform on Integrating Arab e-Infrastructure in a Global Environment, e-AGE 2014” to be held in Grant Hyatt Muscat, Oman, 10 – 11 December, 2014.[6]
- CHAIN-REDS DART Challenge. Biredial-ISTEC 2014, At Porto Alegre, Rio Grande du Sul Brazil. DOI: 10.13140/2.1.3532.8961 [7]

## 31.5 Contributors

Please feel free to contact us any time if you have any questions or comments.

**Authors** Rita **RICCERI** - Italian National Institute of Nuclear Physics (INFN),

Giuseppina **INSERRA** - Italian National Institute of Nuclear Physics (INFN),

Carla CARRUBBA - Italian National Institute of Nuclear Physics ([INFN](#))

---

## PI CALCULATION

---

### 32.1 About



The **PICALC** consists of a portlet example able to submit a parallel MPI job that calculates the PI into one or more distributed infrastructures (DCIs). This portlet contains almost all GUI elements to provide a complete input interface together with the necessary code to deal with DCIs settings, portlet preferences etc. The aim of this portlet is to use its code as a template that Science Gateway developers may customize to fit their own specific requirements. To facilitate the customization process, a `customize.sh` bash script is included inside the source code package.

### 32.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure the preference settings into the portlet preferences pane.

Preferences are splitted in three separate parts: Generic, Infrastructures and the application execution setting. The generic part contains the **Log level** which contains one of following values, sorted by decreasing level: info, debug, warning and error. The **Application Identifier** refers to theId field value of the GridEngine 'UsersTracking' database

table: GridInteractions. The infrastructure part consists of different settings related to the destination of users job execution. The fields belonging to this category are:

**Enable infrastructure:** A true/false flag which enables or disable the current infrastructure;

**Infrastructure Name:** The infrastructure name for these settings;

**Infrastructure Acronym:** A short name representing the infrastructure;

**BDII host:** The Infrastructure information system endpoint (URL). Infrastructure preferences have been thought initially for the elite Grid based infrastructures;

**WMS host:** It is possible to specify which is the brokering service endpoint (URL);

**Robot Proxy values:** This is a collection of several values which configures the robot proxy settings (Host, Port, proxyID, VO, Role, proxy renewal);

**Job requirements:** This field contains the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc.

Generic preferences	
Log Level	error
Application Identifier	113
Infrastructure numbers	6

Infrastructure preferences	
Enable infrastructure	yes
Infrastructure number	1
Infrastructure name	EUMEDGRID-Support infrastructure
Infrastructure acronym	EUMEDGRID
BDII Host	ldap://bdii.eumedgrid.eu:2170
WMS Hosts	
Proxy Robot host server	etokenserver.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	false
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	eumed
Proxy Robot VO Role	eumed
Proxy Robot Renewal Flag	true
Job Requirements	

Application job requirements	
<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">Application pilot job</div>	pilot_script.sh
Use the Local Proxy	

Set preferences

Reset

Back to application

Actually, depending on the infrastructure, some of the fields above have an overloaded meaning. Please contact the support for further information or watch existing production portlet settings.

## 32.3 Usage

The usage of the portlet is simple; the user can select to upload a file, or insert inside the text field any input text. The **job identifier** text is a human readable value that users will use to keep track of any job execution on DCIs Following buttons: Demo, Submit, Reset values and About are respectively:

*Submit* - Executes the given macro on the distributed infrastructure

*Reset* - Resets the input form

*About* - Gives an overview of the portlet

## 32.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics (INFN),

Riccardo BRUNO - Italian National Institute of Nuclear Physics (INFN),



---

## R - STATISTICAL COMPUTING

---

### 33.1 About



R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

### 33.2 Installation

Following instructions are meant for science gateway maintainers while generic users can skip this section. To install the portlet it is enough to install the war file into the application server and then configure the infrastructure settings into the portlet preferences pane.

Preferences are splitted in three separate parts: Generic, Infrastructures and the application execution setting. The generic part contains the **Log level** which contains one of following values, sorted by decreasing level: info, debug, warning and error. The **Application Identifier** refers to theId field value of the GridEngine 'UsersTracking' database table: GridInteractions. The infrastructure part consists of different settings related to the destination of users job execution. The fields belonging to this category are:

**Enable infrastructure:** A true/false flag which enables or disable the current infrastructure;

**Infrastructure Name:** The infrastructure name for these settings;

**Infrastructure Acronym:** A short name representing the infrastructure;

**BDII host:** The Infrastructure information system endpoint (URL). Infrastructure preferences have been thought initially for the elite Grid based infrastructures;

**WMS host:** It is possible to specify which is the brokering service endpoint (URL);

**Robot Proxy values:** This is a collection of several values which configures the robot proxy settings (Host, Port, proxyID, VO, Role, proxy renewal);

**Job requirements:** This field contains the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc.

Generic preferences	
Log Level	error
Application Identifier	113
Infrastructure numbers	6
Infrastructure preferences	
Enable infrastructure	yes
Infrastructure number	1
Infrastructure name	EUMEDGRID-Support infrastructure
Infrastructure acronym	EUMEDGRID
BDII Host	ldap://bdii.eumedgrid.eu:2170
WMS Hosts	
Proxy Robot host server	etokenserver.ct.infn.it
Proxy Robot host port	8082
Proxy Robot secure connection	false
Proxy Robot Identifier	bc681e2bd4c3ace2a4c54907ea0c379b
Proxy Robot Virtual Organization	eumed
Proxy Robot VO Role	eumed
Proxy Robot Renewal Flag	true
Job Requirements	
<div> <div>&lt;</div> <div>-</div> <div>+</div> <div>&gt;</div> </div>	
Application job requirements	
Application pilot job	pilot_script.sh
Use the Local Proxy	
<div> <div>Set preferences</div> <div>Reset</div> <div>Back to application</div> </div>	

Actually, depending on the infrastructure, some of the fields above have an overloaded meaning. Please contact the support for further information or watch existing production portlet settings.

## 33.3 Usage

The usage of the portlet is simple; the user can select to upload a local R macro file selecting the Browse button in the **Application input file** section, or insert inside the text field the R macro text by pasting a text or editing directly on the larger text box below. The **job identifier** text is a human readable values that users will use to keep track of any job execution. Following buttons: Demo, Submit, Reset values and About are respectively:

*Demo* - Fills the Macro Text box with an R-Macro example

*Submit* - Executes the given macro on the distributed infrastructure

*Reset* - Resets the input form

*About* - Gives an overview of the portlet

The screenshot displays a web portlet interface. At the top, there is a section titled "Application' input file" which contains a "Browse..." button and the text "No file selected.". Below this is a large text area with the placeholder text "Insert here your text file, or upload a file". Underneath the text area is a label "Insert below your job identifier" followed by a text input field containing the placeholder text "Job execution details ...". At the bottom of the portlet, there are four buttons: "Demo", "Submit", "Reset values", and "About".

## 33.4 Contributor(s)

To get support such as reporting a bug, a problem or even request new features, please contact

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Riccardo BRUNO - Italian National Institute of Nuclear Physics ([INFN](#)),

## SEMANTIC SEARCH

### 34.1 About

The Semantic Search portlet aims to correlate enriched [Open Access Document Repositories](#) and [Data Repositories](#) to create linked data and discover new knowledge through semantic enrichment of metadata. The user can search in more than 100 languages across more than 30 million resources contained in the thousands of semantically enriched OADRs and DRs included in the CHAIN Knowledge Base. Search results are ranked according to the [Ranking Web of Repositories](#).

### 34.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

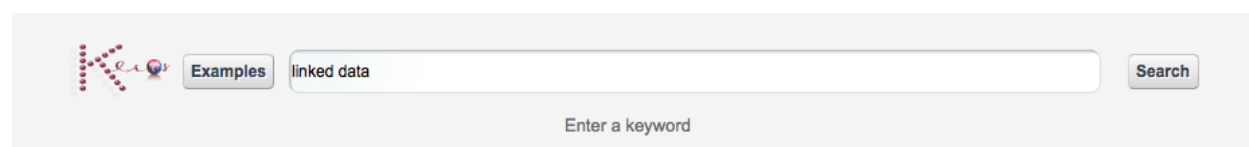
### 34.3 Usage

The search can be performed:

- By typing one or more keywords in the textbox search. After entering the search string, clicking the “Search” button the system submit a SPARQL query that retrieve the results from the CHAIN-REDS repository.

Example of query to retrieve the resources with the keyword in title.

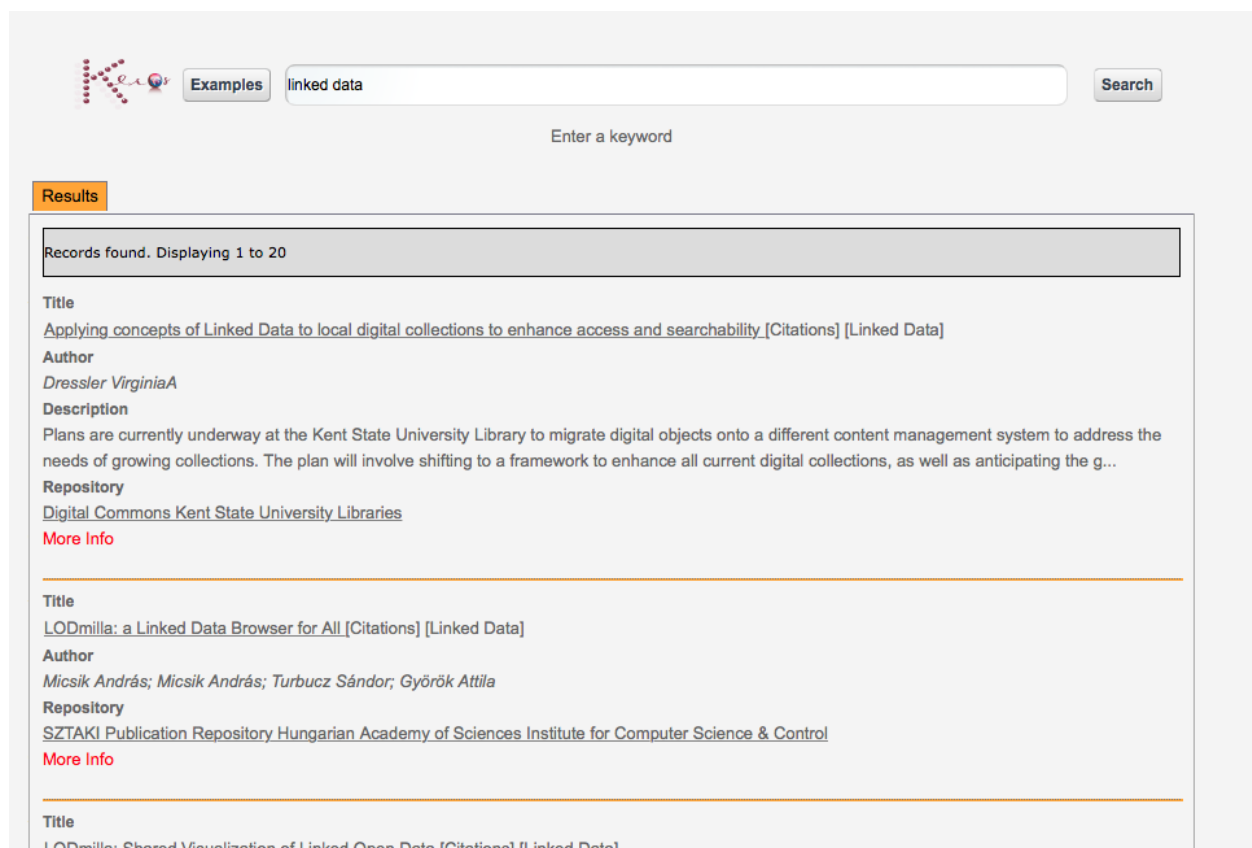
```
SELECT distinct ?s WHERE { ?s dc:title ?title. ?title bif:contains " + keyword + ". ?s
<http://semanticweb.org/ontologies/2013/2/7/RepositoryOntology.owl#isResourceOf>
?rep. ?rep <http://www.semanticweb.org/ontologies/2013/2/7/RepositoryOntology.owl#rank>
?rank. }ORDER BY ASC(?rank) limit 20 offset 0
```



- Using search filters. Inserting in textbox search string, for example, “author: G. Smith” and clicking “Search” will be launched a SPARQL query that will return only the resources of CHAIN-REDS that have at least one author who matches “Smith G.”. The possible filters (according to the Dublin Core Standard) are : dc: author, dc: subject dc: type and dc: publisher. Clicking the “Examples” button are show examples of search filters in 4 of the 110 languages supported by the system.



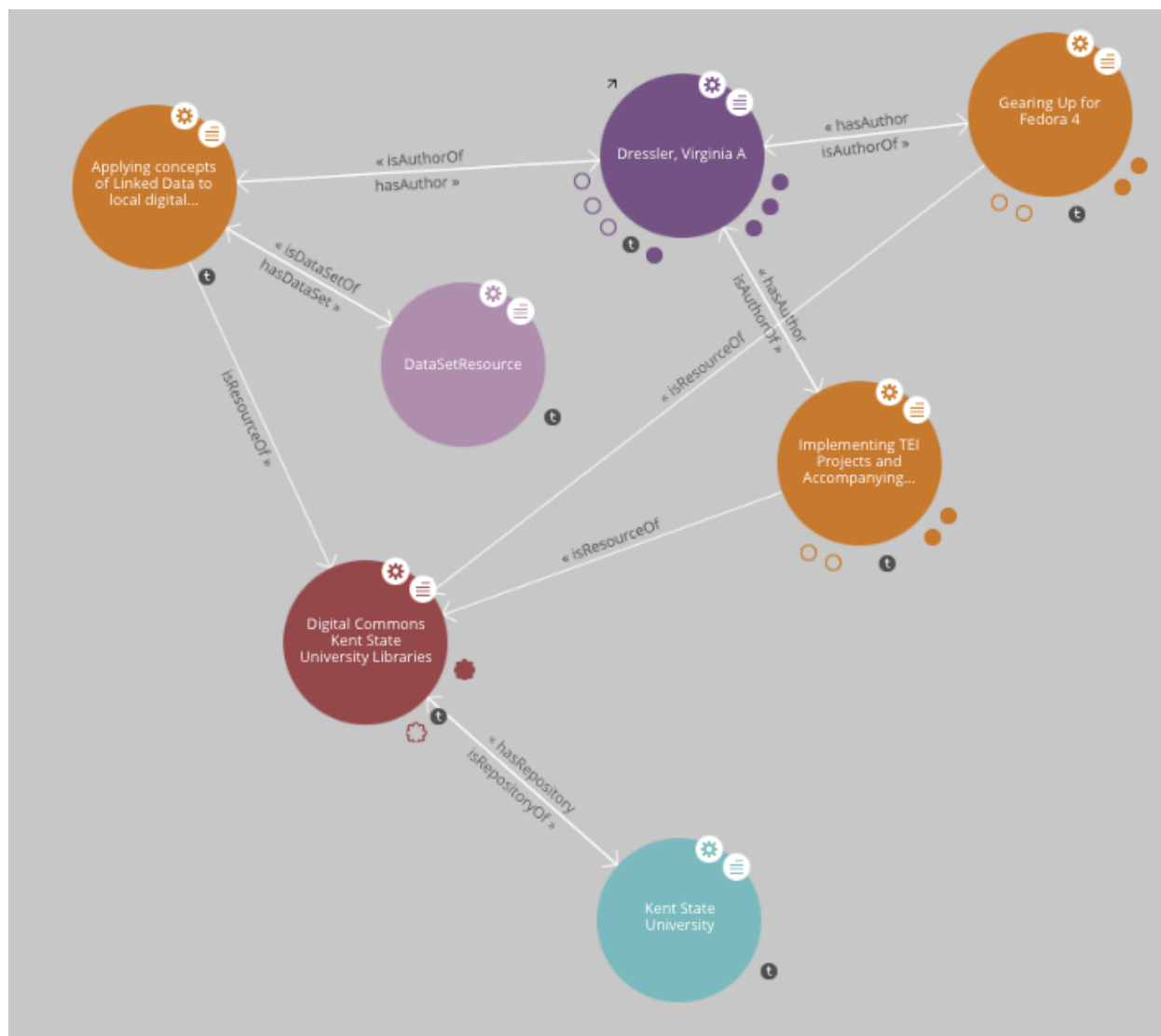
In both cases the results of the query will be processed further, to obtain the final results, as shown in Figure



Clicking on “More Info” link you can access all the details of the resource.

Clicking on “[Citations]” link, if available, are displayed more information from Google Scholar about the versions and the quotations of the resource.

Clicking on “[Linked Data]” link will be open a new tab that shows a [LodLive](#) graph of the resource with all its metadata.



## 34.4 References

- ‘A CHAIN-REDS Perspective about Data Access and Metadata Management’. 3rd International Platform on Integrating Arab e-infrastructure in a Global Environment, e-AGE 2013. December 12-13, 2013. Tunisia. [1]
- The CHAIN-REDS Semantic Search Engine.[2]
- Science Gateways for Semantic-Web- Based Life Science Applications. Studies in Health Technology and Informatics, Volume 175: HealthGrid Applications and Technologies Meet Science gateways for Life Sciences, pp 119-130. DOI: 10.3233/978-1-61499-054-3-119 [3]
- The CHAIN-REDS Knowledge Base and Semantic Search Engine. e-Infrastructures for e-Sciences 2013. A CHAIN-REDS Workshop organised under the eegis of the European Commission (eleS2013). October 22, 2013 Beijing, P.R. of China. [4]
- A CHAIN-REDS solution for accessing computational services. Cuarta Conferencia de Directores de Tecnología de Información, TICAL2014 Gestión de las TICs para la Investigación y la Colaboración, Cancún, del 26 al 28 de mayo de 2014.[5]

- CHAIN-REDS application use cases on data management. “4th International Platform on Integrating Arab e-Infrastructure in a Global Environment, e-AGE 2014” to be held in Grant Hyatt Muscat, Oman, 10 – 11 December, 2014.[6]
- CHAIN-REDS DART Challenge. Biredial-ISTEC 2014, At Porto Alegre, Rio Grande du Sul Brazil. DOI: 10.13140/2.1.3532.8961 [7]

## 34.5 Contributors

Please feel free to contact us any time if you have any questions or comments.

**Authors** Rita [RICCERI](#) - Italian National Institute of Nuclear Physics ([INFN](#)),  
Giuseppina [INSERRA](#) - Italian National Institute of Nuclear Physics ([INFN](#)),  
Carla [CARRUBBA](#) - Italian National Institute of Nuclear Physics ([INFN](#))



---

## SONIFICATION

---

### 35.1 About

Data Sonification is the representation of data by means of sound signals, so it is the analog of scientific visualization, where we deal with auditory instead of visual images. Generally speaking any sonification procedure is a mathematical mapping from a certain data set (numbers, strings, images, ...) to a sound string.

Data sonification is currently used in several fields, for different purposes: science and engineering, education and training, since it provides a quick and effective data analysis and interpretation tool. Although most data analysis techniques are exclusively visual in nature (i.e. are based on the possibility of looking at graphical representations), data presentation and exploration systems could benefit greatly from the addition of sonification capacities.

### 35.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRID-Support e-Infrastructure.

Portlet Settings

?

Enabled \*

☒

?

Infrastructure \*

EUMED

?

VOname \*

eumed

?

TopBDII \*

ldap://bdii.eumedgrid.eu:2170

?

WMS Endpoint \*

wms://wms.ulakbim.gov.tr:7443/glite\_wms\_wmproxy\_server

?

MyProxyServer \*

myproxy.ct.infn.it

?

eTokenServer \*

etokenserver.ct.infn.it

?

Port \*

8082

?

Serial Number \*

bc681e2bd4c3ace2a4c54907ea0c379b

?

Role

?

Proxy Renewal

☒

?

Disable VOMS

☐

« Prev

1

2

3

4

5

6

7

8

Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;


**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the [CHAIN-REDS](#) Science Gateway.

**Portlet Settings**

? AppID *	123
? Output Path *	/tmp
? SoftwareTAG	POVRAY-3.6
? HostName	
? UserName	
? Password	
? SMTP	smtp.gmail.com
? Sender	credentials-admin@ct.infn.it



« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | Next »

## 35.3 Usage

To run the simulation the user has to:

- click on the *third* accordion of the portlet and,
- select the input file (e.g. *.ski* or *.mid* files) OR select a demo from the list as shown in the below figure:

3
Specify your Input Settings

Please, paste the text you want to sonify in the textarea below, OR upload it in an ASCII file

?
☐ Upload your ASCII file (Max 2,5MB) \*

Scegli file
Nessun file selezionato

?
☐ Insert here the text to be sonified \*

This is a simple text I want to sonify!


?
Description

Please, insert here a description for your job

?
Algorithm \*


From ASCII text to MIDI file ▼

?
☒ MIDI Analysis


?
☐ Notification


?
Advanced Settings ►


Configure advanced settings for the selected algorithm (if any)

 Range Min. \*


28

 Range Max. \*

236

 Multiscale \*

Dodecatonic Tones ▼

 Speed \*

Slow ▼

Each simulation will produce:

- *std.txt*: the standard output file;
- *std.err*: the standard error file;
- *.wav*: the final MIDI file produced during the sonification process;

- `.png`: a list of 3D rendering images produced with [POV-Ray](#) if enabled.

If MIDI Analysis is enabled, a compilation of functions to analyze and visualize MIDI files in the [Matlab](#) computing environment will be used.

A typical simulation produces, at the end, the following files:

```
]$ tree DataSonificationSimulationStarted_148682
DataSonificationSimulationStarted_148682/
-- messages.mid
-- output.README
-- Text2Midi.err
-- Text2Midi.out
```

## 35.4 References

- Proceedings of the International Conference on Computational Science, ICCS2010, doi:10.1016/j.procs.2010.04.043: “*Data sonification of volcano seismograms and Sound/Timbre reconstruction of ancient musical instruments with Grid infrastructures*” – May, 2010 Amsterdam, The Netherlands [1];

## 35.5 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Roberto BARBERA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Giuseppe LA ROCCA - Italian National Institute of Nuclear Physics ([INFN](#)),  
Mariapaola SORRENTINO - Conservatory of Music of Avellino ([4]),  
Domenico VICINANZA - DANTE ([DANTE](#))



---

## TEMPLATE PORTLET

---

### 36.1 About

**Version Available 1.0**

**Template Portlet**

- Liferay 6.1.1 ([link](#))
- Catania Grid & Cloud Engine 1.5.11 or above ([link](#))



The **template-portlet** consists of a portlet example able to submit jobs towards different kinds of *Distributed Computing Infrastructures (DCIs)*. This portlet contains the relevant elements needed to deal with DCIs. This portlet developed in the contest of the [Sci-GaIA](#) project to support the application development process during the [Sci-GaIA Winter School](#). The aim of the **template-portlet** is to provide an application template that Science Gateway developers can customize to fit their own specific requirements. To make easier the customization process, a *customize.sh* bash script is included inside the source code package.

### 36.2 Installation

This section explains how to deploy and configure the **template-portlet**.

1. Move into your Liferay plugin SDK portlets folder and clone the template-portlet source code through the git clone command:

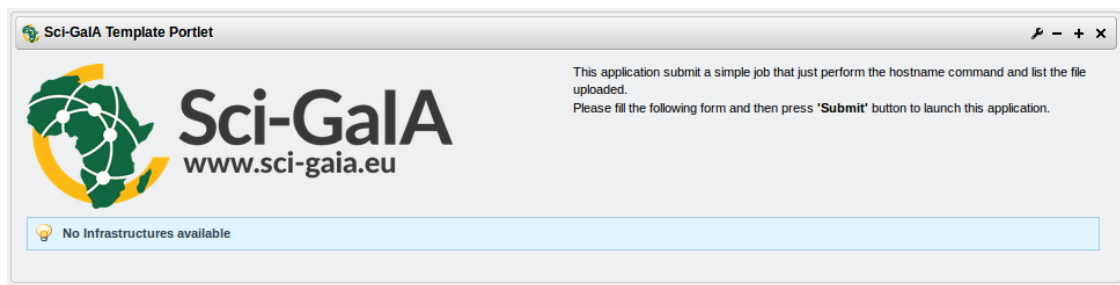
```
git clone https://github.com/csgf/template-portlet.git
```

2. Now, move into the just created template portlet directory and execute the deploy command:

```
ant deploy
```

When the previous command has completed, verify that the portlet was “*Successfully autodeployed*”, look for a string like this in the Liferay log file under `$LIFERAY_HOME/glassfish-3.1.2/domains/domain1/logs/server.log`.

3. Then, open your browser and point at your Science Gateway instance and from there click Add > More in the Sci-GaIA category, click on Add button to add this new portlet. Following picture shows the correctly result:



As soon as the portlet has been successfully deployed you have to configure it using the portlet configuration menu. Portlet configuration is splitted in two parts: *Generic application preferences*, *Infrastructures preferences*.

### 36.2.1 Generic application preferences

The generic part contains:

- **Application Identifier** the identifier assigned to the application in the GridInteractions database table.
- **Application label** (*Required*) a short meaningful label for the application.
- **Production environment** a boolean flag that specifies if the portlet will be used in a production or development environment.
  - if *true* the development environment preferences will be shown
    - \* **UserTrackingDB hostname** hostname of the Grid and Cloud Engine Usertracking database. Usually *localhost*
    - \* **UserTrackingDB username** username of the Grid and Cloud Engine Usertracking database user. Usually *user\_tracking*
    - \* **UserTrackingDB password** password specified for the Usertracking database user. Usually *usertracking*
    - \* **UserTrackingDB database** Grid and Cloud Engine Usertracking database name. Usually *user-tracking*
- **Application requirements** the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc. defined using JDL format.



Sci-GalA Template Portlet - Configuration

Setup Permissions Sharing

Archive/Restore Setup

**Generic application preferences**

Save

**Application identifier**  
81

**Application label (Required)**  
Template Portlet

**Infrastructure number**  
1

☒ Production environment

**Application requirements**

Edit pilot script

**Available Infrastructures**

Add New

Name	Infrastructure Middleware/Acronym	Status	
SEEGRID	SEEGRID	💡	⏮️ ⚙️ ⏭️ Actions

Showing 1 result.

**Note:** You can get the *Application Identifier* inserting a new entry into the **GridOperations** table:

```
INSERT INTO GridOperation VALUES ('<portal name>', 'Template Portlet');
-- portal name: is a label representing the portal name, you can get the
-- right value from your Science Gateway instance.
```

## 36.2.2 Infrastructure preferences

The infrastructure preferences section shows the e-Infrastructures configured. Using the actions menu on the right side of the table, you can:

- Activate / Deactivate
- Edit
- Delete

an available infrastructure. The *Add New* button is meant to add a new infrastructure available to the application. When you click this button a new panel, will be shown with several fields where you can specify the Infrastructure details.

The fields belonging to this panel are:

- **Enabled** A boolean which enable or disable the current infrastructure.
- **Infrastructure Name (Required)** The infrastructure name for these settings.

- **Middleware** (*Required*) The middleware used by the current infrastructure. Here you can specify 3 different values.
  - **an acronym** for gLite based middleware.
  - **ssh** for HPC Cluster.
  - **rocci** for cloud based middleware.

Following fields will be translated in the relevant infrastructure parameters based on the value specified in this field.

- **BDII host:** The Infrastructure information system endpoint (URL).
  - If Middleware is **ssh** here you can specify a “;” separated string with ssh authentications parameters (username;password or username for key based authentication).
  - If Middleware is **rocci** here you can specify the name of the compute resource that will be created.
- **WMS host:** is the service endpoint (URL).
- **Robot Proxy host server:** the robot proxy server hostname.
- **Robot Proxy host port:** the robot proxy server port.
- **Proxy Robot secure connection:** a boolean to specify if robot proxy server needed a SSL connection.
- **Robot Proxy identifier:** the robot proxy identifier.
- **Proxy Robot Virtual Organization:** the virtual organization configured.
- **Proxy Robot VO Role:** the role virtual organization configured.
- **Proxy Robot Renewal Flag:** a boolean to specify if robot proxy can be renewed before its expiration.
- **RFC Proxy Robot:** a boolean to specify if robot proxy must be RFC.
  - If Middleware is **rocci** this field must be checked.
- **Local Proxy:** the path of a local proxy if you want use this type of authentication.
- **Software Tags:** infrastructure specific information.
  - If Middleware is **rocci** here you can specify a “;” separated string with  
`<image_id>;<flavor>;<link_resource>`

Sci-GaIA Template Portlet - Configuration

Setup Permissions Sharing

Archive/Restore Setup

Infrastructure preferences « Back

☒ Enabled

Infrastructure name (Required)

Middleware (Required) ⓘ

BDII Host

WMS Hosts

Proxy Robot host server

Proxy Robot host port

☐ Proxy Robot secure connection

Proxy Robot Identifier

Proxy Robot Virtual Organization

Proxy Robot VO Role

☐ Proxy Robot Renewal Flag

☐ RFC Proxy Robot

Local Proxy

Software Tags

Save Reset

## 36.3 Usage

The usage of the portlet is really simple. The user can optionally select to upload a file and/or specify **job label**, that is a human readable value used to identify the job execution on DCIs. Each field are optional, if you don't specify any label a default one will be created with the username and a timestamp.

Sci-GaIA Template Portlet

Sci-GaIA  
www.sci-gaia.eu

Application input

Input file

Browse... No file selected.

Job label ⓘ

test\_20160524180942

Submit Cancel

This application submit a simple job that just perform the hostname command and list the file uploaded.  
Please fill the following form and then press 'Submit' button to launch this application.

## 36.4 Contributor(s)

If you have any questions or comments, please feel free to contact us using the Sci-GaIA project discussion forum ([discourse.sci-gaia.eu](https://discourse.sci-gaia.eu))

**Authors** Roberto BARBERA - University of Catania ([DFA](#)),  
Bruce BECKER - Council for Scientific and Industrial Research ([CSIR](#)),  
Mario TORRISI - University of Catania ([DFA](#))

---

## TEMPLATE SPECIAL JOB PORTLET

---

**Version Available 1.0**

### Template Special Job Portlet

- Liferay 6.1.1 ([link](#))
- Catania Grid & Cloud Engine 1.5.11 or above ([link](#))

### 37.1 About



The **template-special-job-portlet** consists of a portlet example able to submit special jobs towards different kinds of *Distributed Computing Infrastructures (DCIs)*. This portlet template contains the relevant elements needed to deal with DCIs, it has been developed in the contest of the [Sci-GaIA](#) project to support the application development process during the [Sci-GaIA Winter School](#). The aim of the **template-special-job-portlet** is to provide an application template that Science Gateway developers can customize to fit their own specific requirements. To make easier the customization process, a *customize.sh* bash script is included inside the source code package.

The **template-special-job-portlet** handles three different kinds of special jobs, a user can choose among:

1. **Job Collection:** is a simple parallel application that spawns N sub-jobs; when all these are successfully completed the whole collection becomes DONE.
2. **Workflow N1:** is a parallel application that spawns N sub-jobs, waits until all these are correctly completed and then submits a new job whose input files are the outputs of the N sub-jobs. When also this “*final job*” is successfully executed, the whole Workflow N1 becomes DONE.
3. **Job Parametric:** is a parallel application that spawns N sub-jobs with the same executable and wi

### 37.2 Installation

This section explains how to deploy and configure the **template-special-job-portlet**.

1. Move into your Liferay plugin SDK portlets folder and clone the template-special-job-portlet source code through the git clone command:

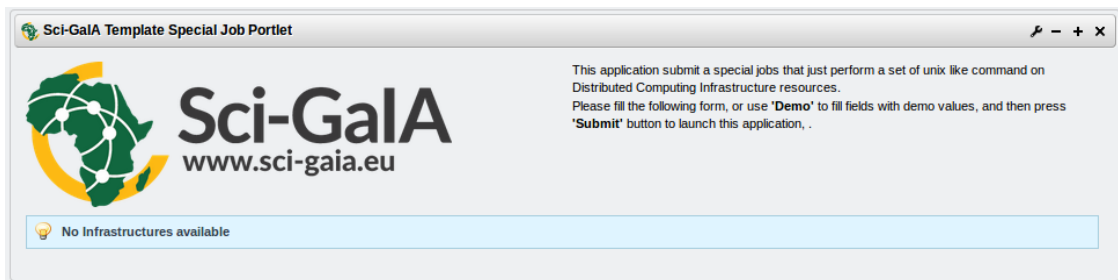
```
git clone https://github.com/csgf/template-special-job-portlet.git
```

2. Now, move into the just created portlet folder and execute the deploy command:

```
ant deploy
```

When the previous command has completed, verify that the portlet was “*Successfully autodeployed*”, look for a string like this in the Liferay log file under `$LIFERAY_HOME/glassfish-3.1.2/domains/domain1/logs/server.log`.

3. Then, open your browser and point at your Science Gateway instance and from there click Add > More in the Sci-GaIA category, click on Add button to add this new portlet. Following picture shows the correctly result:



As soon as the portlet has been successfully deployed you have to configure it using the portlet configuration menu. Portlet configuration is splitted in two parts: *Generic application preferences*, *Infrastructures preferences*.

### 37.2.1 Generic application preferences

The generic part contains:

- **Application Identifier** the identifier assigned to the application in the GridInteractions database table.
- **Application label** (*Required*) a short meaningful label for the application.
- **Production environment** a boolean flag that specifies if the portlet will be used in a production or development environment.
  - if *true* the development environment preferences will be shown
    - \* **UserTrackingDB hostname** hostname of the Grid and Cloud Engine Usertracking database. Usually *localhost*
    - \* **UserTrackingDB username** username of the Grid and Cloud Engine Usertracking database user. Usually *user\_tracking*
    - \* **UserTrackingDB password** password specified for the Usertracking database user. Usually *usertracking*
    - \* **UserTrackingDB database** Grid and Cloud Engine Usertracking database name. Usually *user-tracking*
- **Application requirements** the necessary statements to specify a job execution requirement, such as a particular software, a particular number of CPUs/RAM, etc. defined using JDL format.

Sci-GalA Template Special Job Portlet - Configuration

Setup Permissions Sharing

Archive/Restore Setup

**Generic application preferences**

Save

**Application identifier**  
82

**Application label (Required)**  
Template Special Job Portlet

**Infrastructure number**  
1

☒ Production environment

**Application requirements**

Edit pilot script

**Available Infrastructures**

Add New

Name	Infrastructure Middleware/Acronym	Status	
SEEGRID	SEEGRID	On	Actions

Showing 1 result.

**Note:** You can get the *Application Identifier* inserting a new entry into the **GridOperations** table:

```
INSERT INTO GridOperation VALUES ('<portal name>' , 'Template Special Job Portlet');
-- portal name: is a label representing the portal name, you can get the
-- right value from your Science Gateway instance.
```

### 37.2.2 Infrastructure preferences

The infrastructure preferences section shows the e-Infrastructures configured. Using the actions menu on the right side of the table, you can:

- Activate / Deactivate
- Edit
- Delete

an available infrastructure. The *Add New* button is meant to add a new infrastructure available to the application. When you click this button a new panel, will be shown with several fields where you can specify the Infrastructure details.

The fields belonging to this panel are:

- **Enabled** A boolean which enable or disable the current infrastructure.
- **Infrastructure Name** (*Required*) The infrastructure name for these settings.

- **Middleware** (*Required*) The middleware used by the current infrastructure. Here you can specify 3 different values.
  - **an acronym** for gLite based middleware.
  - **ssh** for HPC Cluster.
  - **rocci** for cloud based middleware.

Following fields will be translated in the relevant infrastructure parameters based on the value specified in this field.

- **BDII host**: The Infrastructure information system endpoint (URL).
  - If Middleware is **ssh** here you can specify a “;” separated string with ssh authentications parameters (username;password or username for key based authentication).
  - If Middleware is **rocci** here you can specify the name of the compute resource that will be created.
- **WMS host**: is the service endpoint (URL).
- **Robot Proxy host server**: the robot proxy server hostname.
- **Robot Proxy host port**: the robot proxy server port.
- **Proxy Robot secure connection**: a boolean to specify if robot proxy server needed a SSL connection.
- **Robot Proxy identifier**: the robot proxy identifier.
- **Proxy Robot Virtual Organization**: the virtual organization configured.
- **Proxy Robot VO Role**: the role virtual organization configured.
- **Proxy Robot Renewal Flag**: a boolean to specify if robot proxy can be renewed before its expiration.
- **RFC Proxy Robot**: a boolean to specify if robot proxy must be RFC.
  - If Middleware is **rocci** this field must be checked.
- **Local Proxy**: the path of a local proxy if you want use this type of authentication.
- **Software Tags**: infrastructure specific information.
  - If Middleware is **rocci** here you can specify a “;” separated string with  
<image\_id>;<flavor>;<link\_resource>



Sci-GalA Template Special Job Portlet - Configuration

Setup Permissions Sharing

Archive/Restore Setup

« Back

Infrastructure preferences

☒ Enabled

Infrastructure name (Required)

Middleware (Required) ⓘ

BDII Host

WMS Hosts

Proxy Robot host server

Proxy Robot host port

☐ Proxy Robot secure connection

Save Reset

Supported middleware are:

- an acronym for gLite based middleware;
- ssh for HPC Cluster;
- rocci for cloud based middleware.

Proxy Robot Identifier

Proxy Robot Virtual Organization

Proxy Robot VO Role

☐ Proxy Robot Renewal Flag

☐ RFC Proxy Robot

Local Proxy

Software Tags

## 37.3 Usage

The usage of the template-special-job-portlet is really simple. The user has to specify the task number he wants like to perform, then he has to specify which kind of special job he wants execute from the provided combobox, then clicking on the *OK* button the interface will be automatically updated to show a set of input fields that the user should fill with a unix like command and the arguments. Furthermore the application provide a *Demo* button that allows to the user to submit a preconfigured *Job Collection* consist of 3 sub-jobs.

Optionally the user can specify also a **job label**, that is a human readable label, used to identify the job execution on DCIs, if he doesn't specify any label a default one will be created with the username and a timestamp.

The screenshot shows a web application window titled "Sci-GaIA Template Special Job Portlet". The header features the Sci-GaIA logo (a green map of Africa with a yellow orbit) and the text "Sci-GaIA www.sci-gaia.eu". A text box on the right explains the application's purpose: "This application submit a special jobs that just perform a set of unix like command on Distributed Computing Infrastructure resources. Please fill the following form, or use 'Demo' to fill fields with demo values, and then press 'Submit' button to launch this application, .".

**Application input**

Select special job  
Job Collection

Task number (Required) 3 OK

**Sub Jobs**

Executables	Arguments
hostname	
ls	
pwd	

Job label ⓘ  
Demo Collection: 25/5/2016 - 13:50:22

Demo Submit Cancel

## 37.4 Contributor(s)

If you have any questions or comments, please feel free to contact us using the Sci-GaIA project dicussion forum ([discourse.sci-gaia.eu](https://discourse.sci-gaia.eu))

**Authors** Roberto BARBERA - University of Catania ([DFA](#)),

Bruce BECKER - Council for Scientific and Industrial Research ([CSIR](#)),

Mario TORRISI - University of Catania ([DFA](#))

## 38.1 About



---

The Center for Atmospheric Research (**CAR**) is an activity Centre of the Nigerian National Space Research and Development Agency, **NASRDA**, committed to research and capacity building in the atmospheric and related sciences.

**CAR** is dedicated to understanding the atmosphere—the air around us—and the interconnected processes that make up the Earth system, from the ocean floor through the ionosphere to the Sun’s core.

The **NASRDA** Center for Atmospheric Research provides research facilities and services for the atmospheric and Earth sciences community.

Tropospheric Data Acquisition Network, **TRODAN**, is a project that was designed to monitor the lower atmosphere which covers region from the surface of the Earth to the altitude of about 11 km.

This project is designed to collect and provide real-time meteorological data from different locations across Nigeria using for the purpose of research and development.

At moment **TRODAN** equipment include atmospheric monitoring facilities such as Automatic Weather Stations, Micro Rain Radar facilities and Vantage Pro. This present data is obtained using Campbell Scientific Automatic Weather Station.

**GNU PLOT** is the portable command-line driven graphing utility for Linux, OS/2, MS Windows, OSX, VMS, and many other platforms used to visualize the Trodan Data Repository.

### Conditions of Use of the **TRODAN** data

The data made available by **CAR** are provided for research use and are not for commercial use or sale or distribution to third parties without the written permission of the Centre. Publications including theses making use of the data should include an acknowledgment statement of the form given below. A citation reference should be sent to the **TRODAN** Project Manager ([trodan@nassrda.com](mailto:trodan@nassrda.com)) for inclusion in a publications list on the **TRODAN** website.

### Disclaimer

CAR-NASRDA accepts no liability for the use or transmission of this data.

## 38.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**MetaData Host:** The Metadata hostname where download/upload digital-assets (e.g. *glibrary.ct.infn.it*);

**MetaData Port:** The Metadata port (e.g. *3000*);

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRIDSupport e-Infrastructure [1].

Portlet Settings

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	EUMED
VOname *	eumed
TopBDII *	ldap://bdii.eumedgrid.eu:2170
WMS Endpoint *	wms://wms.ulakbim.gov.tr:7443/glite_wms_wmproxy_server
WMS Endpoint *	N/A
MyProxyServer *	etokenserver.ct.infn.it
eTokenServer *	etokenserver.ct.infn.it
Port *	8082
MetaData Server	glibrary.ct.infn.it
Port	3000
Serial Number *	bc681e2bd4c3ace2a4c54907ea0c379b
Role	
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | 5 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Log Level:** The log level for the application (e.g.: *INFO* or *VERBOSE*);

**Repository:** The MetaData Repository for this project (e.g.: *trodan*);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;

**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the Africa Grid Science Gateway [2].

**Portlet Settings**

? AppID *	172
? Log Level *	VERBOSE
? Output Path *	/tmp
? Repository	trodan
? SoftwareTAG	TRODAN
? HostName	
? UserName	
? Password	
? SMTP	smtp.gmail.com
? Sender	sgwadmin@africa-grid.org



« Prev | 1 | 2 | 3 | 4 | **5** | Next »

## 38.3 Usage

To run the PoC the user has to click on the *third* accordion of the portlet and

- select the Meteorological station(s) to analyze as shown in the below figure:

**3 Specify your Input Settings**

\* These fields are required

A.) Specify the Meteorological Station(s) you want to analyze \*

- ☒ Abuja
- ☐ Akungba
- ☐ Akure
- ☐ Anyigba
- ☐ Eburumili
- ☒ Lagos
- ☐ Lapai
- ☐ Makurdi
- ☐ Minna
- ☒ Nsukka
- ☐ Ogbomoso
- ☒ Porthacourt
- ☐ Redeemers
- ☐ Yola
- ☐ Toggle Station(s)

- select the Meteorological Pattern(s) to analyze as shown in the below figure:

B.) Specify the Meteorological Pattern(s) you want to analyze \*

<input checked="" type="checkbox"/> Rain Rate (mm)	<input checked="" type="checkbox"/> Solar Radiation (W/m <sup>2</sup> )
<input checked="" type="checkbox"/> Air Temperature (°C)	<input checked="" type="checkbox"/> Relative Humidity (%)
<input checked="" type="checkbox"/> Soil Temperature (°C)	<input checked="" type="checkbox"/> Wind Speed (m/s)
<input type="checkbox"/> Wind Direction (°)	<input type="checkbox"/> Barometric Pressure (mbar)
<input type="checkbox"/> Volumetric Water	<input type="checkbox"/> PA_uS (μs)
<input type="checkbox"/> Toggle Meteorological Pattern(s)	

C.) Specify some additional settings before to start

☐ Plot Style\* lines (Default) ▼

☐ Description Insert here your description

☐ Notification @

- click on *Advanced plot options* and the *Date* range and *Plot Option* as shown in the below figure:



? Advanced plot options ▼

i Date range may vary from [ 31/07/2007 to 21/05/2013 ]

? Date \* From 04/04/2008 To 05/04/2008

? Plot Options \*

- ☐ Diurnal  
Each variable is averaged to give hourly output (24 data point per day)
- ☐ Daily  
All readings for each day averaged to give one value per variable for each calendar day
- ☐ Monthly  
Data averaged to give one value per variable for each month of observation data (12 data points per year)
- ☐ NoOption

Each simulation will produce:

- *std.txt*: the standard output file;
- *std.err*: the standard error file;

Here follows a graphical representation of the following Meteorological Patterns generated with [GNU PLOT](#) in PDF format:

- AirTemperature,
- SoilTemperature,
- RainPrecipitation,
- RelativeHumidity,
- SolarRadiation,
- WindSpeed

## 38.4 Contributor(s)

Please feel free to contact us any time if you have any questions or comments.

**Authors** TRODAN Project Manager



---

## VISUALIZE INFECTION PORTLET

---

### 39.1 About



The visualize portlet is just a simple demonstration graph tool that allows users to see the graphical visualisation of the results output file, which has been generated using the infection model-portlet. This helps users to understand the significance of their results, by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier by making use of this tool. Consequently, when a job is ready and the output is collected, a user can upload the output file, using this portlet (infection model visualisation tool) on the science gateway and a graphical view of their jobs would be generated. The visualize tool graph has two major axis, namely the population and the time (in days). It also has a view to represent the different types of population. i.e the recovered, susceptible and the infected population.

### 39.2 Installation

This section explains how to deploy and configure the visualize-infection-model-portlet into a Science gateway to submit some preconfigures experiments towards Distributed Computing infrastructures.

1. Move into your Liferay plugin SDK portlets folder and clone the infectionModel-portlet source through the following git command:

```
git clone https://github.com/csgf/visualize-infection-model-portlet.git
```

2. Now, move into the just created infectionModel-portlet directory and execute the deploy command:

```
ant deploy
```

When the previous command has completed, verify that the portlet is “*Successfully autodeployed*”, look for a string like this in the Liferay log file under \$LIFERAY\_HOME/glassfish-3.1.2/domains/domain1/logs/server.log.

3. Then, open your browser and point at your Science Gateway instance and from there, click Add > More in the Brunel University category, click on Add button to add this new portlet. The Following picture shows the correct view:

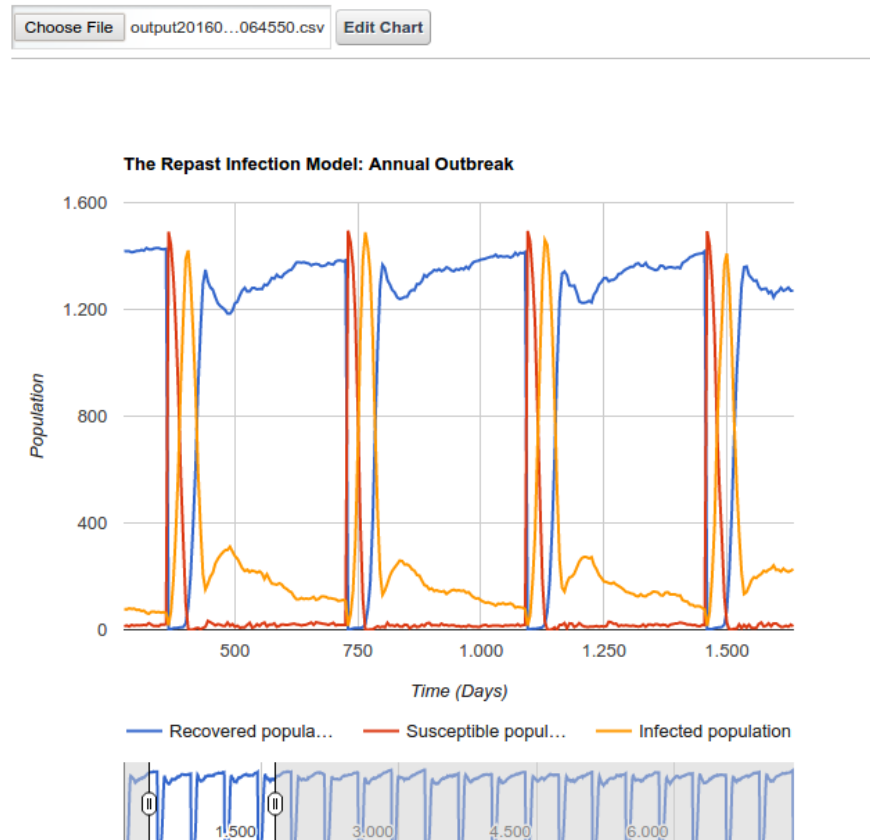
This is an example of how a portlet can be used to visualise the output of the infection model. Select the appropriate **output(timestamp).csv** from your experiments and you will see the results.

Choose File	No file chosen	Edit Chart
-------------	----------------	------------

## 39.3 Usage

The following figure show the view of the visualize-infection-model-portlet and how it can be used to visualize output file results on a cloud based system.

This is an example of how a portlet can be used to visualise the output of the infection model. Select the appropriate **output(timestamp).csv** from your experiments and you will see the results.



The visualize portlet can simply be used by uploading the output file, which was generated from the infection model portlets. This is done by simply clicking on the choose file icon, on the visualzse portlet page as shown above, and users can select the appropriate output.csv file from among their experiments. This will automatically generate a graphical view of their jobs.

## 39.4 Contributor(s)

If you have any questions or comments, please feel free to contact us using the Sci-GaIA project dicussion forum ([discourse.sci-gaia.eu](https://discourse.sci-gaia.eu))

**Authors** Roberto BARBERA - University of Catania ([DFA](#)),  
 Adedeji FABIYI - Brunel University London ([BRUNEL](#)),  
 Simon TAYLOR - Brunel University London ([BRUNEL](#)),  
 Mario TORRISI - University of Catania ([DFA](#))

## 40.1 About



---

The Weather Research and Forecasting (WRF) modelling system [1] is a widely used meso-scale numerical weather prediction system designed to serve both atmospheric research and operational forecasting needs.

WRF has a large worldwide community counting more than 20,000 users in 130 countries and it has been specifically designed to be the state-of-the-art atmospheric simulation system being portable and running efficiently on available parallel computing platforms.

## 40.2 Installation

To install this portlet the WAR file has to be deployed into the application server.

As soon as the portlet has been successfully deployed on the Science Gateway the administrator has to configure:

- the list of e-Infrastructures where the application can be executed;
- some additional application settings.

1.) To configure a generic e-Infrastructure, the following settings have to be provided:

**Enabled:** A true/false flag which enables or disable the generic e-Infrastructure;

**Infrastructure:** The acronym to reference the e-Infrastructure;

**VOName:** The VO for this e-Infrastructure;

**TopBDII:** The Top BDII for this e-Infrastructure;

**WMS Endpoint:** A list of WMS endpoint for this e-Infrastructure (max. 10);

**MyProxyServer:** The MyProxyServer for this e-Infrastructure;

**eTokenServer:** The eTokenServer for this e-Infrastructure;

**Port:** The eTokenServer port for this e-Infrastructure;

**Serial Number:** The MD5SUM of the robot certificate to be used for this e-Infrastructure;

In the following figure is shown how the portlet has been configured to run simulation on the DIT e-Infrastructure [4].

Portlet Settings

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	DIT
Login *	rbarbera
Password *	.....
Cluster Server *	ssh://41.93.32.21
	N/A
MyProxyServer	N/A
eTokenServer	N/A
Port	N/A
Serial Number	N/A
Role	N/A
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Next »

In the following figure is shown how the portlet has been configured to run simulation on the CHAIN-REDS Cloud Testbed [5].

Portlet Settings

?

Enabled \*

☒

?

Infrastructure \*

CHAIN-REDS

?

VOname \*

vo.chain-project.eu

?

TopBDII

?

Cloud

rocci://stack-server-01.ct.infn.it:8787/

?

Resource \*

rocci://nebula-server-01.ct.infn.it:9000/

?

MyProxyServer \*

N/A

?

eTokenServer \*

myproxy.ct.infn.it

?

Port \*

etokenserver.ct.infn.it

?

Serial Number \*

8082

?

Role

bc681e2bd4c3ace2a4c54907ea0c379b

?

Proxy Renewal

vo.chain-project.eu

?

Disable VOMS

☒

?

Disable VOMS

☐

« Prev

1

2

3

4

5

6

7

8

Next »

In the following figure is shown how the portlet has been configured to run simulation on the EUMEDGRIDSupport e-Infrastructure [3].

Portlet Settings

Enabled *	<input checked="" type="checkbox"/>
Infrastructure *	EUMED
VOname *	eumed
TopBDII *	ldap://bdii.eumedgrid.eu:2170
WMS Endpoint *	wms://wms.ulakbim.gov.tr:7443/glite_wms_wmproxy_server
	N/A
MyProxyServer *	myproxy.ct.infn.it
eTokenServer *	etokenserver.ct.infn.it
Port *	8082
Serial Number *	bc681e2bd4c3ace2a4c54907ea0c379b
Role	
Proxy Renewal	<input checked="" type="checkbox"/>
Disable VOMS	<input type="checkbox"/>

« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Next »

2.) To configure the application, the following settings have to be provided:

**AppID:** The ApplicationID as registered in the UserTracking MySQL database (GridOperations table);

**Software TAG:** The list of software tags requested by the application;

**SMTP Host:** The SMTP server used to send notification to users;


**Sender:** The FROM e-mail address to send notification messages about the jobs execution to users;

In the figure below is shown how the application settings have been configured to run on the Africa Grid Science Gateway [2].



**Portlet Settings**

? AppID *	164
? Log Level *	VERBOSE
? Output Path *	/tmp
? SoftwareTAG	CLUSTALW2
? HostName	
? UserName	
? Password	
? SMTP	smtp.gmail.com
? Sender	credentials-admin@ct.infn.it




« Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | Next »

## 40.3 Usage







To run the PoC the user has to click on the *third* accordion of the portlet and start the forecasting analysis as shown in the below figure:

3

VM Settings




Please, use the drop-down list to choose the demo you want to run on the available infrastructures (Grid, Cloud or HPC clusters).

 Virtual Server *	WRF-3.5 ▼
 VM Template *	small ▼
 Simulation *	wrf_demo
 Description	Please, insert here your description
 <input type="checkbox"/> Notification	
 About the demo ►	

~ The current demonstrative WRF simulation refers to a region in Africa (Saudi Arabia, LAT=31.96, LONG=37.98) in the period that comes from 16th. to 17th. of March 2003.

~ The results of this demo will report only the raw data.



The WRF simulation refers to a region in Africa and to a period of two days:





---

## AGINFRA SG MOBILE

---

### 41.1 About



**agINFRA SG Mobile** is a mobile application developed in the contest of [agINFRA project](#). The Android version is available from [Google Play](#). The main aim of this mobile app is to provide an easy way to access, from your mobile appliances, digital assets and metadata stored in different kinds of storage:

- Local storage
- Grid storage
- Cloud storage

The agINFRA SG Mobile currently provides access to the Soil Map data repository.

### 41.2 Installation

To install agINFRA SG Mobile on your devices simply download the app from the store



or scan the following QR code



## 41.3 Usage

To use the agINFRA SG Mobile you need federated credentials issued by an Identity Provider. If the organisation you belong to has an Identity Provider, proceed with the download; otherwise, you can first get federated credentials registering to the “open” Identity Provider, which belongs to the GrIDP federation.

Once the application is installed on your mobile device, you can access the repository using your federated credentials and selecting the organization you belong to and the Identity Provider (see [Figure 1](#)).

If your credentials are correct, the application shows the main view from which you could access the repository, as the [Figure 2](#) shows.

Selecting the repository, the application shows a list of available digital assets (see [Figure 3](#)) from which you could select the digital object. The application provides also a hierarchical filter mechanism that allows you to easily retrieve the asset and metadata you are looking for.

The [Figure 3](#) also shows the replica where the digital asset is available to download it on your device.

## 41.4 Contributors

Antonio CALANDUCCI

Mario TORRISI

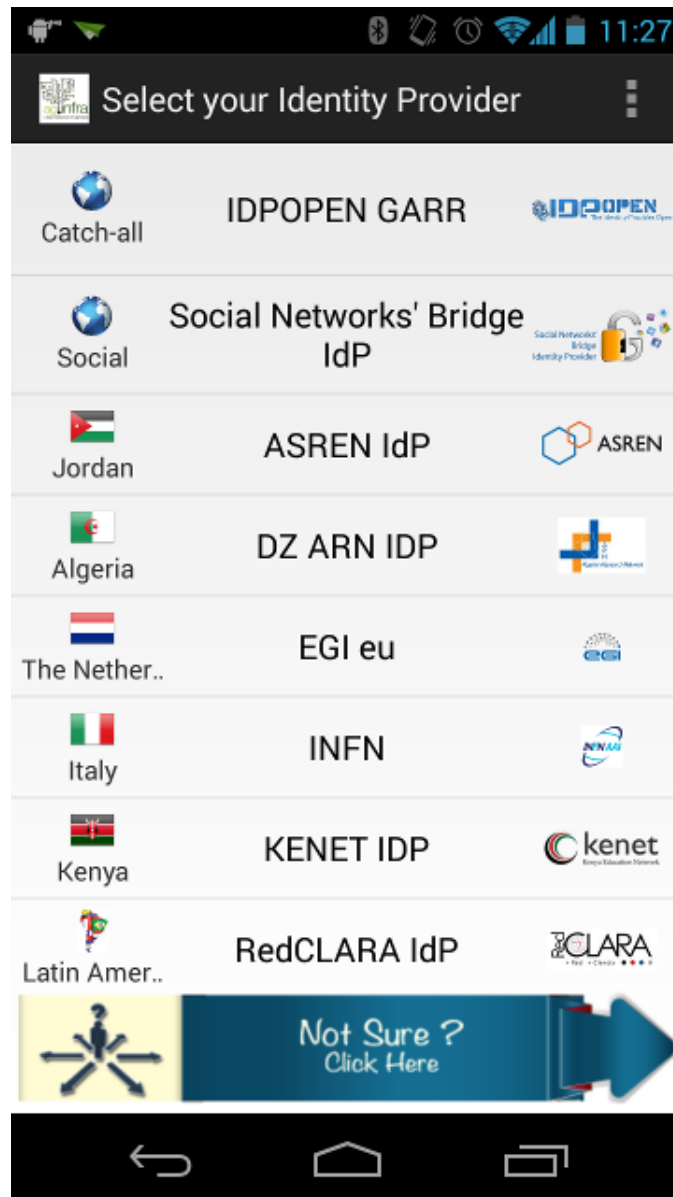


Fig. 41.1: Identity Provider List

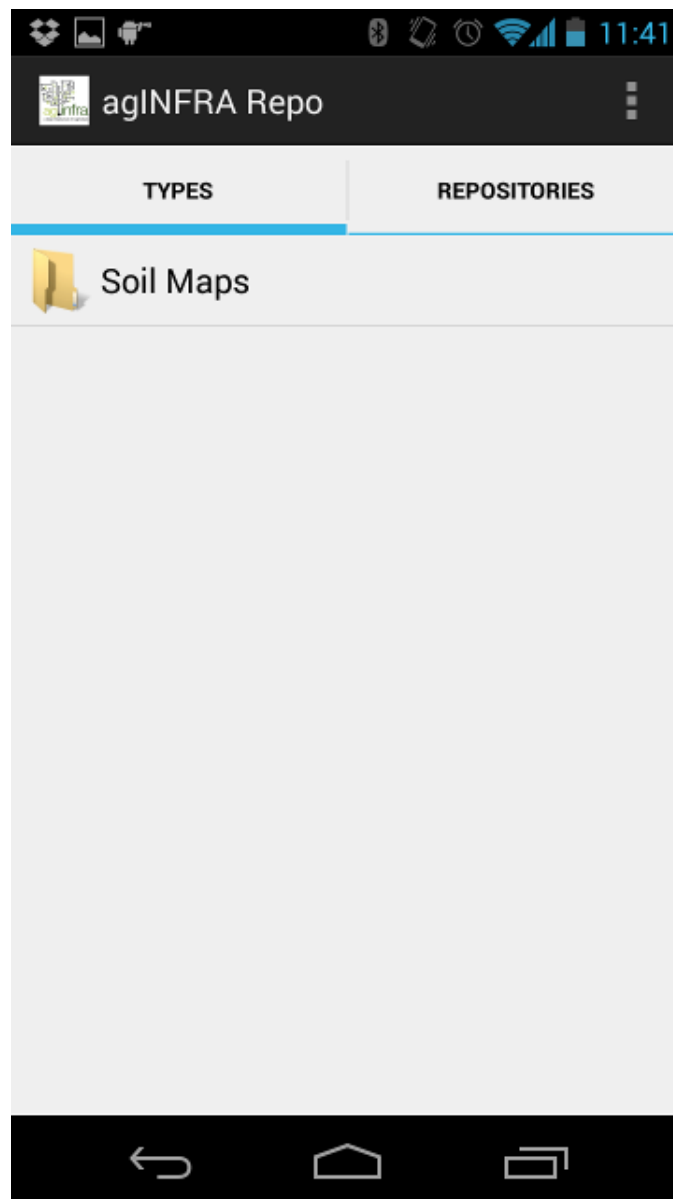


Fig. 41.2: agINFRA Soil Map Repository



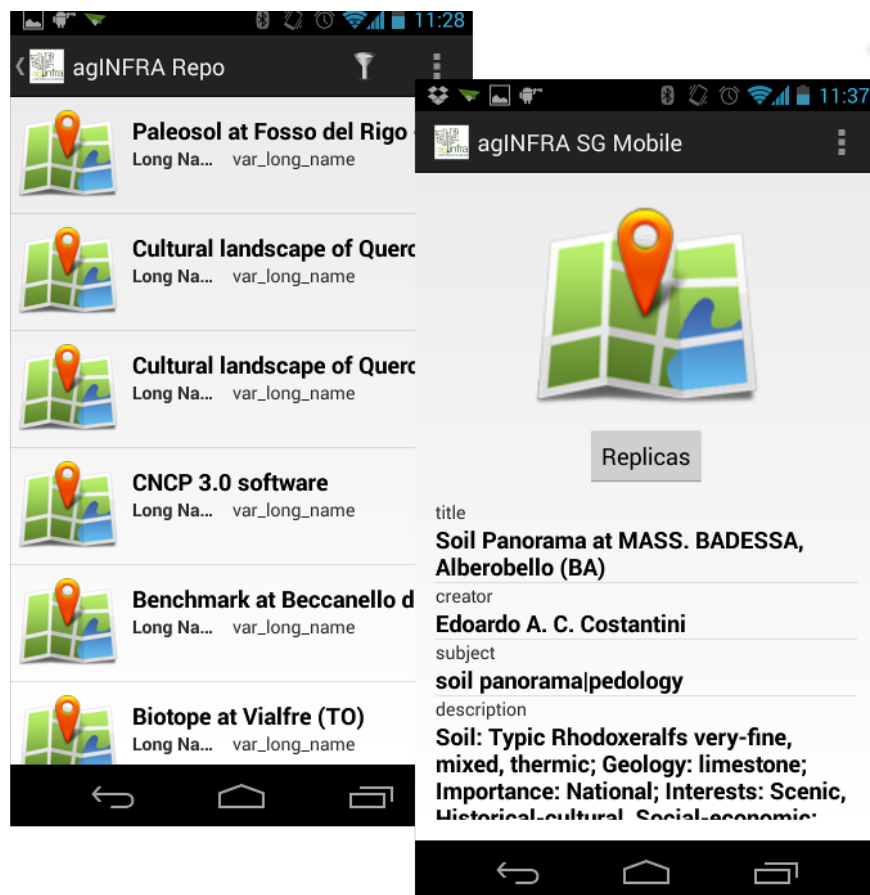


Fig. 41.3: Asset download



---

## DCH-RP ECSG MOBILE

---

### 42.1 About



---

**DCH-RP eCSG Mobile** is a mobile application developed in the contest of [DCH-RP project](#) Android and iOS versions are available from [Google Play](#) and [App Store](#). The main aim of this mobile app is to provide an easy way to access, from your mobile appliances, digital assets and its metadata stored in different kinds of storage:

- Local storage
- Grid storage
- Cloud storage

DCH-RP eCSG Mobile currently provides access to the following repositories:

1. Belgian Science Policy Office ([BELSPO](#))
2. Istituto Centrale per il Catalogo Unico ([ICCU](#))
3. Digital Repository of Federico De Roberto works ([De Roberto DR](#))
4. Digital Repository of the Architectural and Archaeological Heritage in the Mediterranean Area
5. China Relics Data repositories([China Relics DR](#))
6. Center for Documentation of Cultural and Natural Heritage ([CULTNAT Collections](#))

### 42.2 Installation

To install DCH-RP eCSG Mobile on your devices simply download the app from the store



or scan one of the following QR code



## 42.3 Usage

To use the DCH-RP eCSG Mobile you need federated credentials issued by an Identity Provider. If the organisation you belong to has an Identity Provider, proceed with the download; otherwise, you can first get federated credentials registering to the “open” [Identity Provider](#), which belongs to the GrIDP federation.

Once the application is installed on you mobile device, you can access the services using your federated credentials selecting the organization you belong and the Identity Provider (see [Figure 1](#)).

---

Fig. 42.1: Identity Provider List

If your credentials are correct, the application shows the main view from which you could access repositories. As example the [Figure 2](#) shows the repositories available for the ICEU proof of concept.

---

Fig. 42.2: ICEU repositories

Selecting the the type of asset you are interested in, the application shows a list of available digital assets (see [Figure 3](#)) from which you could select the digital object. The application provides also a hierarchical filter mechanism that allows you to easily retrieve the asset and metadata you are looking for.

The [Figure 3](#) shows also the available storages where the digital asset is available and a link to download the asset on your device.

---

Fig. 42.3: Asset download

## 42.4 Contributors

Antonio CALANDUCCI

Mario TORRISI

---

## EARTHSERVER SG MOBILE

---

### 43.1 About



---

**EarthServer SG Mobile** is a mobile application developed in the context of [EarthServer project](#), Android and iOS versions are available from [Google Play](#) and the [App Store](#). The main aim of this mobile app is to provide an easy way to access data services that make use of the OpenSource Geospatial Catalogue (OSGC) standards such as WCS and WMS as well as provide data repository access containing metadata.

The **EarthServer SG Mobile** app allows users to interact with the following four services:

1. Climate Data Services provided by the Meteorological Environmental Earth Observation (MEEEO) WCS server
2. Geological Data Service provided by the British Geological Survey ([BGS](#)) WCS server
3. A generic mobile client for WCS and WMS services
4. A repository browser of atmospheric data coming from the ESA MERIS spectrometer

Be aware that the access to some of the services provided by EarthServer Science Gateway Mobile, requires federated credentials issued by an Identity Provider. If the organisation you belong to has an Identity Provider, proceed with the download; otherwise, you can first get federated credentials registering to the “open” Identity Provider, which belongs to the GrIDP federation

### 43.2 Installation

To install EarthServer SG Mobile on your devices simply download the app from the store



or scan one of the following QR code



### 43.3 Usage

Once the application is installed on you mobile appliance, you can access the four services from the main screen and navigate through a well defined path to exploit the application features.

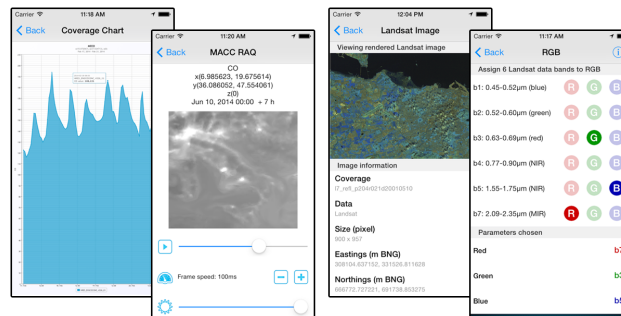


Fig. 43.1: Some screenshots of the MEEO and BGS WCS client.

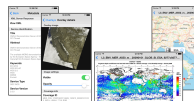


Fig. 43.2: Some screenshots of the generic WCS & WMS client and the MERIS browser.

### 43.4 Contributors

Roberto BARBERA  
 Antonio CALANDUCCI  
 Marco PAPPALARDO  
 Rita RICCIERI

Francesco RUNDO  
Vittorio SORBERA  
Mario TORRISI





---

## Semantic Search

---

### 44.1 About



**Semantic Search Mobile** is a mobile application that allows to browse the almost 3,100 Open Access Data and Documents Repositories integrated in the [CHAIN-REDS](#) project's Knowledge Base, presented either through geographic maps or table. Tapping on a marker on a map or on an element of a table, one gets redirected to the corresponding repository in order to search inside it. The app also allows to use the CHAIN-REDS Semantic Search Engine to search across the more than 30 million resources contained in the Knowledge Base and enriched by means of Semantic Web technologies. For each record found, a short description of the corresponding resource is provided. Tapping on an element of the list, one can access a detailed view with direct links to the document and, if existing, to the related data.

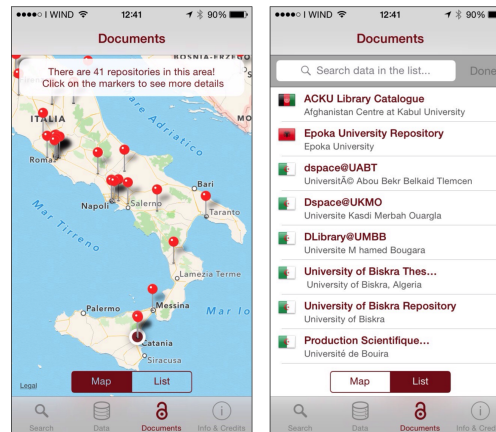
### 44.2 Installation

To install Semantic Search Mobile on your devices simply download the app from the store:



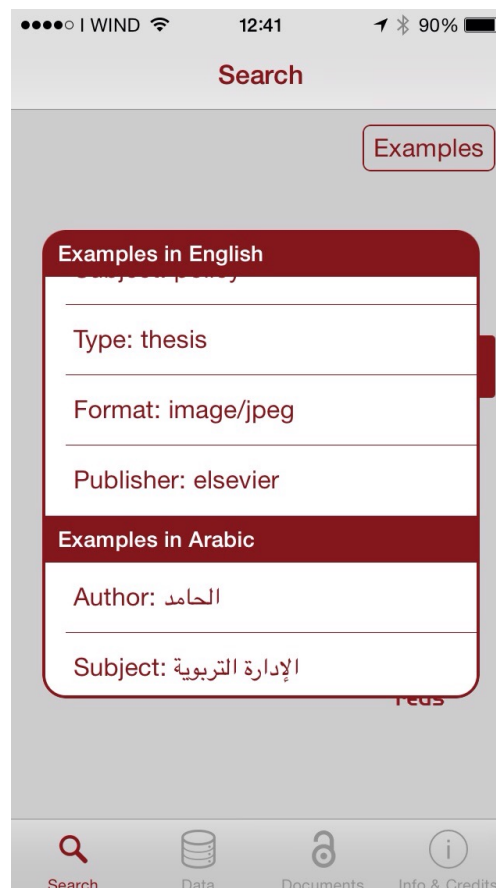
## 44.3 Usage

Once the application is installed on you mobile appliance, the user can select the possible filters (according to the Dublin Core Standard) like : dc: author, dc: subject dc: type and dc: publisher.

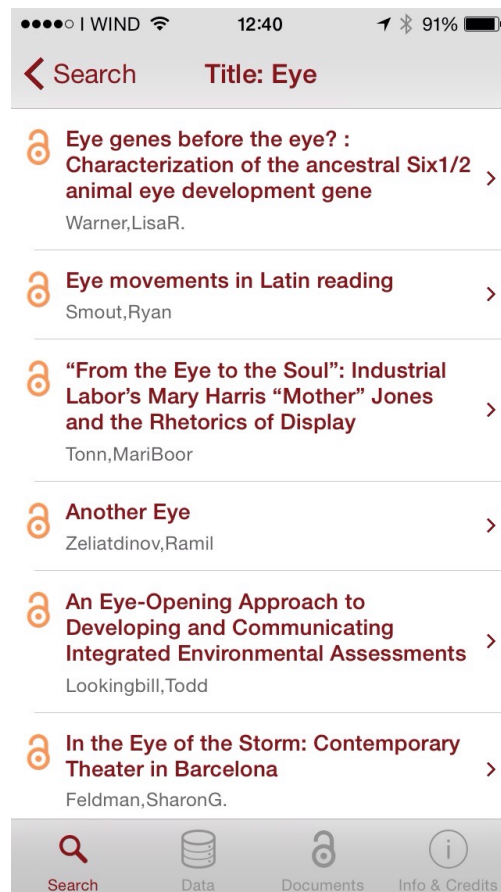


Once choosed the filter (for example “title”) typing one or more keywords in the textbox search, clicking the “Search” button the system submit a [REQUEST \(HTTP\)](#) to get the list of resources that contain the searched keyword inside the title.

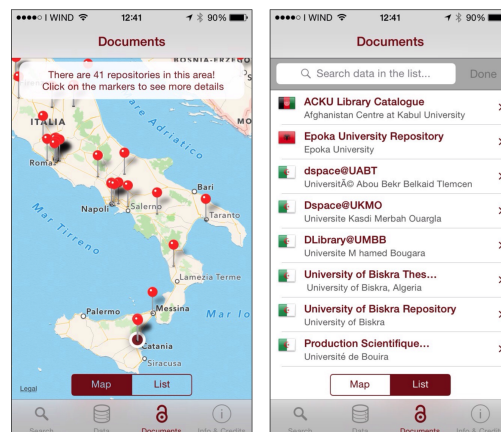
Clicking the “Examples” button are show examples of search filters in 4 of the 110 languages supported by the system.



In both cases the results of the request will be processed to obtain the final results, as shown in Figure



Tapping on Data or Documents the application shows the Data or Documents Repositories in geographic map or in list format.



## 44.4 Contributors

Roberto BARBERA

Carla CARRUBBA

Giuseppina INSERRA

Rita Ricceri

Mario TORRISI

---

## SCOAP3 HARVESTER API

---

### 45.1 About

The project aims to do harvesting of the [SCOAP3](#) resources ,using API KEY, and insert them into [Open Access Repository](#)

### 45.2 Usage

The project contains 3 classes:

- *Scoap3withAPI.java* is the main class. It's mandatory to put your private and public key to do a query (String privateKey and StringpublicKey). Using the startDate parameter, you can find all records created from start date (String startDate)
- *Scoap3\_step2.java* is a class to separate the INFN resources from OTHERS.
- *HmacSha1Signature.java* is a class to generate the signature from private key.

The script returns a folder that contains :

- INFN folder (all INFN resources in MARCXML format )
- OTHER folder (all OTHER resources in MARCXML format )

### 45.3 Contributors

Please feel free to contact us any time if you have any questions or comments.

**Authors** Rita [RICCERI](#) - Italian National Institute of Nuclear Physics ([INFN](#)),  
Giuseppina [INSERRA](#) - Italian National Institute of Nuclear Physics ([INFN](#)),  
Carla [CARRUBBA](#) - Italian National Institute of Nuclear Physics ([INFN](#))



---

## SEMANTIC SEARCH API

---

### 46.1 About

Programmable use of the CHAIN-REDS Semantic Search Engine is possible thanks to a simple RESTful API. The API allows to get and reuse the millions of open access resources contained in the [CHAIN-REDS Knowledge Base](#) and stored in a [Virtuoso](#) RDF-compliant database.

### 46.2 Usage

#### 46.2.1 Get all information about the resources

- REQUEST (HTTP method GET)

To do a request one must insert into the URL three parameters:

1. **keyword**: is the keyword seeked; the keyword can contain any filters for example (keyword=author:SEEKED\_AUTHOR, keyword=subject:SEEKED\_SUBJECT, keyword=type:SEEKED\_TYPE, keyword=format:SEEKED\_FORMAT and keyword=publisher:SEEKED\_PUBLISHER)
2. **limit**: is the maximum number of resources that has to be retrieved by the query;
3. **offset**: is the value in the list of resources to start the retrieval with.

as shown below:

```
http://www.chain-project.eu/virtuoso/api/resources?
keyword=SEEKED_KEYWORD
&
limit=MAX_NUMBER_OF_RESOURCES
&
offset=OFFSET
```

- RESPONSE (application/json)

A collection of resources is represented as a JSON array of objects containing the information about the resources; a single resource is represented as a JSON object. All parameters are Dublin Core Metadata Elements (see <http://dublincore.org/documents/dces/> and <http://dublincore.org/documents/dcmi-terms/>) except the repository's parameters that include the information regarding the repository that contains the resource.

If the keyword is not found, the result is an empty object.

*Example:* Search the first 10 resources (offset=0) that contain the keyword “eye” inside the title.

<http://www.chain-project.eu/virtuoso/api/resources?keyword=eye&limit=10&offset>

## 46.2.2 Get only authors, titles, id and DOI of the resources

- REQUEST (HTTP method GET)

To do a request one must insert into the URL three parameters:

1. **keyword:** is the keyword seeked; the keyword can contain any filters for example (keyword=author:SEEKED\_AUTHOR, keyword=subject:SEEKED\_SUBJECT, keyword=type:SEEKED\_TYPE, keyword=format:SEEKED\_FORMAT and keyword=publisher:SEEKED\_PUBLISHER)
2. **limit:** is the maximum number of resources that has to be retrieved by the query;
3. **offset:** is the value in the list of resources to start the retrieval with.

as shown below:

```
http://www.chain-project.eu/virtuoso/api/simpleResources?
keyword=SEEKED_KEYWORD
&
limit=MAX_NUMBER_OF_RESOURCES
&
offset=OFFSET
```

- RESPONSE (application/json)

A collection of resources is represented as a JSON array of objects containing authors, titles, id and DOI about the resources.

## 46.2.3 Get all information about a single resource

- REQUEST (HTTP method GET)

To do a request one must insert into the URL one parameter:

1. **id:** is the identifier of the resource inside the triple store Virtuoso; it is a uri

as shown below:

[http://www.chain-project.eu/virtuoso/api/singleResource?id=ID\\_RESOURCE](http://www.chain-project.eu/virtuoso/api/singleResource?id=ID_RESOURCE)



- RESPONSE (application/json)

The response is represented as a JSON object containing all information about the single resource.

#### 46.2.4 Get information from Google Scholar by a title

- REQUEST (HTTP method GET)

To do a request one must insert into the URL one parameter:

1. **title**: is the title by which to get information from Google Scholar

as shown below:

[http://www.chain-project.eu/virtuoso/api/infoGS?title=SOURCE\\_TITLE](http://www.chain-project.eu/virtuoso/api/infoGS?title=SOURCE_TITLE)

- RESPONSE (application/json)

The response is represented as a JSON object containing any information from Google Scholar.

#### 46.2.5 Get information from Altmetric by a DOI

- REQUEST (HTTP method GET)

To do a request one must insert into the URL one parameter:

1. **DOI**: is the parameter by which are retrieved all metrics from Altmetric.

as shown below:

[http://www.chain-project.eu/virtuoso/api/infoAltmetric?doi=SOURCE\\_DOI](http://www.chain-project.eu/virtuoso/api/infoAltmetric?doi=SOURCE_DOI)

- RESPONSE (application/json)

The response is represented by a JSON object containing all information from Altmetric.

### 46.3 Contributors

Checkout detailed instructions [here](#)

Please feel free to contact us any time if you have any questions or comments.

**Authors** Rita [RICCERI](#) - Italian National Institute of Nuclear Physics ([INFN](#)),  
Giuseppina [INSERRA](#) - Italian National Institute of Nuclear Physics ([INFN](#)),  
Carla [CARRUBBA](#) - Italian National Institute of Nuclear Physics ([INFN](#))



---

## TRAINING MATERIALS

---

By definition, a Science Gateway is “a community-developed set of tools, applications, and data that is integrated via a portal or a suite of applications, usually in a graphical user interface, that is further customized to meet the needs of a specific community”.

The Catania Science Gateway Framework (CSGF) is fully based on official worldwide standards and protocols, through their most common implementations. These are:

- The [JSR 168](#) and [JSR 286](#) standards (also known as “portlet 1.0” and “portlet 2.0” standards);
- The [OASIS - Security Assertion Markup Language \(SAML\)](#) standard and its [Shibboleth](#) and [SimpleSAMLphp](#) implementations;
- The Lightweight Direct Access Protocol, and its [OpenLDAP](#) implementation;
- The [Cryptographic Token Interface Standard \(PKCS#11\)](#) standard and its [Cryptoki](#) implementation;
- The [Open Grid Forum \(OGF\) Simple API for Grid Applications \(SAGA\)](#) standard and its [JSAGA](#) implementation;
- The [Open Grid Forum \(OGF\) Open Cloud Computing Interface \(OCCI\)](#) standard and its [rOCCI](#) implementation.

Science Gateways implemented with the CSGF are built on top of the [Liferay](#) portal framework and some useful links about Liferay are listed here for convenience:

- [Liferay](#)
- [Liferay Documentation](#)
- [Liferay Downloads](#)
- [Liferay Community](#)
- [Liferay Wiki](#)

The sections linked below provide information and guidelines about how to install and configure the CSGF development environment as well as some examples of basic template portlets that can be further customised/adapted to integrate specific applications in the Science Gateway.

Note: in order to best profit from these training materials, a good knowledge of the Java programming language is required.

### 47.1 Installation and configuration of the development environment

#### Liferay Bundle Installation

- Download Liferay Bundle 6.1.1-ce-ga2 for Glassfish from [here](#)

- Unzip the Liferay Bundle in the folder you prefer.
- set the LIFERAY\_HOME env variable to the folder containing your Liferay Bundle:

```
export LIFERAY_HOME=/Users/Macbook/Downloads/liferay-portal-6.1.1-ce-ga2
```

- set the executable permission to all binary file in glassfish bin folder:

```
chmod +x glassfish-3.1.2/bin/*
```

- start the domain using the following command:

```
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin start-domain domain1
```

You should have an output like the following:

```

RicMac:bin Macbook$ $LIFERAY_HOME/glassfish-3.1.2/bin/asadmin start-domain domain1
Waiting for domain1 to start
Successfully started the domain : domain1
domain Location:
/Users/Macbook/Downloads/liferay-portal-6.1.1-ce-ga2/glassfish-3.1.2/domains/domain1
Log File:
/Users/Macbook/Downloads/liferay-portal-6.1.1-ce-ga2/glassfish-3.1.2/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.

```

- Open a browser window to <http://localhost:8080/> This procedure will take a while during the first connection. At the end you should get the following interface:

**LIFERAY**

**Basic Configuration**

**Portal**

Portal Name  
Liferay  
*For example, Liferay.*

Default Language  
English (United States) **Change**

**Administrator User**

First Name  
Test

Last Name  
Test

Email (Required)  
test@liferay.com

**Database**

Default Database (Hypersonic)  
This database is useful for development and demo'ing purposes, but it is not recommended for production use. [\(Change\)](#)

**Finish Configuration**

Powered By [Liferay](#)

- Press the ‘Finish Configuration’ button; it generates the portal’ configuration file:  
/Users/Macbook/Downloads/liferay-portal-6.1.1-ce-ga2/portal-setup-wizard.properties
- Press the ‘Go My Portal’ button, agree the conditions, set the new password and password retrieval questions.  
After then you’ll be redirected to the Liferay home page.
- To check the Liferay log file:

```
tail -f $LIFERAY_HOME/glassfish-3.1.2/domains/domain1/logs/server.log
```

- References: [Official instructions](#) (the example is for Tomcat only)

### MySQL - Installation and Configuration

In case you have already a MySQL server in your system, you can skip this step just verifying that your version is < 5.6 due to an incompatibility issue between newer MySQL versions and the jdbc-connector.jar library provided with the current version of Liferay bundle.

- Install [MySQL](#) (MySQL Community Server).

You could skip the subscription to the ORACLE Web Login.

### DB\_MACOSX:

Instructions are available inside the README.txt file.

Select the DMG file and execute the two pkgs icons from the terminal.app execute:

```
sudo /Library/StartupItems/MySQLCOM/MySQLCOM start
(your password will be requested)
```

Add the PATH to the .profile:

```
export PATH=$PATH:/usr/local/mysql/bin
```

Start the service

```
RicMac:liferay-portal-6.1.1-ce-ga2 Macbook$ sudo /Library/StartupItems/MySQLCOM/MySQLCOM start
Password:
Starting MySQL database server
```

### DB\_LINUX:

On L5/6 it is possible to install MySQL with:

```
yum install mysql-server
```

Then the follow commands will enable mysql to start at boot and startup the mysql daemon process

```
chkconfig mysqld on
/etc/init.d/mysqld start
```

- generate the portal-ext.properties file:

```
cat <<EOF > $LIFERAY_HOME/portal-ext.properties
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.url=jdbc:mysql://localhost/lportal?useUnicode=true&characterEncoding=UTF-8&useFastDateP
jdbc.default.username=liferayadmin
jdbc.default.password=liferayadmin
EOF
```

- create Liferay database

```
mysql -u root
CREATE USER 'liferayadmin' IDENTIFIED BY 'liferayadmin';
CREATE DATABASE lportal;
GRANT ALL PRIVILEGES ON lportal.* TO 'liferayadmin'@'localhost' IDENTIFIED BY 'liferayadmin';
```

- Download the mysql-connector from [here](#) and copy it in \$LIFERAY\_HOME/glassfish-3.1.2/domains/domain1/lib/

**\*!** Restart Liferay; this will cause Liferay to identify the DB and create new tables and data.

```
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin stop-domain domain1 && \
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin start-domain domain1
```

### **Liferay Plugins SDK**

- Download the SDK from [here](#) (Liferay Plugins SDK 6.1 GA 2).

You may try clicking [here](#)

- Open the file LIFERAY\_SDK\_HOME/build.properties, uncomment ‘glassfish’ settings and setup the proper file path values. Comment out the default enabled tomcat settings.
- Pay attention that in LIFERAY\_SDK\_HOME/build.properties there are also settings to specify which java compiler will be used by ant; in case of troubles try to setup properly the ‘javac.compiler’ option; for instance switchin to ‘modern’ value.
- Be sure your system has installed ‘ant’ and ‘ecj’ otherwise install them.
- A small test could be the use of:

```
cd $LIFERAY_SDK_HOME/portlets/
./create.sh hello-world "Hello-World"
```

Pay attention that the create.sh file normally does not have enabled the execution permission

```
chmod +x ./create.sh
```

- This should create the ‘hello-world’ portlet folder.
- Enter in hello-world-portlet folder:

```
cd hello-world-portlet
```

- Excute deploy command

```
ant deploy
```

- Liferay log file should contain some lines like this:

Successfully autodeployed :

```
LIFERAY_HOME/glassfish-3.1.2/domains/domain1/autodeploy/hello-world-portlet.[#]
```

## Grid Engine

```
Stop Liferay
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin stop-domain domain1
```

- To create the database and the tables;  
download from [here](#) the UsersTrackingDB.sql file and execute:

```
mysql -u root < UsersTrackingDB/UsersTrackingDB.sql
```

In case the users tracking database already exists, uncomment the line:

```
-- drop database userstracking;
```

Pay attention the line above will destroy the existing database.

- Download Grid Engine and JSAGA libraries from sourceforge and copy them in temporary folder:

```
#
Use curl <namefile> > <namefile> in case you do not have wget
#
wget http://sourceforge.net/projects/ctsciencegtwys/files/catania-grid-engine/1.5.9/Liferay6.1/GridEngine_v1.5.9.zip
```

- Unzip the GridEngine\_v1.5.9.zip inside the temporary folder:

```
unzip GridEngine_v1.5.9.zip
```

- Move the config file from the temporary folder to the Liferay config folder:

```
mv <temp folder path>/GridEngine_v1.5.9/GridEngineLogConfig.xml $LIFERAY_HOME/glassfish-3.1.2/domains/domain1/config
```

- Move all the other files to the Liferay lib folder

```
mv <temp folder path>/GridEngine_v1.5.9/* $LIFERAY_HOME/glassfish-3.1.2/domains/domain1/lib
```

- Startup liferay

```
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin start-domain domain1
```

- If you are using a virtual machine, be aware that Glassfish control panel access normally is forbidden from remote. Following commands are necessary to enable it:

```
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin --host localhost --port 4848 change-admin-password
$LIFERAY_HOME/glassfish-3.1.2/bin/asadmin enable-secure-admin
```

Please refer to the [Glassfish Administration Guide](#) for more details

## **EUGRIDPMA and VOMSDIR**

Each access to any distributed infrastructure requires well defined authentication and authorization mechanisms.

Most of Grid infrastructures are making use of the [GSI](#). This security mechanism relies on X509 digital certificates provided by entities named Certification Authorities which themselves are using X509 certificates.

The CAs are normally registered by the [IGTF](#) a body to establish common policies and guidelines between its Policy Management Authorities (PMAs). The CAs act as an independent trusted third party for both subscribers and relying parties within the infrastructure.

In order to setup CA certificates, it is necessary to perform one of the following instructions. RPM based Linux distributions may try the first approach (Linux systems); the othe platforms must use the second approach (Other systems).

- Linux systems

On linux systems it is possible to install the IGTF CA certificates executing the following steps:

- Other systems (MacOSx):

Execute the following instructions to create the `/etc/grid-security/certificates` and `/etc/grid-security/vomsdir` folders:

```
sudo mkdir -p /etc/grid-security
curl http://grid.ct.infn.it/cron_files/grid_settings.tar.gz > grid_settings.tar.gz
sudo tar xvfz grid_settings.tar.gz -C /etc/grid-security/
```

(!) Archives below will expire timely so that they should be kept updated

(!!) vomsdir must be updated with VO you are going to support

## **VPN Setup to get the access to the eTokenserver**

The eToken server is the responsible to deliver grid proxy certificate to the GridEngine starting form Robot Certificates stored into an eToken USB key.

For security purposes is not possible to access directly the eTokenServer. For portlet developers it is possible to open a VPN connection.

In order to get the necessary certificates you have to send us a [request](#)

The VPN connection information will be released in OpenVPN format, together with the necessary certificate and a password.

For Mac users we may suggest [Tunnelblick for MacOSX platforms](#).

There is also this [video](#) showing how to setup the VPN from the configuration files sent by us. For other platforms like Linux we suggest to install [OpenVPN](#) client and then execute from the same directory holding the certificate:

```
openvpn --config <received_conf_file>.ovpn
```

Please notice that on CentOS7 VPN will not work by default since provided VPN certificates are encrypted using MD5 and SHA1 which are no longer supported on CentOS 7. To be able to use the VPN certificate anyway it is possible to enable Md5 support on CentOS7;

just executing as root:



```
cat >> /usr/lib/systemd/system/NetworkManager.service <<EOF
[Service]
Environment="OPENSSL_ENABLE_MD5_VERIFY=1 NSS_HASH_ALG_SUPPORT=+MD5"
EOF
systemctl daemon-reload
systemctl restart NetworkManager.service
```

Further details about this issue are available [here](#) (Thanks to [Manuel Rodriguez Pascual](#))

## Development

### WARNING

For architectural reasons the constructor of GridEngine object must be declared differently than the portlet code written for the production environment

**The constructor must be created with:**

```
MultiInfrastructureJobSubmission multiInfrastructureJobSubmission = new MultiInfrastructureJobSubmission
("jdbc:mysql://localhost/userstracking", "tracking_user", "usertracking");
```

In the portlet examples the constructor call lies inside the *submitJob* method

### Integrated Development Environment (IDE)

We recommend [NetBeans](#) as IDE to develop portlets and other Liferay plugins. In order to create Liferay plugins you can use the [Plugin Portal Pack](#) extension of NetBeans or configure the plugin to use the Liferay SDK

### References

[Liferay Plugin SDK - How to](#)

[Plugin Guide](#)

## 47.2 Creation of a simple parameterised portlet

This complete example of portlet code contains anything you need to develop your own first portlet.

For this reason developers can use it as a template source code to customize according to their own specific requirements.

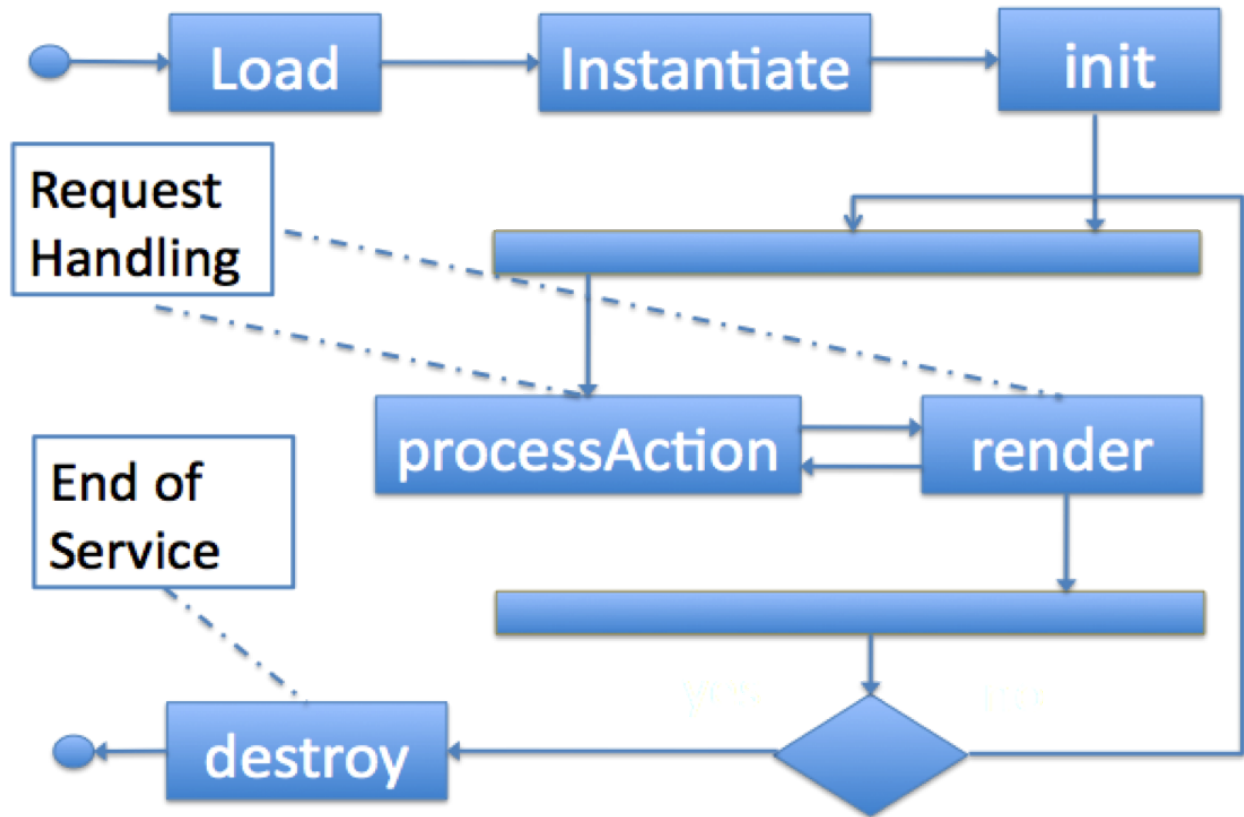
Following instructions provide a step by step guide explaining how to customize the template in order to obtain in the fastest possible way a full featured web application.

### Portlet Workflow

Before to start with the portlet template it is important to understand the internal workflow of a standard portlet (JSR168/286).

The picture below well depicts the entire workflow among the different portlet components.

Components in the figure are simply class methods of the GenericPortlet Java class provided by the portlet SDK.



```

Class GenericPortlets{
init(PortletConfig);
processAction(ActionRequest, ActionResponse);
render (RenderRequest, RenderResponse);
destroy();
doView (Request, Response);
doEdit (Request, Response);
doHelp (Request, Response);
}

```

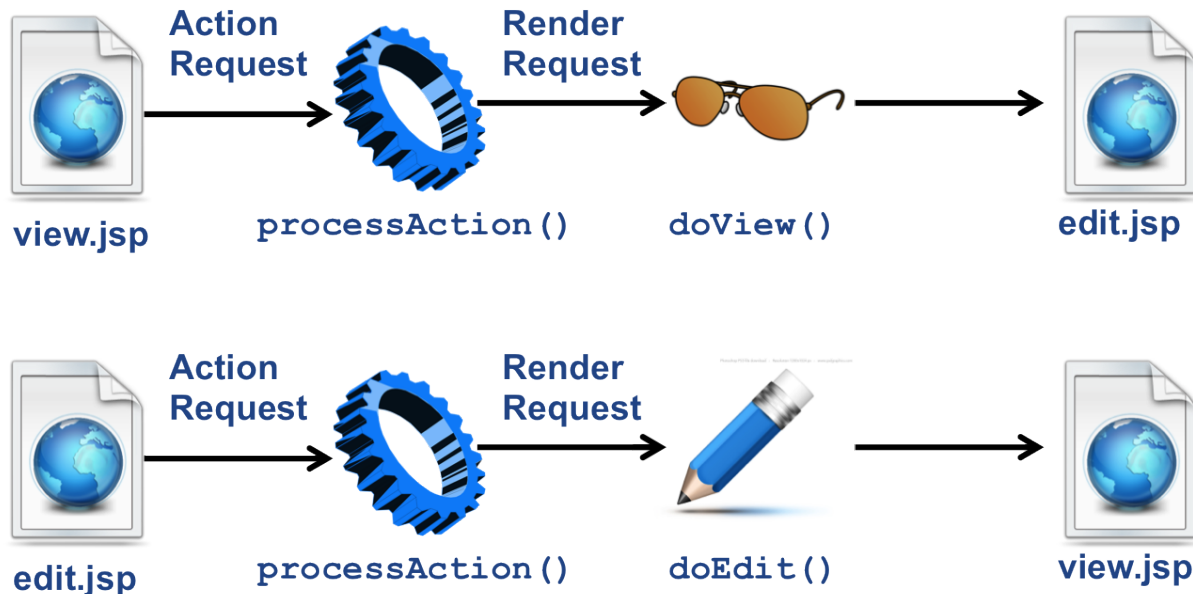
- The above figure depicts the whole lifecycle of a portlet; the most important loop foresees the exchange between the **ProcessAction** and **Render** methods, respectively responsible of the action selected by the user in the input forms and then the interface to show back to the user as consequence of the user action.

### Portlet Modes

Standard portlets operate in 3 different modes: **VIEW**, **EDIT**, **HELP**

- **VIEW**: generates the normal user interface
- **EDIT**: used to store portlet preferences
- **HELP**: show usage instruction

The **Render** method is responsible to call a different **GenericPortlet** class method accordingly to the current portlet mode as shown in the figure:



**Render** method will call then **GenericPortlet** methods: **doView**, **doHelp**, **doEdit**. Each method is responsible to present the appropriate user interface accordingly to the user action and portlet status.

### Data Exchange between Java and JSP pages

During the user interaction there is a continuous data exchange between the **GenericPortlet** class and the **JSP** pages responsible of the user interface presentation. Following paragraphs show how exchange data between jps pages and the Java code.

#### JSP -> Java

Inside the JSP code place all JAVA' input fields into a web form:

```

<form action="<portlet:actionURL portletMode="view">
<portlet:param name="param_name_1" value="paramvalue 1" />
...
<portlet:param name="param_name_n" value="paramvalue n"/>
...
<input ... />
<input ... />

<input type="submit" ... />
</form>

```

Inside the JAVA code get the input interface values with:

```

doView/doHelp/doEdit(RenderRequest request,...
// To obtain the parameter just set ...
String param_i= request.getParameter("param_name_i");

```

#### Java -> JSP

Inside the JAVA code get the input interface values with:

```
doView/doHelp/doEdit(RenderRequest request,...
// To obtain the parameter just set ...
String param_i= request.setAttribute("param_name_i", "param_value_i");
```

Inside the JSP page load parameter values with:

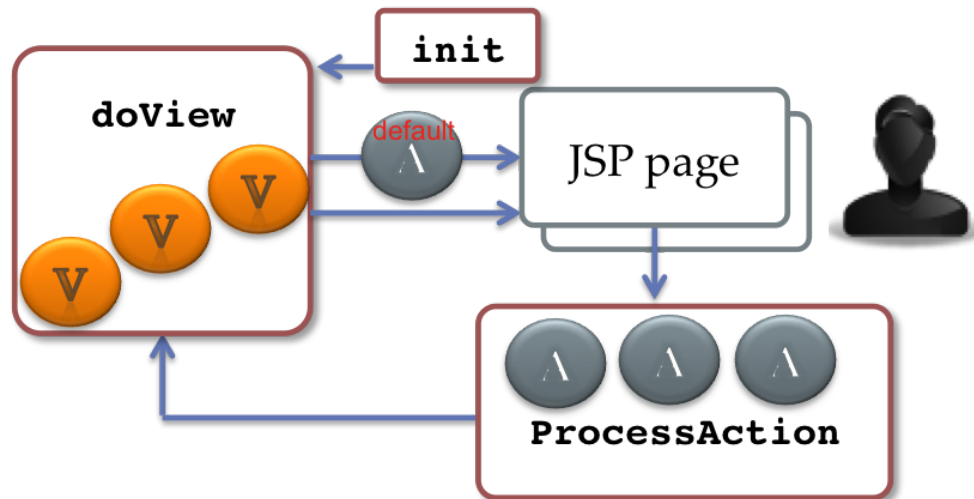
```
<%
// To load variables from PortletClass ...
%>
<jsp:useBean id="param_name_k" class="<variable type k>" scope="request"/>

<%
// To reference a paramvalue
%>
```

Reference parameter\_name' value with: `<%=param_name_k%>`

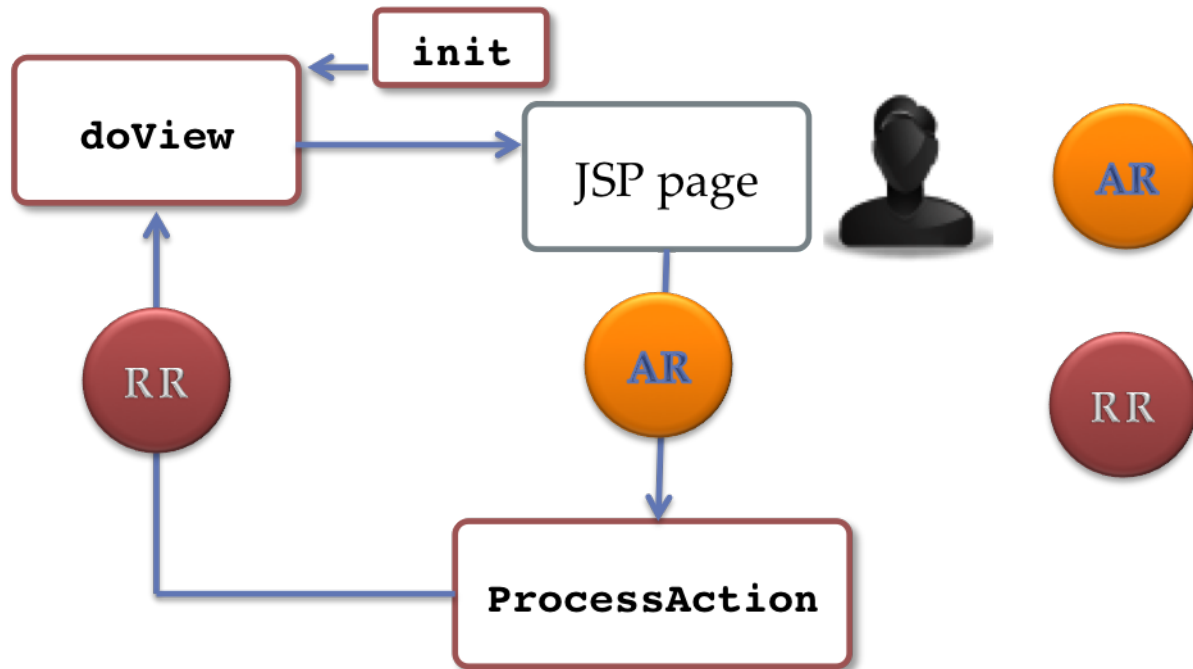
### GenericPortlet main workflow

The following picture shows the internal workflow inside the GenericPortlet class while the user interacts with the WebApplication:



The loop starts with the **Init()** method then the entire workflow plays around the methods **ProcessAction** and **doView** (assuming the VIEW mode). For each User Action a different View will be selected

During this loop two important object instances are used to exchange data between the **doView** and **processAction** methods as shown below:



**actionRequest** input of processAction method which prepares the render object for view methods **renderRequest** input of View methods: **doView/doHelp/doEdit**

### Deploy myFirstPortlet

In this section we can see how the steps that you have to follow to deploy the **myFirst-portlet** in your liferay bundle installation.

1. Move in your Liferay plugin SDK portlets folder

```
cd $LIFERAY_SDK_HOME/portlets/
```

2. Download myFirst-portlet source code through svn command:

```
svn checkout svn://svn.code.sf.net/p/ctsciencegtwys/liferay/trunk/gilda/myFirst-portlet
```

3. Move into myFirst-portlet/ folder
4. Deploy portlet with the following command (and see Liferay log):

```
ant deploy
```

If built process complet successfully , you can see in Liferay Log somethins like this:

```
Successfully autodeployed : LIFERAY_HOME/glassfish-3.1.2/domains/domain1/autodeploy/myFirst-portlet.
```

5. Open web browser at <http://localhost:8080>, click on Add > More > CataniaSG > myFirst-portlet.

myFirst-portlet

🔍 - + ✕



This is an example portlet!  
Exchange data between jsp page and java code

Input value

Go a new JSP page without Action

### Customize myFirstPortlet

This section describes the steps to create a new portlet from the template provided by myFirst-portlet.

- Move into Liferay plugin SDK portlets folder
- Copy myFirst-portlet folder in your\_portlet\_name-portlet

```
cp -R myFirst-portlet your_portlet_name-portlet
```

- Move into your\_portlet\_name-portlet folder
- Edit the customize.sh file, set the following parameters as you prefer:

– AUTH\_EMAIL= your@email

—AUTH\_NAME= your name

– AUTH\_INSTITUTE= your\_institute

Pay attention: the APP\_NAME value must be set with the name that you assigned in your portlet folder name

- APP\_NAME= your\_potlet\_name
- Run customize.sh script, with

```
./customize.sh
```

- Then delpoy portlet with ant deploy

To see the result follow step 5 in previous section changing “myFirst-portlet” with “your\_potlet\_name-portlet”

### Web application editors

This is the right moment to create a project using a high level web application editor like NetBeans or Eclipse.

Following instructions are valid for **NetBeans**

- Download Netbeans IDE
- Open **New Project** > Java Web > Web Application with **Existing Sources** and press ‘Next’;
- In Location browse the “your\_potlet\_name”-portlet directory and press ‘Next’;
- Accept any suggestion and proceed and press ‘Next’;
- Add other directory places;

WEB-INF Content: Select the docroot/WEB-INF directory inside the your\_potlet\_name-portlet directory;

- Then press the ‘Finish’ button and the project will be created
- Right click on the project name and click on Preferences, then Libraries.
- Select all jars pointed by

```
$LIFERAY_HOME/glassfish-3.1.2/domains/domain1/lib
```

(in your liferay bundle)

Following instructions are valid for **Eclipse**

- Download Eclipse IDE for java EE Developers;
- Set the Eclipse Workspace to the “portlets” \$LIFERAY\_SDK\_HOME/portlets/ directory;
- Select File > New > Web > Dynamic Web Project and press ‘Next’

Fill the Web Dynamic Web Project Wizard with

- the project name: your\_potlet\_name-portlet;
- the default location only if the default one is not correct;>
- the glassfish target runtime (if doesn’t exist create a new one with the New Runtime... wizard);
- leave the default values for Dynamic Web module\_vesion and Configuration fields and press ‘Next’;
- Change the Content Directory to “docroot”;
- Change the Java Source Directory to “docroot/WEB-INF/src” and press ‘Finish’;
- **In order to fix some library dependencies could be necessary add external Jars.** Right click on the project name and click on “Properties” > Java Build Path > Libraries and select all jars pointed by \$LIFERAY\_HOME/glassfish-3.1.2/domains/domain1/lib (in your liferay bundle)

Start to develop the interface modifying jsp files and change java code enums with correct Actions and Views modese with the proper identifiers. For simple user interfaces there will be no need to add other jsp or action/view modes.

## 47.3 Creation of a portlet that submits a job

This page shows a complete example of portlet code contains anything you need to develop your own portlet to submit and run a sequential application.

Following instructions provide a step by step guide explaining how to just deploy a portlet that submit and run a sequential Grid applications in a distributed Grid infrastructure. Then shows the steps that developers can follow to customize the template in order to obtain in the fastest possible way a full featured web application able to submit jobs in a distributed Grid infrastructure.

To correctly execute following steps, you must have successfully completed the “To correctly execute the following steps, you must have successfully completed the “Installation and configuration of the development environment” tutorial.

It is also highly recommended that you have followed the previous tutorial in how to create your first parameterized portlet, in order to understand the portlet workflow.

### Deploy mi-hostname-portlet

This section explains how to deploy mi-hostname-portlet that allows you to submit and run a sequential Grid applications in a distributed Grid infrastructure.

**Steps to deploy the portlet:**

First of all make sure that your liferay is correctly up and running, then:

- Move to your Liferay plugin SDK portlets folder and get the mi-hostname-portlet through svn command:

```
svn checkout svn://svn.code.sf.net/p/ctsciencegtwys/liferay/trunk/gilda/mi-hostname-portlet mi-hostname-portlet
```

- Now move into the just created mi-hostname-potlet directory and execute the deploy command:

```
ant deploy
```

When previous command has completed, verify that if the portlet is “Successfully autodeployed” you can see this in your Liferay log file.

- Then open your browser at <http://localhost:8080> click Add > More you should have the new GILDA menu, click on it and then Add this new portlet. following picture shows the correct result:



mi-hostname-portlet

✎ - + ✕



Please fill the following form and then press '**SUBMIT**' button to launch this application.

Requested inputs are:

- An input file (its text or a file to upload)
- Any other input ...

Pressing the '**Demo**' Button input fields will be filled with Demo values.

Pressing the '**Reset**' Button all input fields will be initialized.

Pressing the '**About**' Button information about the application will be shown

Application' input file  Nessun file selezionato

Insert here your text file, or upload a file

Insert below your **job identifier**

multi-infrastructure job description





Now you should be able to submit your first sequential job on a distributed Grid infrastructure, insert an example text into the shown text area and a brief example description, then click on “Submit” button.

### Install MyJob portlet

To check the job status and to retrieve the output when job is done, you should install our MyJob portlet, in order to do this you have to make some configuration in your liferay environment.

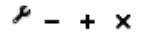
- Open the Glassfish Administration Console (<http://localhost:4848>).
- Create a new JDBC connection pool for MyJob portlet:

- On the left menu select Resources > JDBC > JDBC Connection Pools
- Click New... to create a new pool with the following settings:
  - \*\* Pool Name: usertrackingPool
  - \*\* ResourceType: javax.sql.DataSource
  - \*\* Database Driver Vendor: select MySql
- Click Next and left the default parameters;
- Select and remove all the properties from the “Additional Properties” table (bottom page);
- Click on “Add” and create the following three properties:
  - Name: Url, Value: jdbc:mysql://localhost:3306/usertracking
  - Name: User, Value: tracking\_user
  - Name: Password, Value: usertracking
- Click on “Finish” button to save configuration.
- Click on the ‘Ping’ button to test the connection pool. If everything is working fine the “Ping Succeeded” message should appear on top.
- Create a new JDBC Resource:
  - On the left menu select Resources > JDBC > JDBC Resources
  - Click New... to create a new JDBC Resource with the following settings:
    - \*\* JNDI Name: jdbc/UserTrackingPool
    - \*\* Pool Name: select usertrackingPool
  - Click on “Finish” button to save configuration.
- Restart Glassfish

When restart procedure has completed you can proceed with the installation of the MyJob portlet.

- Download MyJob.war
- Move the downloaded WAR file in the deploy folder under your Liferay bundle installation, see Liferay log until deploy process complete successfully (“Successfully autodeployed”)
- Open your browser at <http://localhost:8080> click Add > More > INFN > MyJob > Add

Now you should see the status of the job that you have submitted previously, see picture below

**MyJobs**

Active Jobs List Done Jobs List

The table below shows the status of your jobs.

Statuses are automatically updated every 15 minutes so there is no need to reload this page more frequently. However, if you don't see your jobs in the table within a reasonable amount of time (a couple of hours at most), click on Help in the the MyWorkspace portlet and notify us the problem.

Once your jobs have finished, you have 96 hours to retrieve their output. Beyond that time, the output of your jobs will automatically be deleted from the Science Gateway in order not to fill its storage with undesired stuff.

Copy Print Save Download Job output

Search:

Show 10 entries

First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	liferay.com	Test	2013-07-04 14:21:29.0	RUNNING

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

When Status column from RUNNING becomes you can download the job output by clicking on this icon .. image:: figures-and-documents/figure15.png

#### Customize mi-hostname-portlet

This section describes the steps to create a new portlet from the template provided by mi-hostname-portlet.

- Move into Liferay plugin SDK portlets folder
- Copy mi-hostname-portlet folder in <your\_portlet\_name>-portlet cp -R mi-hostname-portlet <your\_portlet\_name>-portlet
- Move into <your\_portlet\_name>-portlet folder
- Edit the customize.sh file, set the following parameters as you prefer:
 

```
AUTH_EMAIL=<your@email>
AUTH_NAME='<your name>'
AUTH_INSTITUTE='<your_institute>'
```

Pay attention: the APP\_NAME value must be set with the name that you assigned in your portlet folder name

```
APP_NAME=<your_potlet_name>
```

- Run customize.sh script, with ./customize.sh
- Then deploy portlet with ant deploy (see Liferay log file).

## 47.4 Creation of a portlet that submits a special job

This page contains what you need to develop your own portlet to submit and run special jobs.

You can choose the kind of parallel job you would like to run from a list containing the following elements:

**Job Collection:** is a simple parallel application that spawns N sub-jobs; when all these are successfully completed the whole collection becomes DONE.

**Workflow N1:** is a parallel application that spawns N sub-jobs, waits until all these are correctly completed and then submits a new job whose input files are the outputs of the N sub-jobs. When also this “final job” is successfully executed, the whole Workflow N1 becomes DONE.

**Job Parametric:** is a parallel application that spawns N sub-jobs with the same executable and with different arguments (i.e., input parameters); when all these are successfully completed the whole parametric job becomes DONE.

The following instructions show how to deploy this exemplar portlet, how to use it to submit the above mentioned parallel jobs and how to customize the code to reuse it to develop your own portlets.

Deploy mi-parallel-app-portlet

This section explains how to deploy mi-hostname-collection-portlet that allows you to submit and run a special parallel job on a Distributed Computing infrastructure.

### Steps to deploy the portlet:


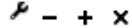
First of all, make sure that your Liferay server is correctly up and running, then:


- Move to your Liferay plugin SDK portlets folder and get the mi-hostname-collection-portlet through svn command:  

```
svn checkout svn://svn.code.sf.net/p/ctsciencegtwys/liferay/trunk/gilda/mi-parallel-app-portlet mi-parallel-app-portlet
```
- Now, move into the just created mi-parallel-app-potlet directory and execute the deploy command:  

```
ant deploy
```

When previous the command has completed, verify that if the portlet is “Successfully autodeployed” you can see this in your Liferay log file.
- Then, open your browser at <http://localhost:8080> click Add > More in the GILDA menu, click on Add button to add this new portlet. following picture shows the correctly result:

 **mi-hostname-collection-portlet** 



Please fill the following form and then press '**SUBMIT**' button to launch this application.  
Requested inputs are:

- **Task number**: number of sub-jobs that belong to the Job Collection
- **Insert executable *f***: the executable you would like run on the distributed infrastructure

Pressing the '**Demo**' Button input fields will be filled with Demo values.  
Pressing the '**Reset**' Button all input fields will be initialized.  
Pressing the '**About**' Button information about the application will be shown

Select collection type: Job Collection ▾

Insert task number:  OK

Insert below your **collection identifier**

Demo Submit Reset values  
About

Now you should be able to submit your first parallel job on a distributed Grid infrastructure, insert an example text and add a brief example description, then click on the “Submit” button.

To check the job status and to retrieve the output when the collection will be DONE, you should use the MyJob portlet, if you haven’t already installed MyJob portlet you can find instruction to install it here.

#### Perform a parallel application

- Select the collection type from the ComboBox:

Select collection type: Job Collection ▾

- Insert the number of tasks that compose this collection:

Insert task number:  OK

- Clickin on the “OK” button, the page will be automatically updated with a number of input text fields equal to the number of tasks entered, fill these input text fields with some command, like hostname, ls, echo, etc. Optionally, you can also specify arguments for that commands into the relatives text filesd.

## Sub Jobs

### Executables

### Arguments


The picture is showing the result if you insert 3 into the above mentioned task number field.

- Now, insert a collection identifier

**Insert below your collection identifier**

**multi-infrastructure-collection job description**

- Finally, click on the Submit button to execute this collection.

Now move to the MyJob portlet and if all went well, this is the result that you should see:

**MyJobs**

Active Jobs List Done Jobs List

The table below shows the status of your jobs.

Statuses are automatically updated every 15 minutes so there is no need to reload this page more frequently. However, if you don't see your jobs in the table within a reasonable amount of time (a couple of hours at most), click on Help in the the MyWorkspace portlet and notify us the problem.

Once your jobs have finished, you have 96 hours to retrieve their output. Beyond that time, the output of your jobs will automatically be deleted from the Science Gateway in order not to fill its storage with undesired stuff.

Copy Print Save Download Job output

Search:

Show 10 entries

First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	mi-hostname	Demo Collection	2013-07-22 15:21:51.0	RUNNING
		User Description	Started on (UTC)	Status
		Demo Collection_1	2013-07-22 15:21:51.0	RUNNING
		Demo Collection_0	2013-07-22 15:21:51.0	SUBMITTED
		Demo Collection_2	2013-07-22 15:21:51.0	SUBMITTED

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

When all sub-jobs belonging to job collection have successfully completed you can download the whole job collection output.

Copy Print Save Download Job output

Search:

Show 10 entries

First Previous 1 Next Last

info job	Application Name	User Description	Started on (UTC)	Status
	Job Collections	mi-collection	Demo Collection	2013-07-23 09:52:10.0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Alternatively, you could click on the Demo button that fills input fields with demo values:

task number equal to 3; the following executables: hostname ls pwd

If you select Workflow N1 the executables demo values are the same as previously seen while the final job executable is a “ls” command. Else, if you select Job Parametric the only executable is the “echo” command; in this case the arguments are mandatory and the demo values inserted consist of a string with job index appended.

### Customize mi-parallel-app-portlet

This section describes the steps to create a new portlet from the template provided by the mi-parallel-app-portlet.

- Move into Liferay plugin SDK portlets folder;
- Copy mi-parallel-app-portlet folder to <your\_portlet\_name>-portlet

```
cp -R mi-parallel-app-portlet <your_portlet_name>-portlet
```

- Move into <your\_portlet\_name>-portlet folder;
- Edit the customize.sh file, set the following parameters as you prefer:

```
AUTH_EMAIL=<your@email>
```

```
AUTH_NAME='<your name>'
```

```
AUTH_INSTITUTE='<your_institute>'
```

Attention: the APP\_NAME value must be set with the name that you assigned in your portlet folder name:

```
APP_NAME=<your_potlet_name>
```

- Run customize.sh script, with

```
./customize.sh
```

- Then deploy the portlet with the ant deploy command (and check the Liferay log file).

When the deploy process has completed you can add the new portlet by opening your browser at <http://localhost:8080>, clicking Add > More in the GILDA menu, and the clicking on Add button.