



*Nordic Testbed for Wide Area Com-
puting And Data Handling*

28/9/2004

THE LOGGER SERVICE

*Functionality Description and Installation Manual**

A.Konstantinov

*Comments to: aleks@fys.uio.no

Contents

1	Introduction	4
2	Logger Service	4
3	Logger Client	4
4	Installation	5
4.1	Install server	5
4.2	Configure MySQL	5
4.3	Configure server and service	5
4.4	Run server	5

1 Introduction

The Logger Service is one of services implemented in HTTP(S,G) and SOAP Framework [1] and run in *httpsd* server. It provides simple frontend to underlying MySQL database to store and retrieve an information about jobs processed by ARC middleware [2].

Corresponding client is provided as part of ARC installed on cluster's frontend.

2 Logger Service

It supports 2 SOAP methods for adding and retrieving records respectively. It also provides simple web interface through HTTP GET and POST methods. Unfortunately this interface is not very usefull because no web browser I know about supports pure SSLv3 which is only SSL protocol globus_io [3] supports.

The Logger service is configured through configuration file of HTTPS/G server. It supports following service-specific commands.

- *acl_read* [*group* [*group* [...]]]
- *acl_write* [*group* [*group* [...]]]
- *acl_query* [*group* [*group* [...]]]

If client belongs to one of groups listed in *acl_read* command it is allowed to retrieve information about job records. Those clients are still restricted to an information about jobs which they are owners of. And those listed in *acl_write* can add new records. Usually those groups match against host credentials of a clusters' frontends. Groups in *acl_query* are allowed to retrieve information about *all* jobs.

Other supported commands are:

- *sqlcontact* [*user* [*password*]]
- *sqlcontactsource* *path*

They provide username and password used to access MySQL database. *sqlcontactsource* points to file which contains username and password in first line. It is better to use *sqlcontactsource* and make it readable only for user who starts

3 Logger Client

Logger service can be communicated using utility *logger* installed into `${NORDUGRID_LOCATION}/libexec` by default. It allows to add new records into database and to retrieve information. First functionality is used by a grid-manager to pass information about processed jobs.

```
logger [-h] [-d level] [-u url] {control_dir ...|-q [-x] [-e element,element,...] [-o offset] [-s size] [
```

Here possible options are

- h show reminder,
- d set debug output to *level*,
- u URL of central logging server (mandatory if -q is given),
- q retrieve information, otherwise - report job to database,
- x print obtained information in XML,
- o from which record to start,
- s how many records to show,
- e specify which elements are desired in output.
Possible values are: *start,end,cluster,user,id,name,failure,lrms,queue,rsl,ui,usedcpu,usedmem*.

query is SQL conditional expression used to select jobs' records.

4 Installation

4.1 Install server

Just follow any of installation instructions available at <http://www.nordugrid.org/papers.html> to build (optionally) and install server part of NorduGrid ARC. If You choose 0.5 branch You will not need to do anything special. In case of 0.4 branch You will have to recompile ARC with `-enable-experimental` option of configure.

4.2 Configure MySQL

Logger service uses MySQL database. So it is essential to configure database properly. Currently You will either need file `grid-manager/https/logger/create.sql` or use content of Appendix A. It is important that You look into that file and modify first 3 lines (especially password). You can keep them as is, but at least make sure they fit Your intentions.

You have to feed that file to MySQL server. Preferably using `mysql` utility. This can look like `cat create.sql | mysql -p -u root .` ut You should check manual of `mysql` and configuration of the server before You type that.

If everything works MySQL server should have new database called *nglogger* and new user also called *nglogger* allowed to do everything inside that database. If You modified `create.sql` names can be different. Inside that database ther should be table called *jobs*. Do not change this name. There will also be table *files*, but it is not used yet.

That's it. You have database configured.

4.3 Configure server and service

Follow instruction in [1] to configure server. Do not forget to read [4] and add '=' if You use central configuration file.

Then follow instructions in section 2 to configure logger service. Make sure `sqlcontact/sqlcontactsources` contain right information.

4.4 Run server

Follow instruction in [1] to start server. Look into log file for line which looks like "Added service logger at /some/path". If You do not see it try to look through file for anything suspicious.

Appendix A

```
GRANT ALL ON nglogger.* TO nglogger@localhost IDENTIFIED BY 'pass';
CREATE DATABASE IF NOT EXISTS nglogger;
USE nglogger;
CREATE TABLE IF NOT EXISTS jobs (
  num INT UNSIGNED AUTO_INCREMENT,
  start DATETIME NULL DEFAULT NULL,
  end DATETIME NULL DEFAULT NULL,
  cluster VARCHAR(255) NOT NULL DEFAULT "",
  user VARCHAR(255) NOT NULL DEFAULT "",
  id VARCHAR(32) NOT NULL DEFAULT "",
  name VARCHAR(255) NULL DEFAULT NULL,
  failure VARCHAR(255) NULL DEFAULT NULL,
  lrms VARCHAR(32) NULL DEFAULT NULL,
  queue VARCHAR(32) NULL DEFAULT NULL,
  rsl TEXT NULL DEFAULT NULL,
  ui VARCHAR(255) NULL DEFAULT NULL,
  usedcpu INT NULL DEFAULT NULL,
  usedmem INT NULL DEFAULT NULL,
  KEY num(num)
);
CREATE TABLE IF NOT EXISTS files (
  job INT UNSIGNED NOT NULL,
```

```
url VARCHAR(255),  
type TINYINT,  
size BIGINT  
);
```

References

- [1] HTTP(S,G) and SOAP Server/Framework.
- [2] NorduGrid middleware, the Advanced Resource Connector. <http://www.nordugrid.org/middleware/>
- [3] The Globus Alliance. <http://www.globus.org>
- [4] Configuration and Authorisation of ARC (Nordugrid) Services.