



Enabling Grids for E-science

User Interface (UI) Installation



Giuseppe La Rocca
INFN Catania - Italy

**First Latin American Workshop for Grid
Administrators**

21-25 November 2005

www.eu-egee.org



- **What is a User Interface (UI) ?**
- **How to install a User Interface.**
- **How to configure a User Interface.**

- **The UI is a suite of clients and API that users and applications can use to access the gLite services.**
- **The gLite UI includes the following components:**
 - **Data Catalog command-line clients and APIs**
 - **Data Transfer command-line clients and APIs**
 - **gLite I/O client and APIs**
 - **R-GMA client and APIs**
 - **VOMS command-line tools**
 - **Workload Management System clients and APIs**
 - **Logging and Bookkeeping clients and APIs**

Installing User Interface (UI)



- **Start from a fresh install of SLC 3.0.4**
- **Installation via**
 - **Installer script** (<http://glite.web.cern.ch/glite/packages>)
 - **APT** <http://glite.web.cern.ch/glite/packages/APT.asp>
- **Installation will install all dependencies, including**
 - other necessary gLite modules
 - external dependencies
- **JAVA is not included in distribution. Install it separately ($\geq 1.4.2_06$)**
<http://java.sun.com/j2se/1.4.2/download.html>

- Request host certificates for the UI.
 - <https://gilda.ct.infn.it/CA/mgt/restricted/srvreq.php>
- Install host certificate (hostcert.pem and hostkey.pem) in **/etc/grid-certificates**.
 - *chmod 644 hostcert.pem*
 - *chmod 400 hostkey.pem*
- If planning to use certificates released by unsupported EGEE CA's, be sure that their public key and CRLs (usually distributed with an rpm) are installed.
 - The CRL of the VO GILDA are available from https://gilda.ct.infn.it/RPMS/ca_GILDA-0.28.1.i386.rpm

1. Verify if apt is present:

- `rpm -qa | grep apt`
- Install apt if necessary:
 - `rpm -ivh http://linuxsoft.cern.ch/cern/slc30X/i386/SL/RPMS/apt-0.5.15cnc6-8.SL.cern.i386.rpm`

2. Add gLite apt repository:

- Put one this line in a file (e.g. `glite.list`) inside the `/etc/apt/sources.list.d` directory (R 1.4)
- `rpm http://glitesoft.cern.ch/EGEE/gLite/APT/R1.4/rhel30 externals Release1.4 updates`
- `apt-get update`
- `apt-get upgrade`

3. Install User Interface:

- `apt-get install glite-ui-config`

See <http://glite.web.cern.ch/glite/packages/APT.asp>

- If the installation is performed successfully, the following components are installed:
 - *gLite* in */opt/glite*
 - *Globus* in */opt/globus*
 - *GPT* in */opt/gpt*

- Configuration comes through the execution of python scripts, which takes as input XML files.
- So services have to be configured by editing these XML files.
- Attributes in XML file are well commented and self-explaining.
- XML files are provided as templates, under **`/opt/glite/etc/config/templates`**
- Copy templates file to **`/opt/glite/etc/config`**
- Edit each of them separately.
- Then we could launch the configurator scripts for the User Interface.

- **List of XML files to configure:**

glite-global.cfg.xml

glite-security-utils.cfg.xml

glite-rgma-common.cfg.xml

glite-rgma.client.cfg.xml

glite-io-client.cfg.xml

glite-file-transfer-service-client.cfg.xml

glite-service-discovery.cfg.xml

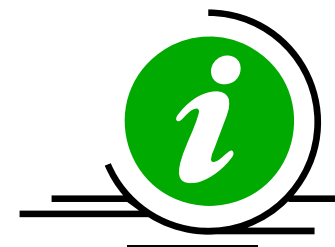
glite-service-discovery.file-based-example.cfg.xml

glite-ui.cfg.xml

```
<JAVA_HOME description="Environment variable  
pointing to the SUN Java JRE or J2SE package for  
example '/usr/java/j2re1.4.2_08/' or '$JAVA_HOME' (if  
it is defined as an environment variable)"  
value="/usr/java/j2re-1.4.2_06"/>
```



Check your java package installed.



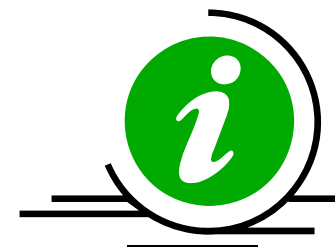
- Set the parameters to correctly build files needed by GSI.
- Enable fetch-crl cron-job

`<install.fetch-crl.cron`

`description="Install the glite-fetch-crl cron job.`

`Possible values are 'true' (install the cron job) or 'false' (do not install the cron job)"`

`value="true"/>`



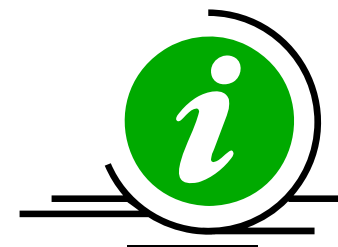
- Enable glite-mkgridmap cron-job.

```
<install.mkgridmap.cron
```

```
  description="Install the glite-mkgridmap cron job and  
  run it once.
```

```
  Possible values are 'true' (install the cron job) or 'false'  
  (do not install the cron job)"
```

```
  value="true"/>
```



<rgma.server.hostname

description="Host name of the R-GMA server.

[Example: lxb1420.cern.ch] [Type: 'string']"

value="rgmasrv.ct.infn.it"/> 

<rgma.schema.hostname

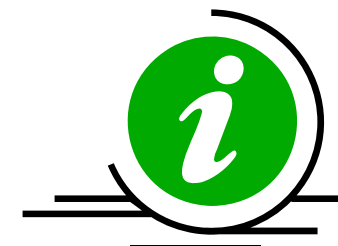
description="Host name of the R-GMA schema service.

(See also configuration parameter 'rgma.server.run_schema_service'

in the R-GMA server configuration file in case you install a server).

[Example: lxb1420.cern.ch] [Type: 'string']" 

value="rgmasrv.ct.infn.it"/>



<rgma.registry.hostname

description="Host name of the R-GMA registry service.

You must specify at least one hostname and you can specify several if you want to use several registries.

(See also configuration parameter 'rgma.server.run_registry_service' in the R-GMA server configuration file in case you install a server).

[Example: lxb2029.cern.ch] [Type: string]

<value>rgmasrv.ct.infn.it</value>

</rgma.registry.hostname>



<service-discovery.file.service_name

description="The globally unique name of the service.

The convention is

serviceHostName_voName_serviceType (for serviceType see parameter service-discovery.file.service_type)."

value="gildav-ioserver_gildav_org.glite.GliteIO"/>

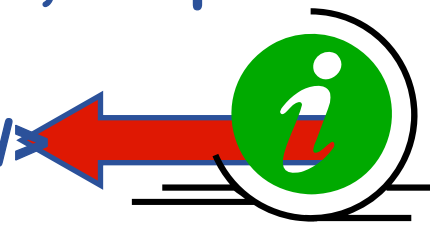


<service-discovery.file.url_endpoint

description="URL endpoint of the service.

The host name is the name of your io server, the port depends on your vo."

value="gliteio://egee015.cnaf.infn.it:9999"/>




```
<instance name="FileTransferService for gildav"
  service="service-discovery.file">
```

```
<parameters>
```

```
<service-discovery.file.service_name
  description="The globally unique name of the
  service."
```

```
value="gildaFts"/>
```

```
<service-discovery.file.url_endpoint
  description="URL endpoint of the service."
```

```
value="https://fts.ct.infn.it:8443/gildav/glite-data-
transfer-fts/services/FileTransfer"/>
```



```
<service-discovery.file.service_name  
value="gildachannel"/>
```



```
<service-discovery.file.url_endpoint  
value="https://fts.ct.infn.it:8443/gildav/glite-data-  
transfer-fts/services/ChannelManagement"/>
```



<service-discovery.type

description="Service discovery implementation to be used. Possible values are:

file use (static) file base service discovery

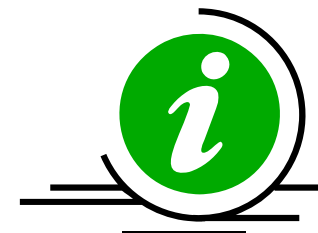
rgma use (dynamic) R-GMA based service discovery

bdii use (dynamic) BDII based service discovery">

<value>file</value>



</service-discovery.type>



```
<service-discovery.bdii.provider
```

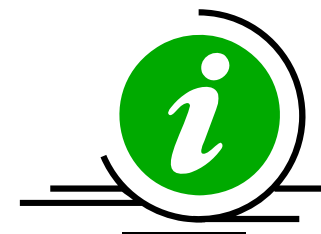
```
  description="Host and port of the BDII service for  
  service discovery.
```

Leave empty or remove parameter if you do not use
BDII as information provider.

Example: lxb1386.cern.ch:2170

```
[Type: 'string']"
```

```
value="grid004.ct.infn.it:2170"/>
```



```
<instance name="FileTransferService for gildav"
  service="service-discovery.file">
```



```
<parameters>
```

```
<service-discovery.file.service_name
```

```
description="The globally unique name of the service.
The convention is serviceHost_voName_serviceType.
serviceType is the value defined in the parameter
service-discovery.file.service_type."
```

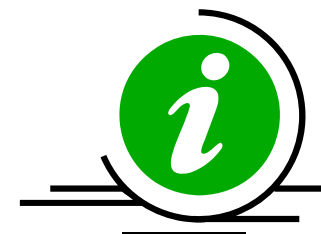
```
value="gildaFts"/>
```



```
<service-discovery.file.url_endpoint  
description="URL endpoint of the service.
```

```
Example:https://your.serviceHostname.com:8443/your  
_service [Type: 'string']"
```

```
value="https://fts.ct.infn.it:8443/gildav/glite-data-  
transfer-fts/services/FileTransfer" ←
```



```
<ui.voms.server
```

```
  value="cert-voms-01.cnaf.infn.it"/>
```

```
<ui.voms.port
```

```
  value="15008"/>
```

```
<ui.voms.cert.subject
```


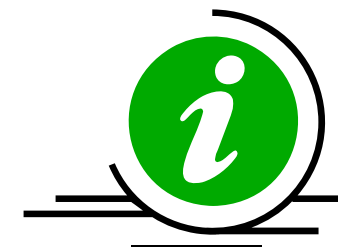
```
  value=/C=IT/O=INFN/OU=Host/L=CNAF/CN=cert-  
voms-01.cnaf.infn.it/>
```



```
<py-ui.DefaultVo
  value="gildav"/>
```



```
<py-ui.requirements
  description='Requirements for matchmaking.
  [Example: other.GlueCEStateStatus == "Production"]
  [Type: "string"]
  value='other.GlueCEStateStatus == "Production"/>
```

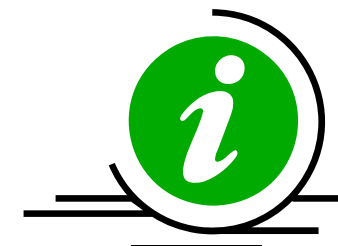

```
<instance name="FiremanCatalog for gildav"
  service="service-discovery.file">
```



```
<service-discovery.file.service_name
value="grid017.ct.infn.it_gilda_org.glite.FiremanCatalog"/
>
```



```
<service-discovery.file.url_endpoint
value="https://grid017.ct.infn.it:8443/gilda/glite-data-
catalog-service-fr-mysql/services/FiremanCatalog"/>
```



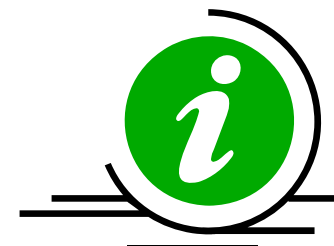
```
<instance name="SEIndex for gildav" service="service-  
discovery.file">
```



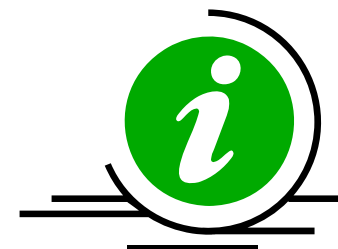
```
<service-discovery.file.service_name  
value="gilda-se.ct.infn.it_gilda_org.glite.SEIndex"/>
```



```
<service-discovery.file.url_endpoint  
value="https://grid017.ct.infn.it:8443/gilda/glite-data-  
catalog-service-fr-mysql/services/SEIndex"/>
```

- Install the GILDA's VOMS server host certificates *gildav-cert-voms-01.cnaf.infn.it.pem* in the directory ***/etc/grid-security/vomsdir***
- Edit the ***/opt/glite/etc/vomses*** file as follow:
 "gildav" "cert-voms-01.cnaf.infn.it" "15008"
 "/C=IT/O=INFN/OU=Host/L=CNAF/CN=cert-voms-01.cnaf.infn.it" "gildav"



- In order to commit configuration, execute

```
python /opt/glite/etc/config/script/glite-ui-  
config.py
```

- To assure the correct functionality of the gLite UI it is necessary to add, as root, the following rows in */etc/profile*

```
source /etc/glite/profile.d/glite_setenv.sh
```

```
PATH=$PATH:/opt/glite/externals/myproxy-1.14/bin:/opt/lcg/bin:/opt/lcg/sbin
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/glite/externals/myproxy-1.14/lib:/opt/lcg/lib
```



