

# Table of Contents

<b>Easysamportal of the certification testbed.....</b>	<b>1</b>
Installation.....	1
Installing the publisher.....	1
Database information.....	1
Installing the web portal.....	1
The database.....	1
Configuring the client part.....	3
Testing the installation.....	4
Test submission.....	4
Collection of tests, CVS.....	5
Problems.....	5
Links, tutorials, How-Tos.....	5

# Easysamportal of the certification testbed

## Installation

### Installing the publisher

The rpm installed is =same-query-x.x.x. Creating the database schema:

```
export CVSROOT=:pserver:anonymous@glite.cvs.cern.ch:/cvs/glite
cvs co same
cd same/server/db/scripts
sqlplus username/password@database < create_same_standalone.sql
```

### Database information

The Oracle database is on CERN central database server.

```
username: lcg_cert_sam
password: ask Gergo
database: INT3R
```

### Installing the web portal

Install easysam rpm from:

```
/afs/cern.ch/project/gd/www/gergo/rpms/lcg-sam-server-easyportal-1.0.0-4.noarch.rpm
```

### The database

You can use Valentin's script (`same/server/db/cron/bdii2oracle.py`) to populate the database using the BDII information. In order to work properly first you have to fill out the `REGION`, `SITE` and `VO` tables. Connect to the database and launch the following commands:

```
insert into region (regionid,regionname,REGIONDESCRIPTION) values(1,'CERTTESTBED','Certification
insert into site values(1,1,1,1,'SL3CERTTB1A','1A CERT SITE','Certified','Production','N');
insert into site values(2,1,1,1,'SL3CERTTB1B','1B CERT SITE','Certified','Production','N');
insert into site values(3,1,1,1,'SL3CERTTB2','2 CERT SITE','Certified','Production','N');
insert into vo values(3,'dteam','N');
```

Due to a bug, please create and fill the following table, as well:

```
create table regions(REGIONID NUMBER(38),NAME VARCHAR2(50), REGIONDESCRIPTION VARCHAR2(255));
insert into regions values(1,'CERTTESTBED','Certification testbed');
```

In order the script to work you have to do the followings:

- Install `python-ldap` package:

```
wget wget http://belnet.dl.sourceforge.net/sourceforge/python-ldap/python-ldap-2.0.7.tar.g
tar -xzvf python-ldap-2.0.7.tar.gz
cd python-ldap-2.0.7
```

Edit the `setup.cfg` file, in order to resolve dependency problems, Change

```
requires = python openldap2-client openssl cyrus-sasl2
```

to

```
requires = python openldap-clients openssl cyrus-sasl
```

### Build and install the rpm:

```
python setup.py bdist_rpm
rpm -i ./build/bdist.linux-i686/rpm/RPMS/i386/python-ldap-2.0.7-1.i386.rpm
```

- Install logging package:

```
wget http://www.red-dove.com/logging-0.4.9.6.tar.gz
tar -xzvf logging-0.4.9.6.tar.gz
cd logging-0.4.9.6
python setup.py bdist_rpm
rpm -i build/bdist.linux-i686/rpm/RPMS/noarch/logging-0.4.9.6-1.noarch.rpm
```

- Modify the script at some point. Since we are not using GOC DB, FCR, and other thing, some of the tables should not be filled, so one should modify the `bdii2oracle.py` script. Here is the `ldiff` output of the original and the modified script:

```
<     SELECT NLS_LOWER(sitename), siteid FROM sites WHERE isdeleted='N'""
---
>     SELECT NLS_LOWER(sitename), siteid FROM site WHERE isdeleted='N'""
622,623c622,623
<         set_action("Getting site BDIIs")
<         services += get_site_bdiis(cursor)
---
> #         set_action("Getting site BDIIs")
> #         services += get_site_bdiis(cursor)
636,638c636,638
<         for res_name in ["CE", "SE"]:
<             set_action("Updating %s resources" % res_name)
<             update_resources(cursor, services, res_name.lower())
---
> #         for res_name in ["CE", "SE"]:
> #             set_action("Updating %s resources" % res_name)
> #             update_resources(cursor, services, res_name.lower())
646,648c646,648
<         set_action("Updating list of core nodes")
<         run_query(cursor, clear_core_flag_sql, {})
<         run_query(cursor, set_core_nodes_sql, {})
---
> #         set_action("Updating list of core nodes")
> #         run_query(cursor, clear_core_flag_sql, {})
> #         run_query(cursor, set_core_nodes_sql, {})
```

### Run the script:

```
./bdii2oracle.py -H ldap://lxb2017:2170 -d LCG_CTBSamectb1@devdb10 -vv
```

In case of error try with more `-vvvvv`. If the script doesn't find a node for some reason, you can still insert a node manually into the DB:

```
insert into node values(1,1,12,'N','lxb2018.cern.ch');
```

Further tables has to be set up:

```
create table NodeMaintenance(nodeid NUMBER(38), inMaintenance CHAR(1));
insert into NodeMaintenance select nodeid, 'N' from node;
```

Setting all node as not beeing under maintenance, ensures that during test submission they will pass the filter applied on the list of available nodes.

## Configuring the client part

Check out sam from CVS (see above). Edit `./same/client/etc/same.conf` to be something like:

```
# Default configuration for SAME
[DEFAULT]
# Settings for locations
workdir=%(home)s/.same
logdir=%(same_home)s/var/log
resdir=%(same_home)s/var/results
secresdir=%(same_home)s/var/results-secure
webdir=%(same_home)s/web
cachedir=%(same_home)s/var/cache

# Logging levels:
# CRITICAL, ERROR, WARNING, INFO, DEBUG, NOTSET
# Logging level for the log file
loglevel=INFO
# Logging level for console messages
verbosity=CRITICAL

[sensors]
common_attrs="sitename nodename inmaintenance"
common_filter="type=Production status=Certified"
CE_filter="serviceabbr=CE ismonitored=Y"
gCE_filter="serviceabbr=gCE"
FTS_filter="serviceabbr=FTS"
FTS_attrs="sitename nodename inmaintenance tier"
SE_filter="serviceabbr=SE"
SRM_filter="serviceabbr=SRM"
LFC_filter="serviceabbr=LFC voname=dteam"
host-cert_filter=""

[statuscode]
ok=10
info=20
notice=30
warning=40
error=50
critical=60
maintenance=100

[submission]
vo=dteam
test_timeout=240

[scheduler]
max_processes=10
default_timeout=300
shell=/bin/sh

[webservices]
publisher_wsdl=http://lxb0714.cern.ch:8080/gridview/services/WebArchiver?wsdl
query_wsdl=http://lxb0714.cern.ch:8080/same-query/services/Database?wsdl
# edit the line below if you want to publish to old-style SFT webservice
# sft_publisher_url=http://lxb0714.cern.ch:8083/sft/publishTuple
```

If you add a new test/sensor do not forget to add a variable called

```
<mytest_name>_filter=""
```

to this config file !

## Testing the installation

- Here are some sqlplus command which could be used to test the installation:

- ◆ List all object: `select object_name from user_objects;`
- ◆ List test result: `select * from testdata;`
- ◆ List/check the services: `select * from serviceinstances;`

- Check the portal, go to <https://lxb0714.cern.ch/sam/sam.py>
- Look into the web portal logs:

```
/opt/lcg/same/server/web/log/sam-portal-db.log
/opt/lcg/same/server/web/log/sam-portal.log
```

- The same query log file:

```
/var/log/tomcat5/same-query.log
```

- As a reference it is always worth to have a look to the official SAM portal : <https://lcg-sam.cern.ch/sam/sam.py>

## Test submission

Submit a simple test:

```
cd same/client/bin
./same-exec SE
```

the normal output should be something like:

```
operation: submit
Sensor: SE
Node attrs: ['sitename', 'nodename', 'inmaintenance']
Node filter: {'status': ['Certified'], 'serviceabbr': ['SE'], 'type': ['Production']}
Sensor args:

Matching nodes: 2
Executing prepare script
-----
prepare-FTS script started.
got nodes:      2 nodes.map
publishing test definitions... DONE
-----
Prepare script finished successfully
Executing check script for all nodes
-----
lxb1921.cern.ch: started
lxb1737.cern.ch: started
lxb1921.cern.ch: publishing
lxb1737.cern.ch: publishing
lxb1921.cern.ch: finished
lxb1737.cern.ch: finished
-----
Executing check scripts finished
```

Check the result on the portal.

## Collection of tests, CVS

- The sensors are be collected in the gLite CVS [↗](#)

## Problems

Here are the current problem we are facing:

- When the SAM portal displays the result it displays only the nodes which are supposed to run the type of service which was tested by the sensor, even if the sensor was running on other hosts, as well. To be changed, corrected in the future.

## Links, tutorials, How-Tos

- See the attachment
- SAM Metric calculation < removed invalid link >
- SAM publishing method < removed invalid link >
- SAM submission framework < removed invalid link >
- ... and of course you can always go to Valentin, Piotr, Judit..... 😊
- Description of SAM tests SAM Tests Twiki page

-- Main.gdebrecz - 10 Jan 2007

---

This topic: LCG > SAMEsettings

Topic revision: r13 - 2011-06-21 - AndresAeschlimann



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback