EUROPEAN MIDDLEWARE INITIATIVE

LOGGING AND BOOKKEEPING – TEST PLAN & TEST SUITE DOCUMENTATION

Document version:	**1.4.15**
EMI Component Version:	**4.x**
Date:	**November 22, 2013**

This work is co-funded by the European Commission as part of the EMI project under Grant Agreement INFSO-RI-261611.

CONTENTS

L&B DOCUMENTATION AND VERSIONS OVERVIEW **5**

1 INTRODUCTION **7**
1.1 TEST CATEGORIES . . . 7
1.2 CERTIFICATION TESTS . . . 7
1.3 INTEGRATION INTO OTHER FRAMEWORKS . . . 8
1.3.1 SERVICE AVAILABILITY MONITORING . . . 8
1.3.2 NAGIOS . . . 8
1.3.3 ETICS . . . 8

2 SERVICE PING TESTS **9**
2.1 TEST SUITE OVERVIEW . . . 9
2.2 LOGGER (LOCAL & INTER) . . . 9
2.2.1 LOCAL TESTS . . . 9
2.2.2 REMOTE TESTS . . . 9
2.3 SERVER . . . 9
2.3.1 LOCAL TESTS . . . 9
2.3.2 REMOTE TESTS . . . 10

3 SYSTEM FUNCTIONALITY TESTS **11**
3.1 TEST SUITE OVERVIEW . . . 11
3.1.1 TEST SCRIPTS . . . 11
3.1.2 EVENT LOGGING EXAMPLES . . . 12
3.2 EVENT DELIVERY . . . 12
3.2.1 NORMAL EVENT DELIVERY . . . 12
3.2.2 JOB REGISTRATION ONLY . . . 12
3.2.3 STANDALONE LOCAL LOGGER – LOG EVENT . . . 12
3.2.4 INTERLOGGER RECOVERY . . . 13
3.3 JOB STATE COMPUTATION . . . 13
3.3.1 NORMAL JOB STATES . . . 13
3.3.2 SANDBOX TRANSFERS . . . 13
3.3.3 COLLECTION-SPECIFIC TESTS . . . 14
3.4 L&B SERVER AND PROXY COMBINED TEST . . . 14
3.5 WS INTERFACE . . . 14
3.6 NOTIFICATIONS . . . 15
3.6.1 SINGLE JOB, ANY STATE CHANGE . . . 15
3.6.2 INCLUDE ANOTHER JOB . . . 15

3.6.3 DELAYED DELIVERY . . . 15
3.6.4 DELIVERY TO ACTIVEMQ . . . 16
3.7 SITE NOTIFICATION MAINTENANCE . . . 16
3.8 SERVER PURGE . . . 17
3.9 OTHER SERVICE TESTS . . . 17
3.9.1 ACL PARSING . . . 17
3.9.2 STATISTICS . . . 18
3.9.3 MULTI-THREADED OPERATION . . . 18
3.9.4 CONFIG AND RUNTIME FILE PERMISSIONS . . . 18
3.9.5 NAGIOS PROBE . . . 19
3.9.6 HTTPS INTERFACE . . . 19
3.9.7 DUMP & LOAD EVENTS . . . 19
3.9.8 COLLECTION-SPECIFIC TESTS . . . 20

4 LB "IN THE WILD"—REAL-WORLD WMS TEST 20

5 REGRESSION TESTING 21
5.1 PUBLISHING CORRECT SERVICE VERSION OVER BDII . . . 21

6 PERFORMANCE AND STRESS TESTS 22

7 NAGIOS PROBE 23
7.1 NAGIOS PROBE FOR THE L&B SERVER . . . 23
7.1.1 TESTS PERFORMED . . . 23
7.1.2 RETURN VALUES . . . 23
7.1.3 CONSOLE OUTPUT . . . 24
7.1.4 RUNNING THE PROBE . . . 25
7.2 NAGIOS PROBE FOR THE L&B INTERLOGGER . . . 26
7.2.1 TESTS PERFORMED . . . 26
7.2.2 RETURN VALUES . . . 26
7.2.3 CONSOLE OUTPUT . . . 26

L&B DOCUMENTATION AND VERSIONS OVERVIEW

The Logging and Bookkeeping service (L&B for short) was initially developed in the EU DataGrid project[1] as a part of the Workload Management System (WMS). The development continued in the EGEE, EGEE-II and EGEE-III projects,[2] where L&B became an independent part of the gLite[3] middleware [1], and then in the EMI Project.[4]

The complete L&B Documentation consists of the following parts:

- **L&B User's Guide** [2]. The User's Guide explains how to use the Logging and Bookkeeping (L&B) service from the user's point of view. The service architecture is described thoroughly. Examples on using L&B's event logging commands to log user tags and change job ACLs are given, as well as L&B query and notification use cases.
- **L&B Administrator's Guide** [3]. The Administrator's Guide explains how to administer the Logging and Bookkeeping (L&B) service. Several deployment scenarios are described together with the installation, configuration, running and troubleshooting steps.
- **L&B Developer's Guide** [4]. The Developer's Guide explains how to use the Logging and Bookkeeping (L&B) service API. Logging (producer), querying (consumer) and notification API as well as the Web Services Interface is described in details together with programing examples.
- **L&B Test Plan** – this document. The Test Plan document explains how to test the Logging and Bookkeeping (L&B) service. Two major categories of tests are described: integration tests (include installation, configuration and basic service ping tests) and system tests (basic functionality tests, performance and stress tests, interoperability tests and security tests).

The following versions of L&B service are covered by these documents:

- *L&B version 4.0*: included in the EMI-3 *Monte Bianco* release
- *L&B version 3.2*: included in the EMI-2 *Matterhorn* release
- *L&B version 3.1*: an update for the EMI-1 *Kebnekaise* release
- *L&B version 3.0*: included in the EMI-1 *Kebnekaise* release
- *L&B version 2.1*: replacement for *L&B version 2.0* in gLite 3.2
- *L&B version 2.0*: included in gLite 3.2 release
- *L&B version 1.x*: included in gLite 3.1 release

L&B packages can be obtained from two distinguished sources:

- **gLite releases**: gLite node-type repositories, offering a specific repository for each node type such as *glite-LB*. Only binary RPM packages are available from that source.

[1] http://eu-datagrid.web.cern.ch/eu-datagrid/
[2] http://www.eu-egee.org/
[3] http://www.glite.org
[4] http://www.eu-emi.eu/

- **emi releases**: EMI repository[5] or EGI's UMD repository,[6] offering all EMI middleware packages from a single repository. There are RPM packages, both source and binary, the latter relying on EPEL for dependencies. There are also DEB packages (starting with EMI-2) and `tar.gz` archives.

Note: Despite offering the same functionality, binary packages obtained from different repositories differ and switching from one to the other for upgrades may not be altogether straightforward.

Updated information about L&B service (including the L&B service roadmap) is available at the L&B homepage: http://egee.cesnet.cz/en/JRA1/LB

[5] http://emisoft.web.cern.ch/emisoft/
[6] http://repository.egi.eu/

1 INTRODUCTION

This document explains how to test the Logging and Bookkeeping (L&B) service.

As part of the EGEE-III project, Specific Service Activity SA3: Integration, testing and certification[7], testing is an essential activity and all important information about gLite software testing should be available from the web page

https://twiki.cern.ch/twiki/bin/view/EGEE/EGEECertification#Test_writing.

This document describes test plan for the L&B service.

1.1 TEST CATEGORIES

According to the gLite Test Writing Guidelines [8], we consider two test categories with the following test types:

Integration tests verify if the software is installable and configurable. They also check for basic, not depending on other grid services, functionality of the component (e.g. daemon is up and running).

- Installation tests
- Configuration tests
- Service ping tests: basic tests if service is up and running, see Section 2

System tests verify if the component works within the grid in interaction with other grid services.

- Functionality tests of fully supported functionality (including APIs and CLI), see Section 3
- Performance and stress tests, see Section 6
- Interoperability tests
- Security tests

The tests could be run either locally (on the install node = where the service is installed, configured and running) or remotely (via another node, where some parts of the software also must be installed).

Apart from these tests, there exist also tests of the individual components run for example during the build process (especially unit tests). They are not described in this document.

1.2 CERTIFICATION TESTS

EGEE Certification team collects tests for L&B in a gLite module `org.glite.testsuites.ctb` in the `LB` directory. All L&B tests are described at https://twiki.cern.ch/twiki/bin/view/LCG/AvailableTests#Logging_and_Bookkeeping_LB as well as next to each test case in the following sections.

The tests can be used as sensors in different monitoring frameworks (see also below).

[7] https://twiki.cern.ch/twiki/bin/view/EGEE/SA3
[8] https://twiki.cern.ch/twiki/bin/view/LCG/LCGgliteTestWritingGuidelines

1.3 INTEGRATION INTO OTHER FRAMEWORKS

1.3.1 SERVICE AVAILABILITY MONITORING

Service Availability Monitoring (SAM)[9] is a framework for the monitoring of production and pre-production grid sites. It provides a set of probes which are submitted at regular intervals, and a database that stores test results. In effect, SAM provides monitoring of grid services from a user perspective.

1.3.2 NAGIOS

Nagios[10] is a host and service monitor designed to inform you of network problems before your clients, end-users or managers do.

There is a Nagios plugin that tests the status of the L&B server. It is discussed in detail in section 7, page 23.

1.3.3 ETICS

ETICS[11] stands for "eInfrastructure for Testing, Integration and Configuration of Software". It provides a service to help software developers, managers and users to better manage complexity and improve the quality of their software. Using cutting edge Grid software and best practices, ETICS allows to fully automate the way your software is built and tested.

Please see the ETICS User Manual [5] for the description of the ETICS service and basic ETICS commands. The command to be issued to test the whole L&B subsystem is:

```
etics-test org.glite.lb
```

It can be issued locally or using the remote build and test system.

General ideas of L&B tests using ETICS are the following

- tests are in CVS together with the code
- tests run the service themselves on some non-default ports and perform a set of elementary actions similar to those from `org.glite.testsuites.ctb/LB/tests/` to test the basic functionality of the service which is stopped again at the end of the test

[9] http://sam-docs.web.cern.ch/sam-docs
[10] http://www.nagios.org
[11] http://etics.web.cern.ch/etics/

2 SERVICE PING TESTS

In this section we describe basic tests of L&B if the services are up and running.

2.1 TEST SUITE OVERVIEW

This subsection gives a comprehensive overview of all service ping tests.

Executable	Status	Use
`lb-test-logger-local.sh`	Implemented	Test job logging facilities on a local machine (processes running, ports listening, etc.).
`lb-test-logger-remote.sh`	Implemented	Test the local logger availability remotely (open ports).
`lb-test-server-local.sh`	Implemented	Test for LB server running on a local machine (processes running, ports listening, etc.).
`lb-test-server-remote.sh`	Implemented	Test the LB server availability remotely (open ports).

2.2 LOGGER (LOCAL & INTER)

2.2.1 LOCAL TESTS

What to test: check if both `glite-lb-logd` and `glite-lb-interlogd` are running, check if `glite-lb-logd` is listening on which port (9002 by default), socket-connect to `glite-lb-logd`, check if enough disk capacity is free for `dglog*` files, socket-connect to `glite-lb-interlogd`.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-logger-local.sh`

2.2.2 REMOTE TESTS

Prerequisities: environment variable `GLITE_WMS_LOG_DESTINATION` set, GSI credentials set

What to test: network ping, check GSI credentials, socket-connect, gsi-connect

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-logger-remote.sh`

2.3 SERVER

2.3.1 LOCAL TESTS

Prerequisities: environment variables `GLITE_WMS_LBPROXY_STORE_SOCK`, and `GLITE_WMS_LBPROXY_SERVE_SOCK` set

What to test: check MySQL (running, accessible, enough disk capacity, ...), check if both daemons `glite-lb-bkserverd` and `glite-lb-notif-interlogd` are running, check if `glite-lb-bkserverd` is listening on which ports (9000, 9001 and 9003 by default), socket-connect to all `glite-lb-bkserverd` ports and sockets, check if enough disk capacity is free for dumps, socket-connect to `glite-lb-notif-interlogd`.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-server-local.sh`

2.3.2 REMOTE TESTS

Prerequisities: environment variable `GLITE_WMS_QUERY_SERVER` set, GSI credentials set

What to test: network ping, check GSI credentials, socket-connect to all server ports, gsi-connect to all server ports, WS getVersion.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-server-remote.sh`

3 SYSTEM FUNCTIONALITY TESTS

3.1 TEST SUITE OVERVIEW

This subsection gives a comprehensive overview of all system functionality tests.

3.1.1 TEST SCRIPTS

Besides pure System Functionality Tests, this list also includes In-the-Wild tests and Regression Tests discussed in a few following chapters. They are used in the same manner and, typically, on the same occasions, which is why they are all listed in the same place.

Executable	Use
`lb-test-job-registration.sh`	Tries to register a job and checks if the registration worked.
`lb-test-event-delivery.sh`	Tries to register a job and log events. Checks if the registration worked and events resulted in state change accordingly.
`lb-test-https.sh`	Test the HTTPs interface.
`lb-test-logevent.sh`	Test if local logger accepts events correctly (i.e. returns 0).
`lb-test-il-recovery.sh`	Tests if interlogger recovers correctly and processes events logged in between when restarted.
`lb-test-job-states.sh`	Test that job state queries return correctly and that testing jobs are in expected states.
`lb-test-collections.sh`	Perform various collection-specific tests.
`lb-test-proxy-delivery.sh`	Test correct event delivery through L&B proxy.
`lb-test-ws.sh`	Query events and job states through the Web-Service interface.
`lb-test-notif.sh`	Test if notifications are delivered correctly for testing jobs.
`lb-test-notif-switch.sh`	Test the correct behavior of a notification once its target jobid changes.
`lb-test-notif-recovery.sh`	Test if notification client receives notifications correctly upon restart.
`lb-test-purge.pl`	Test that L&B server purge works correctly.
`lb-test-wild.pl`	Test L&B “in the wild” (test with real-life WMS).
`lb-test-bdii.sh`	Test L&B server is published correctly over BDII.
`lb-test-sandbox-transfer.sh`	Test L&B’s support for logging sandbox transfers.
`lb-test-changeacl.sh`	Test proper parsing of `ChangeACL` events.
`lb-test-statistics.sh`	Test statistic functions provided by L&B
`lb-test-threaded.sh`	Rudimentary test for threaded clients L&B
`lb-test-notif-msg.sh`	Test delivery of L&B notifications over ActiveMQ
`lb-test-permissions.sh`	Check ownership and permission settings for config and runtime files
`lb-test-nagios-probe.sh`	Run the nagios probe and check results
`lb-test-notif-keeper.sh`	Test the `notif-keeper` script
`lb-test-dump-load.sh`	Test the dump (backup) and load (restore) procedure
`lb-test-collections.sh`	A placeholder for collection-specific regression tests

3.1.2 EVENT LOGGING EXAMPLES

There is an `examples` subdirectory in `GLITE_LOCATION`. It holds various example files—both binaries and scripts. There is—among others—a suite of scripts aimed at testing event delivery and the proper operation of the L&B state machine. Scripts named `glite-lb-<state>.sh`—where `<state>` corresponds with a job state—can be used to generate sequences of events that will always get an existing job into that state. (For example the `glite-lb-running.sh` script logs a series of 12 events resulting in the job state turning to running.) Some of these scripts are used by system functionality tests detailed bellow but all of them can also be used for manual testing.

3.2 EVENT DELIVERY

3.2.1 NORMAL EVENT DELIVERY

Prerequisities: all L&B daemons running (glite-lb-logd, glite-lb-interlogd, glite-lb-bkserverd)

What to test:

1. Register jobs with `edg_wll_RegisterJob`
2. Log reasonable sequences of events with `edg_wll_Log*`, both through logger and/or proxy
3. Check with `edg_wll_JobLog` that the events got delivered afterwards (approx. 10s).
4. Also check delivery and processing of events related to collections.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-event-delivery.sh`

Note: The test includes artificial delays. Takes approx. 60 s to finish.

Expected result: All sub tests (API calls) should return 0. The same events that were logged must be returned.

3.2.2 JOB REGISTRATION ONLY

Prerequisities: running glite-lb-bkserverd

What to test: call `edg_wll_RegisterJob`. Jobids should preferably point to a remote L&B server. Try re-registration using flag `EDG_WLL_LOGLFLAG_EXCL` in various scenarios.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-job-registration.sh`

Expected result: All sub tests (API calls) return 0.

3.2.3 STANDALONE LOCAL LOGGER – LOG EVENT

Prerequisities: running glite-lb-logd only, jobs registered in test 3.2.2.

What to test: call `edg_wll_Log*` for various event types in a sequence resembling real L&B usage, using the same jobids as in test 3.2.2

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-logevent.sh [event-file-prefix]`

Expected result: All sub tests (API calls) return 0, events are added one per line to the local logger files.

3.2.4 INTERLOGGER RECOVERY

Prerequisities: running glite-lb-bkserverd on the machine and port where jobids from 3.2.2 point to; files generated in 3.2.3; glite-lb-interlogd is stopped.

What to test: Make a copy of the files created in 3.2.3, then start glite-lb-interlogd. After approx. 10 s check the jobs with `edg_wll_JobLog` call.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-il-recovery.sh`

Note: The test includes artificial delays. Takes approx. 15 to 75 s to finish.

Expected result:`edg_wll_JobLog` should return the same events that were contained in the local logger files. The files should be removed by interlogger after approx. 1 min.

3.3 JOB STATE COMPUTATION

3.3.1 NORMAL JOB STATES

Prerequisities: glite-lb-bkserverd running, events from 3.2.1 logged.

What to test: Check state of the jobs with `edg_wll_JobStatus`. Check all possible job states (if necessary, log relevant events). Query both server and/or proxy.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-job-states.sh`

Note: The test includes artificial delays. Takes approx. 150 s to finish.

Expected result: The API call should return 0, the jobs should be in the expected states. Thorough tests may also cross check the values supplied in the events (e.g. destination computing element) wrt. the values reported in the job states.

3.3.2 SANDBOX TRANSFERS

Prerequisities: All L&B services running

What to test:

1. Register a compute job.
2. Register input sandbox transfer.
3. Register output sandbox transfer.
4. Generate events to trigger job state changes in one of the sandbox transfer jobs.
   (a) Start the transfer and check that state has changed appropriately.
   (b) Finish the transfer and check that state has changed appropriately.
5. Use another sandbox transfer job registered above to start, then fail the transfer and check that this is reflected by the resulting transfer job status.
6. Check that the compute job and its sandbox transfer jobs link up correctly.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-sandbox-transfer.sh`

Note: The test includes artificial delays. Takes approx. 50 s to finish.

Expected result: Job states should change on event delivery as expect, related jobs should “know” their IDs.

3.3.3 COLLECTION-SPECIFIC TESTS

Prerequisities: All L&B services running

What to test: There is no specific workflow for this test. It is a placeholder for various minor feature and regression tests related to job collections.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-collections.sh`

Expected result: All included checks should finish OK.

3.4 L&B SERVER AND PROXY COMBINED TEST

Prerequisities: running glite-lb-proxy, glite-lb-interlogd and glite-lb-bkserverd

What to test: Register jobs with `edg_wll_RegisterJobProxy`, log events using `edg_wll_LogEventProxy` and check the job states against both lbproxy (using `edg_wll_JobStatusProxy`) and bkserver (using `edg_wll_JobStatus`). Also test job collections and registration, including registration of subjobs. Pay special attention to job reaching final job status and to the automatic purge from proxy.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-proxy-delivery.sh`

`glite-lb-running.sh -x -m LB_HOST:PORT`
- logs sequence of events and returns JOBID

```
Q1:  glite-lb-job_status -x JOBID
Q2:  glite-lb-job_status JOBID
```
- Q1 queries LB proxy, Q2 queries LB server - both should return status of the job

`glite-lb-cleared.sh -x -m JOBID`
- logs sequence of events to JOBID pushing it to terminal state

```
Q1:  glite-lb-job_status -x JOBID
Q2:  glite-lb-job_status JOBID
```
- Q1 returns *error: edg_wll_JobStatusProxy: No such file or directory (no matching jobs found)* while Q2 returns state of the job until it is purged

Expected result: A new job state should be available immediately at the lbproxy and probably with a small delay also at the bkserver. Jobs that reach the final job state are really purged from the proxy.

Note: The test includes artificial delays. Takes approx. 50 s to finish.

3.5 WS INTERFACE

Prerequisities: glite-lb-bkserverd running, events from 3.2.1 logged

What to test: retrieve both events and job states with the L&B WS interface (operations `JobStatus`, `QueryEvents`).

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-ws.sh`

Note: The test includes artificial delays. Takes approx. 10 s to finish.

Expected result: the returned data should match those returned by the legacy API calls.

3.6 NOTIFICATIONS

All notification tests require the L&B server to be run with notifications enabled, i.e. to be run for example with options `-notif-il-sock /tmp/sock.test_notif -notif-il-fprefix /tmp/test_notif`, where `/tmp/sock.test_notif` is a socket of notification interlogger.

Please see also [2], Section 2.4, for other possible scenarios how to test the notification delivery.

3.6.1 SINGLE JOB, ANY STATE CHANGE

Prerequisities: All L&B services running

What to test:

1. Register a job.
2. Start a notification client, register with `edg_wll_NotifNew` for any state changes of the job, and invoke `edg_wll_NotifReceive`.
3. Send events triggering job state changes.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-notif.sh`

Note: The test includes artificial delays. Takes approx. 12 s to finish.

Expected result: All the events should trigger notifications reported by the running notification client.

3.6.2 INCLUDE ANOTHER JOB

Prerequisities: All L&B services running, notification from 3.6.1 still active

How to run:

1. Register another job.
2. Augment the notification registration with the new jobid using `edg_wll_NotifChange`.
3. Start notification client, bind to the registration with `edg_wll_NotifBind`.
4. Send events for the new job.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-notif-switch.sh`

Note: The test includes artificial delays. Takes approx. 25 s to finish. Also note that this test will not work with L&B versions older than 2.0.

Expected result: Notifications should be received by the client.

3.6.3 DELAYED DELIVERY

Prerequisities: All L&B services running

What to test:

1. Register another job.

2. Register a notification as in 3.6.1 but terminate the client immediately.
3. Log events for the job.
4. Restart the client, binding to the notification and call `edg_wll_NotifReceive` repeatedly.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-notif-recovery.sh`

Note: The test includes artificial delays. Takes approx. 25 s to finish.

Expected result: Delayed notifications should be received by the client almost immediately.

3.6.4 DELIVERY TO ACTIVEMQ

Prerequisities: All L&B services running, ActiveMQ broker configured and running

What to test:

1. Register a job.
2. Start a notification client,
3. Retrieve broker address from the server (since *L&B version 3.2*) register with `edg_wll_NotifNew` for any state changes of the job, and listen for messages.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-notif-msg.sh`

Note: The test includes artificial delays. Takes approx. 25 s to finish.

Expected result: All the events should trigger notifications that will be reported through ActiveMQ messages.

3.7 SITE NOTIFICATION MAINTENANCE

Prerequisities: All L&B services running.

What to test:

1. Create or modify a `site-notif.conf` file
2. Run the “keeper” script and check if the effect was as desired
   - Setting up a new notification
   - Changing its conditions
   - Registering for anonymized notifications
   - ...
3. Repeat from step 1 with different settings until all scenarios have been checked

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-notif-keeper.sh`

Note: The test includes artificial delays. Takes approx. 30 s to finish.

Expected result: Notifications must be registered and messages must arrive matching the currently applicable contents of the config file.

3.8 SERVER PURGE

WARNING: This test is destructive, it destroys ALL data in an existing L&B database.

The test is fairly complex but it does not make too much sense to split it artificially.

Prerequisities: All L&B services running, preferably a dedicated server for this test.

What to test:

1. Purge all data on the server with glite-lb-purge
2. Log two sets of jobs, separated with delay of at least 60s so that the sets can be distinguished from each other.
3. Using `edg_wll_JobLog` retrieve events of all the jobs
4. Purge the first set of jobs (by specifying appropriate timestamp), letting the server dump the purged events.
5. Purge the other set of jobs, also dumping the events.
6. Run purge once more.
7. Check if purged jobs turned into zombies.
8. In addition, check if a *cron* task exists to run the *purge* operation regularly and that it logs its output correctly.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-purge.pl`

Note: The test includes artificial delays. Takes approx. 3.5 minutes to finish.

Note: This test is destructive to your data. You need to run it with the `--i-want-to-purge` option to confirm your intention. Also, you need to provide the L&B server `address:port` explicitly as an argument to rule out any confusion.

Expected result: Data dumped in steps 3 and 4 should be the same as retrieved in 3. The final purge invocation should do nothing (i.e. nothing was left in the database).

3.9 OTHER SERVICE TESTS

3.9.1 ACL PARSING

Prerequisities: All L&B services running

What to test:

1. Register a job.
2. Send a `ChangeACL` event to add authorized user.
3. Check job status to see if the event was processed correctly.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-changeacl.sh`

Note: The test includes artificial delays. Takes approx. 15 s to finish.

Note: This test does not check if the setting actually takes effect. Implementing that functionality – relying on the use of at least two different identities – in an automated test is rather challenging.

Expected result: The *test DN* should be listed in the job's ACL.

3.9.2 STATISTICS

Prerequisities: All L&B services running

What to test:

1. Register a series of jobs.
2. Generate events to push jobs to a given state.
3. Run statistics to calculate rate of jobs reaching that state.
4. Query for average time needed by test jobs to go from one state to another.
5. Check if the statistics returned reasonable results.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-statistics.sh`

Note: The test includes artificial delays. Takes approx. 30 s to finish.

Expected result: Meaningful values should be returned by both tests.

3.9.3 MULTI-THREADED OPERATION

Prerequisities: L&B server running

What to test:

1. Register a series of jobs.
2. Run a client using multiple threads to query the server simultaneously.
3. Check if all threads finished OK.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-threaded.sh`

Note: This is not a thorough test. It is not capable of discovering rare or improbable problems but it will check the essentials of multi-threaded opration.

Expected result: The test must not hang. Meaningful results (albeit errors) must be returned by all threads.

3.9.4 CONFIG AND RUNTIME FILE PERMISSIONS

Prerequisities: All L&B services configured and running.

What to test:

1. Decide on permission/ownership masks for various files identified for checking.
2. Check ownership and permission settings for selected files.
3. Compare that with a pre-determined mask.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-permissions.sh`

Expected result: The status of all files should match the pre-determined mask.

3.9.5 NAGIOS PROBE

Prerequisities: All L&B services configured and running, nagios probe[12] installed.

What to test:

1. Check probe for presence
2. Run the probe
3. Check if text and exit code are OK or at least consistent

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-nagios-probe.sh`

Expected result: The probe has returned OK and exit code was 0.

Note: The probe includes artificial delays. The test takes approx. 10 s to finish.

3.9.6 HTTPS INTERFACE

Prerequisities: All L&B services configured and running.

What to test:

1. Register a test job
2. Register a test notification
3. Query user jobs over the HTTPs interface
4. Query test job status over the HTTPs interface
5. Query notification status over the HTTPs interface
6. Download server configuration page and check if essential values are present
7. Download statistics page and check if essential values are present and > 0

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-https.sh`

Expected result: All information was succesfully downloaded and matched expectations.

3.9.7 DUMP & LOAD EVENTS

Prerequisities: All L&B services configured and running.

What to test:

1. Register tests jobs of all applicable types, including collections and DAGs
2. Generate events to change the state of all types of test jobs. At least one subjob in any type of collection must remain free of events to test embryonic registration
3. Check all those jobs are in their expected states.

[12] see also page 23

4. Dump events for all test jobs
5. Purge all test jobs from the L&B server
6. Test that all test jobs were actually purged (status query must return `EIDRM`)
7. Load dumped events
8. Check that all test jobs are in their expected states as before

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-dump-load.sh`

Expected result: All jobs were first purged and then restored to their previous states.

Note: The test registers multiple jobs for each type to reduce the probability of an accidental positive result (could be caused by random event ordering).

Note: The probe includes artificial delays (about 12 s worth) but also performs a lot of action. The test takes approx. 150 s to finish.

3.9.8 COLLECTION-SPECIFIC TESTS

Prerequisities: All L&B services configured and running.

What to test:

1. Register a collection
2. Try querying for jobs by parent ID only

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-collections.sh`

Expected result: Children were returned.

4 LB "IN THE WILD"—REAL-WORLD WMS TEST

Prerequisities: All L&B services running, working grid infrastructure accessible (including permissions).

What to test:

1. Submit a simple *hello-world*-type job.
2. Submit a simple job and cancel it.
3. Submit a collection of simple jobs.
4. Submit a collection and cancel it.
5. Submit a simple job that is sure to fail.
6. Submit a collection of jobs, one of which is sure to fail.

In all above cases: Watch the life cycle. Check the resulting state (*Cleared*, *Cancelled* or *Aborted*). Check events received in the course of job execution; events from all relevant components must be present (NS, WM, JC, LM, and LRMS).

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-wild.sh`

Expected result: Jobs were submitted. Cancel operation worked where applicable. Resulting state was as expected (*Cleared*, *Cancelled* or *Aborted*). Events were received from all components as expected.

Note: The test runs automatically. Bear in mind that in a real-life grid, even valid jobs may not always finish successfully. Analyze failures and re-run if necessary.

Note: The number of jobs to generate may be specified using the `-n` argument

Note: Job submissions are limited to VOCE CEs in normal circumstances. Use the `-w` argument to override.

Note: Due to the nature of grid computing, this test may take hours to complete!

5 REGRESSION TESTING

5.1 PUBLISHING CORRECT SERVICE VERSION OVER BDII

Prerequisities: All L&B and BDII services running.

What to test:

1. Regression test for bug 55482 (`http://savannah.cern.ch/bugs/?55482`)
2. Query for information on the server.
3. Check version returned by the query.

How to run: `org.glite.testsuites.ctb/LB/tests/lb-test-bdii.sh`

Expected result: Query returns proper service status. Proper LB Server version is returned.

Note: The test will also ask the L&B server for a version number through its WS interface, and compare the two values. This, however, will not be done if the machine you use to run the tests does not have the proper binaries installed (`glite-lb-ws_getversion`), or if you do not make a proxy certificate available. Should that be the case, the test only checks if a version is returned and prints it out, without comparing.

6 PERFORMANCE AND STRESS TESTS

In this section we describe only the general idea of performance and stress tests of L&B components. This work is in progress and all necessary information is updated at the wiki page:

http://egee.cesnet.cz/mediawiki/index.php/LB_and_JP_Performance_Testing

The general idea is thus the following:

- All source modifications for tests are in CVS, conditionally compiled only with appropriate symbol.
- To compile LB with performance testing enabled, set environment variable `LB_PERF` to 1 prior to LB build:

```
export LB_PERF=1
etics-build org.glite.lb
```

- Component tests are run by shell scripts located under component directories, these tests may require binaries from other components, though.
- All tests use sequence of events for typical jobs (small job, big job, small DAG, big DAG) prepared beforehand. These events are stored in files in ULM format in CVS (see `org.glite.lb.common/examples`).
- Events are generated by `glite-lb-stresslog` program, which reads ULM text of events for particular test job and logs the event sequence directly by calling `*_DoLogEvent<variant>`. The number of test jobs is configurable. Stresslog inserts into every event timestamp when the event was generated and sent.
- Events are consumed by breaking normal event processing either in the component being tested or the next component in chain, that is instrumented to read and discard events immediately. The consumption itself is done by calling special function which takes current time, extracts timestamp from event and prints the difference (ie.. the event processing time)[13]. These "break points" are chosen to measure throughput of the various component parts and to identify possible bottlenecks within the components.
- Test jobs are preregistered within the LB if the test includes bookkeeping server and/or proxy by the test script program and their id's are stored in separate file to enable re-use by other load-generating tools (status queries, for example).
- Test results:
  - Some numbers must be reported by component themselves, not by the event generator (due to the asynchronous LB nature). The test script collects those numbers and presents them as the test result at the end of testing.
  - After completion test scripts print the table described for the respective tests filled in with measured values (ie.. the table is not filled in manually by human tester).
  - Measure event throughput by

$$\text{event\_throughput} = \frac{1}{\text{time\_delivered} - \text{time\_arrived}}$$

  - Publish the results on the web.

[13] the only exception is test of the logging library itself

7 NAGIOS PROBE

Two Nagios probes exist. One to check the running L&B server, and another one to make qualitative checks on a standalone L&B Interlogger.

7.1 NAGIOS PROBE FOR THE L&B SERVER

There is a Nagios probe to check the service status of an L&B server node. It is distributed from the EMI repository and the name of the package is `emi-lb-nagios-plugins`.

7.1.1 TESTS PERFORMED

Before starting the actual test the probe checks for existence and validity of a proxy certificate, and for availability of commands (essential system commands, various L&B Client commands and grid proxy manipulation commands).

The following tests are performed by the probe. Various tests check the working status of various processes running on the L&B server node:

1. Register job
   - L&B server (`glite-lb-bkserverd`)
2. Register to receive notifications
   - L&B server (`glite-lb-bkserverd`)
3. Log events resulting in state *cleared*
   - L&B logger (`glite-lb-logd`)
4. Check job state
   - L&B server (`glite-lb-bkserverd`)
   - Interlogger (`glite-lb-interlogd`)
5. Receive notifications
   - Notification interlogger (`glite-lb-notif-interlogd`)

The test also tries to drop the test notification and purge the test job to clean up after itself. However, purging the job won't probably be allowed by the L&B server's policy and the test job will remain registered on the L&B server until removed by a regular purge.

7.1.2 RETURN VALUES

Return values follow the Nagios pattern:

0 The service is running normally

1 The service is running but there were warnings

2 The service status is critical

3 The service status is unknown, probe could not run

7.1.3 CONSOLE OUTPUT

Text output indicates the results of the probe and gives a more detailed description of failure causes.

The probe can return one of the following:

WARNING	*Unexpected version output*	The server responded to a query for server version over the WS interface, but the format of the response did not match the expected pattern.
	Unexpected state of test job	The state of the test job did not remain unchanged (`Submitted`) but neither did it reach status `Cleared` in the alotted time. All deamons seem to work but the processing is slow.
	Could not drop notification	The owner should be able to drop their own notification. Failure to do so is unexpected but does not mean that the service is not functioning.
DOWN	*Unable to Get Server Version*	The server did not respond to a query for server version over the WS interface. It is probably not running, is unaccessible or SSL handshake failed due to faulty/outdated certificates/CRLs.
	Job Registration Failed Locally	The probe was unable to perform the local side of job registration. This should be rare.
	L&B Server Not Running	The probe was unable to register a test job or a test notification with the L&B server. It is probably not running or is unaccessible.
	Event Delivery Chain (Logger/Interlogger) Not Running	The server process is running but events are not being delivered by L&B's local logger/interlogger. Check the Logger and the Interlogger.
	Notification Interlogger Not Running	Events are being delivered correctly and server responds properly to status queries, but it its not delivering notification messages. The notification interlogger is probably not running.

UNKNOWN	*Probe timed out*	The probe was unable to finish before the alotted time. Consider increasing the timeout with `-t`. The minimum reasonable value is 10 s.
	No server specified	Server address was not specified when running the probe. Give one with `-H`.
	Probe could not write temporary files	The temporary directory was not writable. Check the default location or specify a new one with `-T`.
	Some commands are not available	Probe could not run. Some of the required commands are not present on the system. Run probe from command line with `-v[vv]` and check output.
	No Credentials	No proxy certificate was found. Probe could not run.
	Credentials Expired	A proxy certificate was found, but expired. Probe could not run.

7.1.4 RUNNING THE PROBE

Command Line Arguments The probe recognizes the following command line arguments:

`-h`	`--help`	Print out simple console help
`-v[v[v]]`	`--verbose`	Set verbosity level (`--verbose` denotes a single `v`).
`-H`	`--hostname`	L&B node address. `GLITE_WMS_QUERY_SERVER` environmental variable is used if unspecified.
`-p`	`--port`	L&B server port. Other port numbers (logger, WS interface) are derrived from it. `GLITE_WMS_QUERY_SERVER` environmental variable or default port `9000` are used if unspecified.
`-t`	`--timeout`	Timeout in seconds. The minimum reasonable timeout is approx. 10 s. There is no default, except the internal waiting cycle for notifications, which will time out after approx. 20 s.[14]
`-T`	`--tmpdir`	Directory to store temporary files. By default the probe uses `/var/lib/grid-monitoring/emi.lb` and falls back to `/tmp` if the former does not exist or is not writable.
`-a`	`--notif-endpoint`	Endpoint for listening to notifications. This should only be used if there are strict firewall settings on the client machine and only a specific port is kept open for the probe. By default, a random high port number is used.
`-x`	`--proxy`	User proxy file. It only needs to be specified if the proxy cannot be found at the default location, or pointed to by environmental variables.

Environmental Variables In essence the probe recognizes the same environmental variables as the L&B client. No environmental variables need to be set if hostname is specified as a command line argument to the probe.

[14] The probe adjusts the internal waiting cycle to spend a maximum of $\frac{3}{4}$ of the specified timeout interval while waiting for notifications to deliver. It will finish correctly before timing out if undelivered notifications are the only problem.

GLITE_WMS_QUERY_SERVER	**The** L&B server. This is the server that will be contacted and tested if no hostname is supplied to the probe.
GLITE_LB_SERVER_PORT GLITE_LB_LOGGER_PORT	Specifies the L&B server port or the L&B local logger port, respectively. It is used only in case a hostname is given as a command line argument to the probe wit no port number.
X509_USER_PROXY	Alternative location of the user's proxy certificate to use in the test.

Sample Nagios Service Definition Simple definition to be included in `/etc/nagios/commands.cfg`:

```
define command{
        command_name    check-lb-server
        command_line    /usr/libexec/grid-monitoring/probes/emi.lb/LB-probe \$HOSTADDRESS$
        }
```

7.2 NAGIOS PROBE FOR THE L&B INTERLOGGER

This probe is intended for checking runtime parameters of an L&B interlogger. It is distributed from the EMI repository and the name of the package is `emi-lb-nagios-plugins`. It is intended for use on the target machine. It cannot measure remotely.

7.2.1 TESTS PERFORMED

The following tests are performed:

1. The total size of interlogger files (waiting for processing) is measured and compared to pre-set thresholds.
2. The age of the interlogger socket is read

7.2.2 RETURN VALUES

Return values follow the Nagios pattern:

0 The service is running normally

1 The service is running but there were warnings

2 The service status is critical

3 The service status is unknown, probe could not run

7.2.3 CONSOLE OUTPUT

Text output indicates the results of the probe and gives a more detailed description of failure causes.

The probe can return one of the following:

WARNING	*Total of InterLogger files exceeds bounds*	The total size of IL files is higher than the warning threshold. It is possible that the files are not being processed.
CRITICAL	*Total of InterLogger files exceeds bounds* *Interlogger socket not being refreshed anymore*	The total size of IL files is higher than the critical threshold. It is possible that the files are not being processed. The socket has not been refreshed for over 60 s. Under normal circumstances, a running Interlogger refreshes the socket every second.
UNKNOWN	*Some commands are not available*	Probe could not run. Some of the required commands are not present on the system. Run probe from command line with `-v[vv]` and check output.

Command Line Arguments The probe recognizes the following command line arguments:

`-h`	`--help`	Print out simple console help
`-v[v[v]]`	`--verbose`	Set verbosity level (`--verbose` denotes a single `v`).
`-t`	`--timeout`	Timeout in seconds.
`-f`	`-file-prefix`	Path and prefix for event files
`-s`	`-sock`	Path and prefix for IL socket
`-S`	`-sock-timeout`	Timeout for the IL socket (default 60 s)
`-w`	`-warning`	Log file size limit (kB) to trigger warning (default 10 MB)
`-c`	`-critical`	Log file size limit (kB) to trigger state critical (default 128 MB)
`-P`	`-proxy`	Use default prefix for Proxy Interlogger files rather than regular IL
`-N`	`-notif`	Use default prefix for Notification Interlogger files rather than regular IL

REFERENCES

[1] E. Laure, F. Hemmer, F. Prelz, S. Beco, S. Fisher, M. Livny, L. Guy, M. Barroso, P. Buncic, P. Kunszt, A. Di Meglio, A. Aimar, A. Edlund, D. Groep, F. Pacini, M. Sgaravatto, and O. Mulmo. Middleware for the next generation grid infrastructure. In *Computing in High Energy Physics and Nuclear Physics (CHEP 2004)*, 2004.

[2] A. Křenek et al. L&B User's Guide. http://egee.cesnet.cz/en/JRA1/LB/.

[3] A. Křenek et al. L&B Administrator's Guide. http://egee.cesnet.cz/en/JRA1/LB/.

[4] A. Křenek et al. L&B Developer's Guide. http://egee.cesnet.cz/en/JRA1/LB/.

[5] Etics user manual. https://edms.cern.ch/file/795312//ETICS-User_Manual-latest.pdf.