

Table of Contents

WMS Service test suite.....	1
Introduction.....	1
Testsuite description.....	2
WMS command tests.....	2
WMS-command_job-delegate-proxy.sh.....	2
WMS-command_job-list-match.sh.....	2
WMS-command_job-info.sh.....	2
WMS-command_job-submit.sh.....	3
WMS-command_job-status.sh.....	3
WMS-command_job-logging-info.sh.....	4
WMS-command_job-output.sh.....	4
WMS-command_job-cancel.sh.....	5
WMS-command_job-perusal.sh.....	5
Functionality tests.....	5
WMS-job-cycle.sh.....	5
WMS-job-resubmit.sh.....	5
WMS-job-parametric.sh.....	6
WMS-prologue-epilogue.sh.....	6
Regression tests.....	6
Releases and Changelog.....	6

WMS Service test suite

Introduction

The WMS-service testsuite is a set of shell scripts that interacts with the WMS command line tools in order to perform several functional tests.

Location: org.glite.testsuites.ctb/WMS/WMS-service

Developers: Alessio Gianelle

The testsuite should be divided into two parts: the first one is composed by some scripts one for each commands available in the WMS-CLI (`glite-wms-job-*`), the purpose of these tests is to check every single command and to check if the service reacts as expected; these tests can also be used to check the WMS command line commands. Then there are some general scripts which check singles features of the WMS service.

There are two ways to configure the tests: you can use a configuration file (`wms-command.conf`) like this one:

```
### Required parameters
# WMS used for tests
WMS="wms.pd.infn.it"
# LB used in the test
LB="lb.pd.infn.it"
# User VO
VO="dteam"

### Optional parameters
# Default requirements
DEFAULTREQ="other.GlueCEStateStatus == \"Production\""
# Number of retrievals before to stop test (timeout)
NUM_STATUS_RETRIEVALS=100
# Seconds between two retrievals
SLEEP_TIME=30
```

Or you can set the required parameters through command line options:

```
-h          this help
-l          save output in a file
-d <level > print verbose messages (level = (1|2|3)
-i          interactive mode (it asks for proxy pwd)
-n          nagios mode
-c <conf>   configuration file
-W <wms>   WMS host (required if conf file is not used)
-L <lb>    LB host (required if conf file is not used)
-C <ce>    CE host
-v <vo>    User VO (required if conf file is not used)
```

As you can see there are some **required** parameters: `WMS` which is the WMS hostname to test, `LB` which is the LB hostname and `VO` which is the name of the user's VO.

To use the WMS service a valid proxy is needed. There are three ways to set it for the testsuite: you can create a valid proxy before execute the testsuite (i.e. the default location pointed by `X509_USER_PROXY` is used) or you can left the testsuite to create it for you digitig the passphrase in an interactive mode (i.e. use option `-i`) or setting a default key without password. Note that some tests require the possibility to create limited proxies so they work only in the second case (i.e. if you give to the testsuite the opportunity to create the proxy).

Testsuite description

WMS command tests

WMS-command_job-delegate-proxy.sh

This test is intended to verify glite-wms-job-delegate-proxy command

Test the command with these options:

- --version
- --logfile
- --output
- --endpoint
- --autm-delegation

success means that command has success, delegation is stored in the WMS and that we are able to submit a job

The following tests are possible only in *interactive* mode (i.e. use option `-i`)

- Delegate a shorter proxy
- Try to delegate with an expired proxy
- Try to submit with an expired delegation

WMS-command_job-list-match.sh

This test is intended to verify glite-wms-job-list-match command

Test the command with these options:

- --logfile
- --output
- --endpoint
- --rank

success if command returns 0

- Test a failure matching (requirements = false). Success if no matching is found.
- Test a restricted matching (only Cream CEs). Success if none LCG CE matches.
- Test delegation proxy option `--delegationid` (delegate before the mm). Success if command returns 0.

WMS-command_job-info.sh

This test is intended to verify glite-wms-job-info command

Test these options for all the command: `--jdl-original --jdl --proxy --delegationid`

- --version
- --input
- --config
- --output
- --logfile

- --endpoint
- all options together

success if command returns 0

- Check if the JobID is set correctly in the retrieved jdl
- Check if the expiration time of the delegation is set correctly"

WMS-command_job-submit.sh

This test is intended to verify glite-wms-job-submit command

Test the command with these options:

- --version option
- --endpoint option
- --output option
- --logfile option
- --nomsg option
- all options together

success if command returns 0

- --input option (with --noint)
- --resource option

success if job is submit to the required CE

- --register-only option

success if job is in state *Submitted*

- --start option

success if job correctly terminated

- --transfer-files --proto option

success if job output is as expected (i.e. files are correctly transferred)

- --valid option
- --to option

success if *ExpiryTime* is correctly set in the jdl and if job aborted with reason *request expired*

- --default-jdl option

success if the attribute id the *default-jdl* is add to the job's jdl

WMS-command_job-status.sh

This test is intended to verify glite-wms-job-status command

Test the command with these options:

WMS-command_job-info.sh

- --version
- --config
- --output
- --logfile
- --input

success if command returns 0

Testing these options requires certain indexing capabilities to be enabled on the LB server, so they can failed.

- --exclude (exclude Waiting, Done, Cleared and Aborted jobs)
- --status (look for Waiting job)
- --user-tag

success if command output is as expected.

WMS-command_job-logging-info.sh

This test is intended to verify glite-wms-job-logging-info command

Test the command with these options:

- --version option
- --config
- --output
- --logfile
- --input
- --to
- --from
- --event (show only ACCEPTED events)
- --exclude (exclude ACCEPTED events)
- all the options together (extract only EnQueued events)

success if command returns 0 and output is as expected.

WMS-command_job-output.sh

This test is intended to verify glite-wms-job-output command

Test the command with these options:

- --version
- --logfile
- --dir
- --nosubdir
- --list-only
- --nopurge
- --input

success means that the status is "CLEARED" (otherwise a WARNING is produced) and that output files std.out and std.err are correctly retrieved.

- Try to retrieve output of a Cleared job. Success if command failed.

WMS-command_job-cancel.sh

This test is intended to verify glite-wms-job-cancel command

Test the command with these options:

- --version
- --config
- --output
- --logfile
- --input
- all options together

success means that the final state of the job is "Cancelled"

- cancel a finished job. Success if command fails.

WMS-command_job-perusal.sh

This test is intended to verify glite-wms-job-perusal command

Test the command with these options:

- --version
- --set
- --get
- --get with --dir
- --get with --nodisplay and --all

success if the retrieved pieces of file are exactly the job's output.

Functionality tests

WMS-job-cycle.sh

Test a complete job cycle: from submission to get-output

- submission of a *normal* job to an LCG-CE
- submission of a *normal* job to a CREAM CE

success if job terminated correctly and the retrieved output files are the two expected files (std.out and std.err)

You can try submission to a specific CE using command's option `-c`

WMS-job-resubmit.sh

Test the two types of possible resubmissions.

- shallow resubmission
- deep resubmission

Success if job fails with the expected reason

WMS-job-parametric.sh

Test parametric job.

- Test integer version of Parameters
- Test integer version of Parameters changing 'start' and 'step' values
- Test list version of Parameters

For these three tests job is only register and the produced jdl is checked.

- Test submission and check the output
- Test submission using a list as parameters

Success means that job finished with *Done Success*, that the required produced output files are correctly retrieved and that they are as expected.

Known issues:

- bug #79027 ParameterStart must be equal to 0
- bug #79144 NodesCollocation is not support
- bug #79165 Parameters list must be composed by strings
- bug #79308 parameters should not be negative (SegFault)

WMS-prologue-epilogue.sh

Test jdl's attributes Prologue (PrologueArguments) and Epilogue (EpilogueArguments). Success means that job has been correctly submitted, and that at the end output files are correctly retrieved. Then some checks are done on the output files written by the prologue, the executable and the epilogue scripts.

Regression tests

Location: <http://glite.cvs.cern.ch/cgi-bin/glite.cgi/org.glite.testsuites.ctb/WMS/regression/>

Releases and Changelog

- 2011-03-16 (Ale) Change run_command function. Now all tests are perform reporting how many of them eventually fail.
- 2011-03-11 (Ale) Added WMS-job-parametric.sh test

-- AlessioGianelle - 08 Jul 2008

This topic: EgeeJra1It > WMSWorkPlan

Topic revision: r10 - 2011-04-27 - AlessioGianelle



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

Ideas, requests, problems regarding TWiki? Send feedback