Developing OpenDJ with Maven

Introduction

OpenDJ 3 build has been migrated from Ant to Maven starting from revision 11702 (February, 10 2015).

The OpenDJ server is now located under the master branch as the **opendj-server-legacy** module (use git with a community login):

```
https://stash.forgerock.org/projects/OPENDJ/repos/opendj-public/browse
```

Although it is still possible to run some ant tasks, a full build can't be done any more using Ant, due to incompatible changes related to messages properties files.

Requirements

Minimum requirement is Apache Maven 3.0.4

If you have a version already installed and previously built with Ant, run a clean to ensure all generated classes are deleted:

```
ant clean
```

How-to

You can either build from `trunk/opendj` (do it at least once when starting) or directly from `trunk/opendj/opendj-server-legacy` (once you have built the complete opendj project, provided that not dependent changes occured in another maven sub-module)

- Build the project (Note: it does not run the tests for `opendj-server-legacy` module)

  ```
  mvn clean install
  ```

- Build with precommit (includes checkstyle, copyright check, running tests)

  ```
  mvn clean install -Pprecommit
  mvn clean install -Pprecommit -DskipTests=true      # without tests
  mvn clean install -Pprecommit -Pupdate-copyrights   # also update the copyright dates of modified files
  ```

- Only update the copyright dates of modified files

  ```
  mvn validate -Pupdate-copyrights                                                  # updates copyright
  for files touched compared to branch "origin/master"
  mvn validate -Pupdate-copyrights -DcheckCopyrightDiffReferenceBranchName=branch/name   # updates copyright
  for files touched compared to branch "branch/name"
  mvn validate -Pprecommit -DcheckCopyrightDiffReferenceBranchName=branch/name          # run precommit
  checks comparing touched files to branch "branch/name"
  ```

- Regenerate messages and configuration classes (does not re-compile)

  ```
  mvn process-resources
  ```

- Running tests. `opendj-server-legacy` tests are run in the integration-test phase (not the test phase) because a lot of them requires a running server.

  ```
  mvn install -Pprecommit
  mvn integration-test -Pprecommit  # to stop just after running tests
  ```

  Options to run the tests:

  ```
  mvn integration-test -Pprecommit -Dgroups=group1,group2            # to include TestNG groups
  mvn integration-test -Pprecommit -DexcludedGroups=group1,group2    # to exclude TestNG groups
  mvn integration-test -Pprecommit -Dit.test=MyTest                  # to run only MyTest class
  mvn integration-test -Pprecommit -Dit.test=MyTest,MyTest2          # to run MyTest and MyTest2 class
  mvn integration-test -Pprecommit "-Dit.test=Backend*"              # to run all classes starting with
  "Backend". Note that quotes are mandatory to avoid shell interpretation of the '*'
  mvn integration-test -Pprecommit "-Dit.test=MyTest#myTestMethod"   # to run only myTestMethod() in MyTest
  class
  mvn integration-test -Pprecommit "-Dit.test=Backend*#testProcess*"  # to run all methods starting with
  "testProcess" in all test classes starting with "Backend"
  ```

The full documentation of failsafe plugin, used behind the scene to run TestNG tests, is available here: http://maven.apache.org/surefire/maven-failsafe-plugin/integration-test-mojo.html

If you want to debug the test, add the following option to the maven command line: `"-Dmaven.failsafe.debug="-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000 -Xnoagent -Djava.compiler=NONE""`

If you only want to run the tests, see the Also see forgerock community forum: https://forgerock.org/topic/how-to-_only_-run-tests

- Build with SNMP

```
mvn clean install -Dopendmk.lib.dir=/path/to/lib
```

Tips

Working with an IDE

When you're using Maven in an IDE you often find the need for your IDE to resolve source code and Javadocs for your library dependencies.

There's an easy way to accomplish that goal:

```
mvn dependency:sources
mvn dependency:resolve -Dclassifier=javadoc
```

Running slow tests

In `opendj-server-legacy/pom.xml`, remove or comment the following lines from maven-failsafe-plugin configuration:

```
<property>
  <name>excludegroups</name>
  <value>slow</value>
</property>
```

And then run:

```
mvn integration-test -Pprecommit -Dgroups=slow
```