

# Table of Contents

<b>WMS TEST SUITE.....</b>	<b>1</b>
Introduction.....	1
Requirements.....	1
Configuration.....	2

# WMS TEST SUITE

## Introduction

The WMS test suite is a set of python scripts that interacts with the WMS command line tools in order to perform automatically several functional tests.

Location:

- Svn repository url: svn://150.140.139.200/wms\_tests
- username: guest
- password: glitetests

Developers: Alessio Gianelle, Aristotelis Kretsis

The test suite is divided into two packages. The first package (WMS-cli) is composed by some python scripts, one for each commands available in the WMS-cli (glite-wms-job-\*). The purpose of these tests is to check every single command using various meaningful options and to check if the service reacts as expected. The second package (WMS-service) is composed by some general python scripts which check the most critical features of the WMS service.

Each python script contains a number of individual test cases. When some test case fails for any reason the test does not terminate but it continues to the next available test case. During the test execution a working directory is created which contains among the other three files very useful for debugging. The first one is the **WMSService-TS\_<test id>.log** (for WMS-service package) or **WMSCli-TS\_<test id>.log** (for WMS-cli package) which contains general debug information for the whole execution of the test. At this file is logged all the useful information for each test case, like test number and description, jobids, status of jobs, commands and their status and output. The log level is configured by the user (see configuration details) and there are three possible options WARNING, INFO and DEBUG. The second debug file (**errors.log**) contains information related to the errors which occur during the execution of test, like the test number and description, the failed operation and the failure reason. At third file (**traceback\_errors.log**) users can find the traceback information for each failure.

*Note that when a test case fails the working directory is not cleaned and the user can use the various files to determine the failure reason.*

## Requirements

There are only two requirements for someone to use the test suite:

- have a valid proxy because it is necessary in order to use the WMS service
- provide the required configuration parameters

There are three ways to set a valid proxy for the test suite:

1. create a valid proxy before execute the test suite (i.e. the default location pointed by X509\_USER\_PROXY is used)
2. left the test suite to create it for you by giving the passphrase in an interactive mode (i.e. use option -i)
3. set a default key without password

*Note that some tests (testing delegation and proxy renewal operations) require the possibility to create limited proxies so they work only with the second option (i.e. if you give to the test suite the opportunity to create the*

proxy). In these cases the user is informed of further tests with an appropriate message.

## Configuration

The configuration parameters are divided into two categories the required and the optional parameters. In first case belong the following three parameters:

- **WMS:** the WMS hostname to test
- **LB:** the LB hostname
- **VO:** the name of the user's VO

While the optional parameters are:

- **MYPROXYSERVER:** MyProxyServer hostname
- **LOG\_LEVEL:** the log level , WARNING,INFO and, DEBUG (default value DEBUG)
- **DEFAULTREQ:** the default requirements to use at jdl files ( default value `other.GlueCEStateStatus = "Production"= )`
- **NUM\_STATUS\_RETRIEVALS:** the number of retrievals before to stop test (timeout) (default value 120)
- **SLEEP\_TIME:** the seconds before the next status check (default value 30)

There are two ways to configure the tests:

1. use a configuration file (default is `wms-command.conf` )
2. use command line options

When we use the first method we can set values for both required and optional configuration parameters while the command line options are limited only to required parameters. We can also specify a different configuration file by using the `-c` command line option. *Note that for what concern configuration there is an hierarchy. First of all it is sourced the default configuration file if exists (i.e. `wms-command.conf`) then we read the command line options and at least we sourced the configuration file specified by the user with the `-c` option. This means that in every step options are eventually overwritten.*

Finally, the names of the configuration parameters should be exactly the same as listed above.

Example of configuration file:

```
### Required parameters

# WMS used for tests
WMS="WMS_HOSTNAME"

# LB used for test
LB="LB_HOSTNAME"

# User VO
VO="USER_VO"

### Optional parameters

# Log level
LOG_LEVEL=INFO

# Default requirements
DEFAULTREQ=other.GlueCEStateStatus == "Production"

# Number of retrievals before to stop test (timeout)
NUM_STATUS_RETRIEVALS=100
```

```
# Seconds before the next status check  
SLEEP_TIME=30
```

Tests provide the following command lines options:

```
-h                this help  
-l                save output in a file  
-v                print log messages to stdout  
-d <level >     define log level (1=WARNING, 2=INFO, 3=DEBUG)  
-i                interactive mode (it asks for proxy pwd)  
-c <conf>        configuration file overwrites command line options  
-W <wms>         WMS host (Required)  
-L <lb>          LB host (Required)  
-C <ce>          CE host  
-V <vo>          User VO (Required)  
-j <jdl file>    External jdl file  
-s                List all available tests  
-t <test numbers> Run specific test cases. Syntax number1,numberK,numberM or number1-numberN c
```

This topic: WMS > WmsTestSuite

Topic revision: r1 - 2011-07-13 - AlessioGianelle



Copyright © 2008-2023 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback