

ANT to Maven2 Transition Review (fcrepo-394)

approach

1. limit changes to structure
2. maintain a working ANT
3. lay initial foundation

new fedora sub-projects

```
<modules>
  <module>generate</module>
  <module>common</module>
  <module>server</module>
  <module>client</module>
  <module>integrationtest</module>
  <module>localservices</module>
  <module>democontent</module>
  <module>installer</module>
</modules>
```

convention over configuration
project structure

pom.xml [<packaging>pom; <modules>0,1,2]

module0/pom.xml [<packaging>jar]

module0/src/main/java

module0/src/main/resources

module0/src/test/java

module0/src/test/resources

module1/pom.xml [<packaging>war]

module1/src/main/java

module1/src/main/resources

module1/src/main/webapp

module1/src/main/webapp/WEB-INF

module1/src/test/java

module1/src/test/resources

module2/pom.xml

module2/...

build lifecycle phases

'goals' are bound to 'phases'

<http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

- validate
- initialize
- generate-sources
- process-sources
- generate-resources
- process-resources
- compile
- process-classes
- generate-test-sources
- process-test-sources
- generate-test-resources
- process-test-resources
- test-compile
- process-test-classes
- test
- prepare-package
- package
- pre-integration-test
- integration
- post-integration-test
- verify
- install
- deploy

maven plugins used

<http://maven.apache.org/plugins/index.html>

<http://mojo.codehaus.org/plugins.html>

1. axistools-maven-plugin
2. build-helper-maven-plugin
3. buildnumber-maven-plugin
4. exec-maven-plugin
5. maven-antrun-plugin
6. maven-assembly-plugin
7. maven-clean-plugin
8. maven-compiler-plugin
9. maven-jar-plugin
10. maven-javadoc-plugin
11. maven-surefire-plugin
12. maven-war-plugin

notes

1. wsdl2java
2. adds client/stubwrappers to build-src-list
3. makes svn-revision-number and timestamp available
4. executes java commands
5. executes ant tasks
6. creates zips and additional jars
7. deletes files
8. compiles
9. jars
10. javadocs
11. executes junit
12. wars

maven profiles found in fedora

<http://maven.apache.org/guides/introduction/introduction-to-profiles.html>

1. fedora-installer
2. configQ
3. configA
4. configB
5. ant-prepare-jdbc3
6. ant-prepare-jdbc4

fcrepo build sequence

[phase:module:plugin:goal]

```
validate
  top:buildnumber-maven-plugin:create
initialize
generate-sources
  common:axistools-maven-plugin:wsdl2java (api-m & api-a)
  client:build-helper-maven-plugin:add-source (add stubwrappers to source path)
process-sources
  common:exec-maven-plugin:java (BuildAxisStubWrapper, api-m & api-a)
  server:maven-antrun-plugin:run (replace jdbc3/4 tokens)
generate-resources
process-resources
compile
process-classes
  common:maven-clean-plugin:clean
  common:maven-antrun-plugin:run (replace wsdd tokens) - profile:fedora-installer
  server:exec-maven-plugin:java (Release2_1Namespace, create vocabulary.txt) - profile:fedora-installer
generate-test-sources
process-test-sources
generate-test-resources
  server:exec-maven-plugin:java (convert demo objects to mets) - profile:fedora-installer
  server:exec-maven-plugin:java (convert demo objects to atom) - profile:fedora-installer
  server:exec-maven-plugin:java (convert demo objects to atomzip) - profile:fedora-installer
process-test-resources
test-compile
process-test-classes
test
prepare-package
package
  common:maven-jar-plugin:jar (excludes some wsdl-generated classes from artifact)
  common:maven-jar-plugin:test-jar
  server:maven-jar-plugin:test-jar
  server:maven-jar-plugin:jar (create rmi-journal-receiver main/main.jar)
  server:maven-assembly-plugin:single (create rmi-journal-receiver.jar)
  server:maven-assembly-plugin:single (create rmi-journal-receiver.zip)
  server:exec-maven-plugin:java (create axis.Admin generates server-config.wsdd) - profile:fedora-installer
  server:maven-war-plugin:war (create server-webapp.war) - profile:fedora-installer
  server:maven-jar-plugin:jar (create installer main/main.jar) - profile:fedora-installer
  client:maven-jar-plugin:jar (create fedora-client.jar)
  client:maven-jar-plugin:jar (create fedora-messaging-client.jar)
  client:maven-assembly-plugin:single (create messaging-client.zip)
pre-integration-test
integration-test
  top:maven-surefire-plugin:test
  server:maven-surefire-plugin:test
  integrationtest:maven-surefire-plugin:test (skip by default)
  integrationtest:maven-surefire-plugin:test (select config tests) - profile:configQ
  integrationtest:maven-surefire-plugin:test (select config tests) - profile:configA
  integrationtest:maven-surefire-plugin:test (select config tests) - profile:configB
  installer:maven-assembly-plugin:single (create fedora-home.zip) - profile:fedora-installer
  installer:maven-assembly-plugin:single (create fedora-installer.jar) profile:fedora-installer
post-integration-test
verify
install
deploy
```

questions

1. Plans for .classpath / .project?
2. Version numbers? <version/>? or svn#? or both?
3. api.jar needed?

issue to address: auto-generation wsdl artifacts

logical steps

compile BuildAxisStubWrapper.java
execute wsdl2java on api-m & api-a wsdl's
execute BuildAxisStubWrapper on generated api-m/a java files
remove unwanted generated files (..SOAPHTTPImpl & ..ServiceLocator)
replace tokens in [un]deploy.wsdd files
create server-config.wsdd from deploy.wsdd

steps as implemented

generate-sources

common:axistools-maven-plugin:wsdl2java (api-m & api-a)

client:build-helper-maven-plugin:add-source (add stubwrappers to source path)

process-sources

common:exec-maven-plugin:java (BuildAxisStubWrapper, api-m & api-a)

process-classes

common:maven-clean-plugin:clean

common:maven-antrun-plugin:run (replace wsdd tokens)

compile

generate:maven-compiler-plugin:compile (compile BuildAxisStubWrapper.java)

package

common:maven-jar-plugin:jar (excludes some wsdl-generated classes from artifact)

server:exec-maven-plugin:java (create axis.Admin generates server-config.wsdd)

