

# OpenSSO QA Test Automation

## Logging module Testing Setup & Development

Prepared by Kandula Srinivas

### **1. Introduction**

This document describes in detail the QA test automation focusing on Logging related tests. It explains the following:

- The Requirements for this module
- How tests are organized.
- Execution details
- Tests in the framework
- Interpreting the report
- Debugging the test failures
- How to add new tests

### **2. Requirements**

The primary requirements are:

- Setup OpenSSO server
- Setup MySQL Database.

### **3. Organization of Tests**

In the section, we shall discuss how tests are organized. Currently there are testcases for following features

- Modify logging service configuration.
- Create log records of type INFO,SEVERE,WARNING,CONFIG,FINE,FINEST,FINER and logging using remote logging api.
- Create log records of type INFO,SEVERE,WARNING,CONFIG,FINE,FINEST,FINER and logging using remote logging api to MySQL database.
- Taking backup of existing log configuration.
- Restore the log configuration after the execution.

Test organization details:

1. This is how the directory and file's are laid out for log module
  - <TEST\_HOME>/xml/testng contains log testng xml files.

- <TEST\_HOME>/resources/log contains all properties files for log module testing.
  - <TEST\_HOME>/source/com/sun/identity/qatest/log contains all sources of Java implementations for log tests
  - <TEST\_HOME>/source/com/sun/identity/qatest/common This directory contains the common classes and methods that are used by log test module.
  - <TEST\_HOME>/<server name>/built/classes will contain xml files which will be generated at run time for log module testing.
2. Logging tests are divided under different groups such as
- ff\_ds: Embedded for User management & Directory Server as the service data repository.
  - ds\_ds: Directory Server for User management and the service data repository
  - ff\_ds\_sec : Embedded for User management & Directory Server as the service data repository with security enabled
  - ds\_ds\_sec : Directory Server for User management and the service data repository with security enabled

## **4. Execution details**

### **4.1 Steps for executing Log tests**

1. Deploy Federated Access Manager 8.0 war on a host
2. Install MySQL on a host.
3. Before executing this test module, user should create a <TEST\_HOME>/resources/Configurator-<servername>.properties corresponding to server war deployment. Please refer to OpenSSO QATest automation framework document for details.
4. Change following parameters in <TEST\_HOME>/build.properties file
  - Change the value of QATEST\_HOME
  - Change the value of TEST\_MODULE to “log”.
  - Change the value of EXECUTION\_MODE to appropriate group name as described in section 3
  - Change the value of REPORT\_DIR to desired location
5. Modify DB Host name, DB port no and DB password in <TEST\_HOME>/resources/log/DBConfigInfo.properties
6. Download mysql-connector-java-3.1.14-bin.jar from MySQL site and copy

it into qatest/lib directoy.

7. Copy the mysql-connector-java-3.1.14-bin.jar to FAM server machine and add it to container classpath.
8. Run following command to execute log module:  
ant -DSERVER\_NAME1=<server host name as mentioned in configuration propertiesfile> module

#### 4.2. The execution details

When Log test is executed, it will :

1. It creates a identity name “testuser” and gets the SSO Token for the user which is used for creating log records.
2. It will Modify the logging service configuration as specified by the key “attr\_value\_pair” in LogToDBTest.properties file
3. Creates a log record with the message as specified by “message” key.
4. Sets Remote logging service level.
5. Logs the message using remote logging API.
6. In case of DB logging it creates DB connection and creates database name “IDENTITY” in the DB with table “qatest\_access”
7. If the log service is configured for DB. For each testcase the last record is read and verified with the “expected\_message”.

### 5. Tests details

#### 5.1 Current Tests classes:

<i>TestClass</i>	<i>Properties file name</i>	<i>Description</i>
LogToDBTest.java	LogToDBTest.properties DBConfigInfo.properties	Creates the log records and logs to DB.
LogToFileTest.java	LogToFileTest.properties	Creates the log records and logs to File

#### 5.2 Properties files

- LogToDBTest.properties: This property file contains all the testcase information which is used for creating log records to be logged to DB.
- LogToFileTest.properties: This property file contains all the testcase information which is used for creating log records to be logged to File.
- DBConfigInfo.properties: This file contains contains configuration information for connecting to DB.

#### 5.3 Common Files

LogCommon.java: A class which contains the common methods for writing log using remote logging API, getting DB connection and reading last log record from DB.

LogConstants.java: This class contains all the Constants.

## **6. Interpreting the Log automated testing results**

- After execution of log test module, the results will be located under  
`<REPORT_DIR>/<SERVERNAME>/<EXECUTION_MODE>/<EXECUTION_DATE_TIME>`.
- Open index.html residing in this directory from the browser. It will display the overall Result of the test execution with the following details:
  - A link named "`<EXECUTION_MODE>-log`" which points to the detailed test report
  - The number of tests which passed.
  - The number of tests which failed.
  - The number of test skipped.
  - A link to the TestNG XML file used in this test run
- To learn more about the specific tests click on the link "`<EXECUTION_MODE>-log`". In the left frame of the resulting page, the individual results of all the tests which were executed. Passing tests will have a background color of green. Failed tests will have a background color of red.
- To find out more information on the results a particular test click on the "Results" link for that test. This will provide you more information about the test such as when the test was executed, the duration of the test in seconds, the test method being executed, and any exception that was thrown during execution of the test.
- To view all the log messages which were displayed for a particular test go to the file  
`<REPORT_DIR>/<SERVERNAME>/<EXECUTION_MODE>/<EXECUTION_DATE_TIME>/logs`. In this file, search for the name of the test of interest. Below the name the log records produced during the three phases of this test's execution, setup, verification, and cleanup, can be viewed.

## **7. Debugging the log automated test failures**

- Make sure the debug mode on client side set to default value, message. If not, edit `<TEST_HOME>/resources/AMClient.properties` and change the value of parameter `com.ipplanet.services.debug.level` to message. The client SDK debug logs are in debug directory of test result directory. For information about test result directory, please refer to section 5.
- To change the log level of QA Test from FINE to SERVER, CONFIG, FINER or FINEST so it can display more debug information about the test, edit `<TEST_HOME>/resources/Configurator-<hostname>.properties` and change the value of parameter `log_level`.
- To display verbose level of testNG test case execution, go to `<TEST_HOME>/xml/testng/<group name>-log-testng.xml` and change the value

of attribute verbose in tag <suite> to 5 .

For example, <suite name="ds\_ds-log" verbose="5">

- For DB logging tests, it will perform the verification step and display a successful or failed message in the logs file in test result directory.
- Common errors and solutions

Problems	Solutions
LDAP Bind Password for the datastore is not appeared on Admin Console's datastore configuration page that causes the test failed.	Edit <TEST_HOME>/resources/Configurator- <hostname>.properties and make sure the value of encryption_key matches with the value of am.encryption.passwd in FAM server's AMConfig.properties file
Not able to get DB connection	Make sure that DB server is running and up. Verify the values given in DBConfigInfo.properties. Make sure the jar file mysql-connector-java-3.1.14-bin.jar is in qatest/lib and in FAM server container classpath. In *unix make sure that this jar file have read permissions for the user.

## 8. How to add new testcases?

This section describes how to add new testcases to the existing or new java class.

### **To add a new test cases:**

- Log test case are specified in the properties files in directory <TEST\_HOME>/resources/log. New testcases should be added to LogToFile.properties or LogToDB.properties. Before adding new testcase, make sure you understand all the existing tests in that class. Especially in DB logging case, make sure that the "expected\_message" have the right value.  
The format and description of each parameters in these properties files available on the header of each properties file.  
<testcase\_name><testcase\_number>.<parameter\_key>=  
<value> :  
LogToDBTest0.logger\_level=info
- After adding a new test case in the properties file, go to <TEST\_HOME>/xml/testng and edit all <execution mode>-log-testng.xml file i.e. ds\_ds-log-testng.xml. Append this entry to each file:  
<test name=new testName>  
<parameter name="testName" value=<newTestName><newtestno>"/>

```

<parameter name="createUser" value=<true/false>/>
<parameter name="restore" value=<true/false>/>
<parameter name="modifyServConfig" value="<true/false>"/>
<groups>
  <run>
    <include name="ds_ds"/>
  </run>
</groups>
<classes>
  <class name="com.sun.identity.qatest.log.LogToDBTest"/>
</classes>
</test>

```

Note:-

- When createUser is set to “true” it creates a user as specified in LogTo\*.properties file with the password same as the user\_id
- If the restore is set to “true” this testcase will restore back the log service configuration.
- If modifyServConfig is set to “true” it will modify the log service configuration as specified in the property  
 <test\_name>.attr\_value\_pair=<log-service-attribute-value-pair> in LogToDBTest.properties file.

### **To add new test cases in new class:**

- Create a new class with appropriate class name. The properties files that is used by this test class should locate in <TEST\_HOME>/resources/log directory. Please follow naming conventions described in the openSSO QA Test Document for the class name and properties file name
- The new class should follow setup, test(s) and cleanup procedures. Cleanup should make sure it remove the identities and realm that was created during the setup or test execution
- Appropriate groups should be assigned to these newly added test cases.
- Update all the log related testNG xml files in <TEST\_HOME>/xml/testng directory
- Run log module with all the tests and make sure all the tests including newly added tests are passing.