# OpenSSO QA Test Automation
# Delegation module Testing Setup & Development

Prepared by Kandula Srinivas

## 1. Introduction

This document describes in detail the QA test automation focusing on Delegation 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

## 3. Organization of Tests

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

- Creating and deleting identities using delegated admin.
- Creating and deleting policies using delegated admin.
- Creating sub realms using delegated admin
- Creating identities under subrealm using delegated admin.
- Modifying service configuration using delegated admin.
- Adding, removing and modifying services to identities.
- Adding and removing members to roles and groups.

### Test organization details:

1. This is how the directory and file's are laid out for Delegation
   - <TEST_HOME>/xml/testng contains delegation testng xml files.
   - <TEST_HOME>/resources/delegation contains all properties files for delegation testing.
   - <TEST_HOME>/source/com/sun/identity/qatest/delegation contains all

sources of Java implementations for delegation.

- <TEST_HOME>/source/com/sun/identity/qatest/common This directory contains the common classes and methods that are used by delegation test module.
- <TEST_HOME>/<server name>/built/classes will contain xml files which will be generated at run time for delegation testing.

2. Delegation 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 delegation tests

1. Deploy Federated Access Manager 8.0 war on a host
2. 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.
3. Change following parameters in <TEST_HOME>/build.properties file
   - Change the value of QATEST_HOME
   - Change the value of TEST_MODULE to “delegation”.
   - Change the value of EXECUTION_MODE to appropriate group name as described in section 3
   - Change the value of REPORT_DIR to desired location
4. Modify LDAP Host name, LDAP port no and LDAP admin password in <TEST_HOME>/resources/delegation/DelegationSMSGlobalDataStore.properties
5. Run following command to execute Delegation module:
   ant -DSERVER_NAME1=<server host name as mentioned in configuration propertiesfile> module

### 4.2. The execution details

When Delegation test is executed, it will :

1. Check and create a new sub realm if it does not exists
2. Create one or more datastore and different type of datastore depending on the datastore configuration specified in DelegationSMSGlobalDataStore.properties
3. Execute the Delegation test cases that are specified in testNG xml file and generate the test results in report directory.
   For each test case to be executed, it will:
   - Read in number of action items and realm name the test is executed in
   - Go through the loop based on the number of action items to
   - Read in the identity attributes, check if the identity type is supported, and perform the action
   - Verify the result of the action.
   - Delete the identities created.
4. Delete the sub realm realm and datastore at the end.

## 5. Tests details

### 5.1 Current Tests classes:

| ***TestClass*** | ***Properties file name*** | ***Description*** |
|---|---|---|
| DelegationConstants.java | | Contains constants. |
| DelegationTest.java | DelegationPolicyGlobal.properties<br>DelegationPolicyTests.properties<br>DelegationPolicyReferral.properties<br>DelegationTests.properties<br>DelegationSMSGlobalDatastoreConfig.properties | A class which reads the test cases properties files and based on the action it executes different methods to create, update, search, delete, add members, remove member from role and group, create and delete policies, create,assign, delete and modify service configurations. |

### 5.2 Properties files

- DelegationGlobal.properties: This property file contains all the messages that are returned by famadm.jsp and are used by delegation qatests for verification.
- DelegationPolicyGlobal.properties: This property file contains URL resource keys, user, group, role prefixes and suffixes for policy definition.
- DelegationPolicyTests.properties: This file contains the properties for the identities and policy creation.

- DelegationPolicyReferral.properties: This file contains the referral resources and the referring organization properties.
- DelegationSMSGlobalDatastoreConfig.properties: This properties file contains the datastore configuration parameters for Sun directory server.
- DelegationTest.properties: This properties file contains test scenarios, each scenario contains multiple actions for creating and deleting users, roles, policies, agents, delegation admins, subrealms and for assigning and modifying service configurations.

### 5.3 Common Files

DelegationCommon.java: A class which contains the common methods for adding and deleting privileges

DelegationConstants.java: This class contains all the Constants.

## 6. Interpreting the Delegation automated testing results

- After execution of delegation 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>-delegation” 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>-delegation”. 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>/<SERVER NAME>/<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 delegation 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.iplanet.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>-delegation-testng.xml and change the value of attribute verbose in tag <suite> to 5 .
  For example, <suite name="ds_ds-delegation" verbose="5">
- For some actions in Delegation tests, it will perform the verification step and display a successful or failed message in the logs file in test result directory. For instance, if the action is to assign privileges after this action it will verify if the privilege is added successfully to the user or not. In case of failure, it will display a message “failed to assign privileges”
- 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 |
| Invalid sso token | Please make sure that the logout action is present at the end of each scenario |

## 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 scenarios:**

- Delegation test case are specified in the properties files in directory <TEST_HOME>/resources/delegation. New testcases scenario should be added to DelegationTest.properties file. Before adding new testcase in the existing scenarios to DelegationTest.properties file, make sure you understand all the existing tests in that class.
  The format and description of each parameters in these properties files available on the header of each properties file.
  <testcase_name><testcase_number>.<action_parameter_key>.<index>= <value> :

for test case group action parameters
or
<testcase_name><testcase_number>.<common_parameter_key>=<value> :

for test case common parameters.
For example,

```
DelegationTest11.action.0=create
DelegationTest11.realm=testrealm
```

Add policy related testcases to DelegationPolicyGlobal.properties, DelegationPolicyReferral.properties and DelegationPolicyTests.properties

- The common parameters for each Delegation test case are:
  realm : realm name that is used to executed Delegation test. Datastore and identities will be created in this realm
  count : number of action group in the test case
  description : description of the test case
  datastore_config_idx: DataStore index for the realm, this index is used for creating the data store using the configuration given in DelegationSMSGlobalDatastoreConfig.properties file

- The action parameters for each delegation test case are:
  action : action to be performed for an identity. The available actions are : create, delete, addmember, removemember, addprivileges, removeprivileges, login, logout, createpolicy, createreferralpolicy, deletepolicy,createsubrealm,addservice, unassignservice, modifyservice. Every scenario should have the login action as the first one and the logout action as the last action

- After adding a new test scenario in the properties file, go to
  <TEST_HOME>/xml/testng
  and edit all <execution mode>-delegation-testng.xml file i.e. ds_ds-delegation-testng.xml.
  Append this entry to each file:

```
<test name="<test name><new test number>">
<parameter name="testNum" value="<new test number>"/>
<parameter name="testName" value="<test name matches with properties file
name>"/>
<groups>
<run>
<include name="<group name>"/>
</run>
</groups>
```

```
<classes>
<class name="com.sun.identity.qatest.delegation.DelegationTest"/>
</classes>
</test>
```

Note:
DelegationTest34 does clean up operation so make sure that it should be the last one to get executed.

**To add a new test case action:**

The new test action should be added to DelegationTest.java class with the following steps:

- Create a new method for this new action with the input data read by testDelegation method.
- Write any reusable functionality as a common method in common classes (DelegationCommon,IDMCommon etc)
- Go to method testDelegation and include this new method in the condition to check for testAction.equals(<new action>)

**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/delegation directory. Please follow naming conventions described in the openSSO QA Test Document for theclass 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 delegation related testNG xml files in <TEST_HOME>/xml/testng directory
- Run delegation module with all the tests and make sure all the tests including newly added tests are passing.