TEAM Engine Tutorial

Editor: Luis Bermudez (OGC)
Contributor: Richard Martell (Galdos)

April 22, 2013

Comments and additional help @
http://cite.opengeospatial.org/forum

Contents

1 Introduction

The Test, Evaluation, And Measurement (TEAM) Engine is a test harness that executes test suites written using the OGC CTL test grammar or the TestNG framework. It is typically used to verify specification compliance and is the official test harness of the OGC Compliance Testing Program (CITE), where it is used to certify implementations of OGC and ISO geomatics standards.

OGC hosts an official stable deployment of TEAM Engine with the approved test suites: http://cite.opengeospatial.org/teamengine/

OGC also hosts a Beta TEAM Engine with the tests in Beta and with new TEAM Engine functionality: http://cite.opengeospatial.org/te2

2 Prerequisites and Needed Tools

To Build TEAM Engine and the OGC Tests you need the following:

- JAVA 1.6 or later http://www.java.com/en/
- MAVEN 3.0 http://maven.apache.org
- Tomcat 7.0 http://tomcat.apache.org
- An SVN client
- A text editor to open configuration files in XML.

3 Source Code

Sourcecode is available at Sourceforge Repository URL is: https://svn.code.sf.net/p/teamengine/code/

3.1 Repository structure

The sctructure of the repository contains these main folders:

```
branches
tags
trunk
```

3.1.1 Branches

Branches are used for developers to fix bugs, test new features.

3.1.2 tags

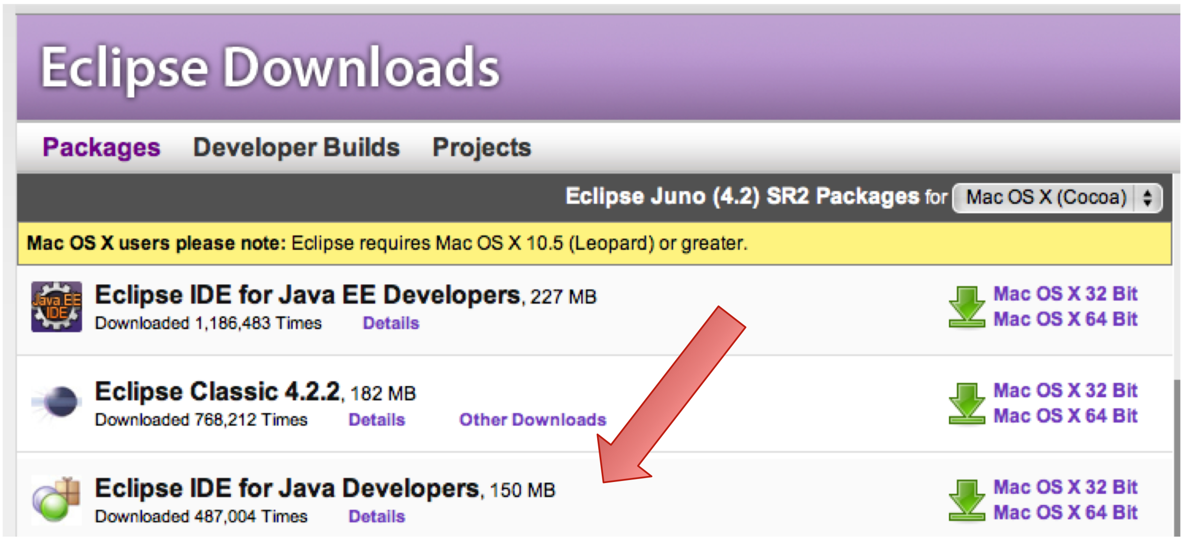
Where all the releases are found, alpha, beta, and production

3.1.3 trunk

Is where the latest development occurs.

3.2 Download Eclipse

Download Eclipse from: http://www.eclipse.org/downloads/



Mac OS X users please note: Eclipse requires Mac OS X 10.5 (Leopard) or greater.

Eclipse IDE for Java EE Developers, 227 MB
Downloaded 1,186,483 Times Details
Mac OS X 32 Bit
Mac OS X 64 Bit

Eclipse Classic 4.2.2, 182 MB
Downloaded 768,212 Times Details Other Downloads
Mac OS X 32 Bit
Mac OS X 64 Bit

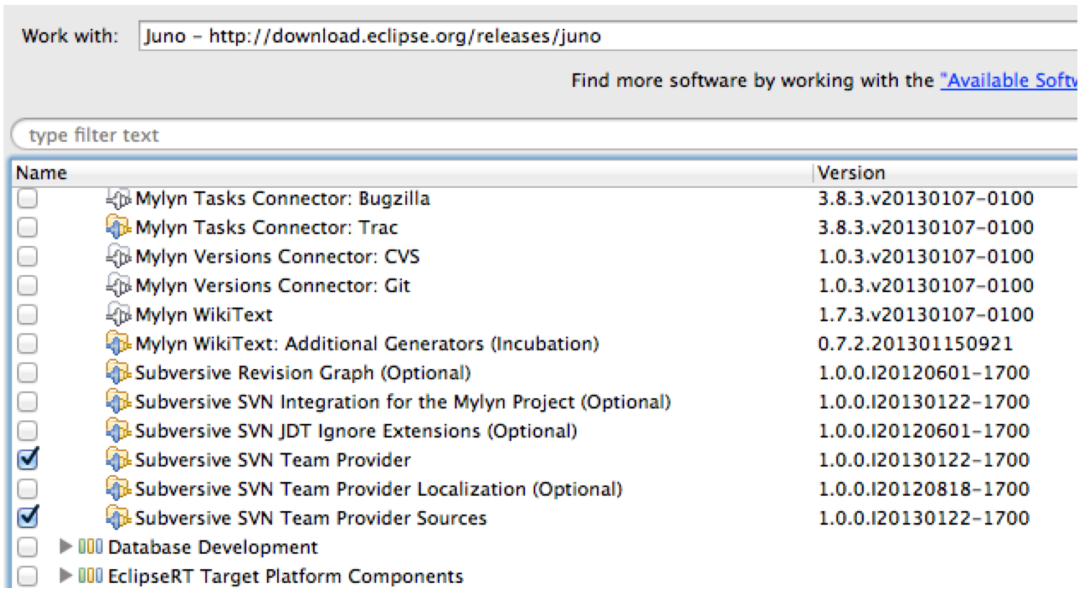
Eclipse IDE for Java Developers, 150 MB
Downloaded 487,004 Times Details
Mac OS X 32 Bit
Mac OS X 64 Bit

Eclipse Juno

3.3 Install SVN

Install subersive following this instructions

3.3.1 Download from Juno Release


Work with: Juno – http://download.eclipse.org/releases/juno

Find more software by working with the "Available Softw

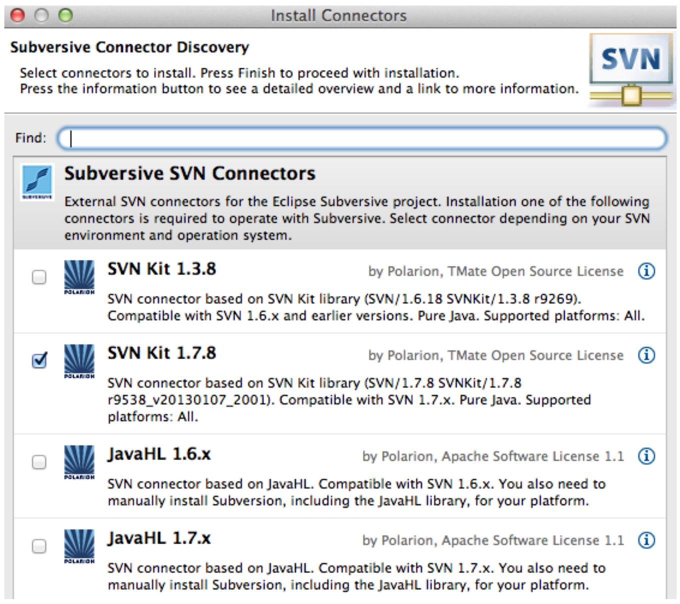
type filter text

	Name	Version
☐	Mylyn Tasks Connector: Bugzilla	3.8.3.v20130107-0100
☐	Mylyn Tasks Connector: Trac	3.8.3.v20130107-0100
☐	Mylyn Versions Connector: CVS	1.0.3.v20130107-0100
☐	Mylyn Versions Connector: Git	1.0.3.v20130107-0100
☐	Mylyn WikiText	1.7.3.v20130107-0100
☐	Mylyn WikiText: Additional Generators (Incubation)	0.7.2.201301150921
☐	Subversive Revision Graph (Optional)	1.0.0.I20120601-1700
☐	Subversive SVN Integration for the Mylyn Project (Optional)	1.0.0.I20130122-1700
☐	Subversive SVN JDT Ignore Extensions (Optional)	1.0.0.I20120601-1700
✓	Subversive SVN Team Provider	1.0.0.I20130122-1700
☐	Subversive SVN Team Provider Localization (Optional)	1.0.0.I20120818-1700
✓	Subversive SVN Team Provider Sources	1.0.0.I20130122-1700
☐	▶ Database Development	
☐	▶ EclipseRT Target Platform Components	

3.3.2 Restart Eclipse

Eclipse will prompt you to restart.

3.3.3 Select connectors


Install Connectors

Subversive Connector Discovery

Select connectors to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.

Find:

Subversive SVN Connectors

External SVN connectors for the Eclipse Subversive project. Installation one of the following connectors is required to operate with Subversive. Select connector depending on your SVN environment and operation system.

☐ **SVN Kit 1.3.8** by Polarion, TMate Open Source License

SVN connector based on SVN Kit library (SVN/1.6.18 SVNKit/1.3.8 r9269). Compatible with SVN 1.6.x and earlier versions. Pure Java. Supported platforms: All.

✓ **SVN Kit 1.7.8** by Polarion, TMate Open Source License

SVN connector based on SVN Kit library (SVN/1.7.8 SVNKit/1.7.8 r9538_v20130107_2001). Compatible with SVN 1.7.x. Pure Java. Supported platforms: All.

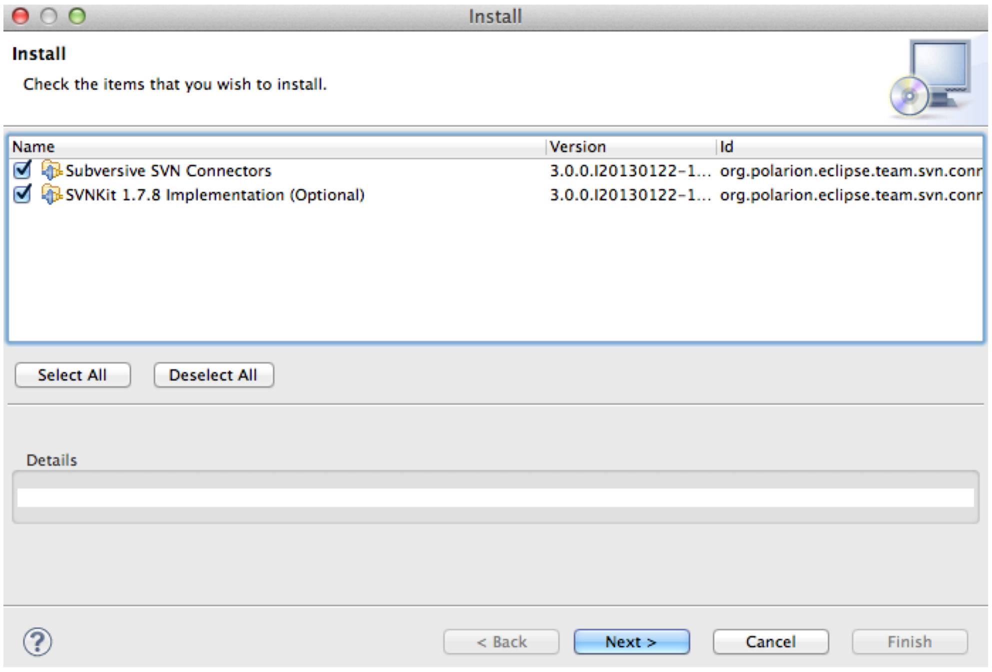
☐ **JavaHL 1.6.x** by Polarion, Apache Software License 1.1

SVN connector based on JavaHL. Compatible with SVN 1.6.x. You also need to manually install Subversion, including the JavaHL library, for your platform.

☐ **JavaHL 1.7.x** by Polarion, Apache Software License 1.1

SVN connector based on JavaHL. Compatible with SVN 1.7.x. You also need to manually install Subversion, including the JavaHL library, for your platform.

3.3.4 Install connectors



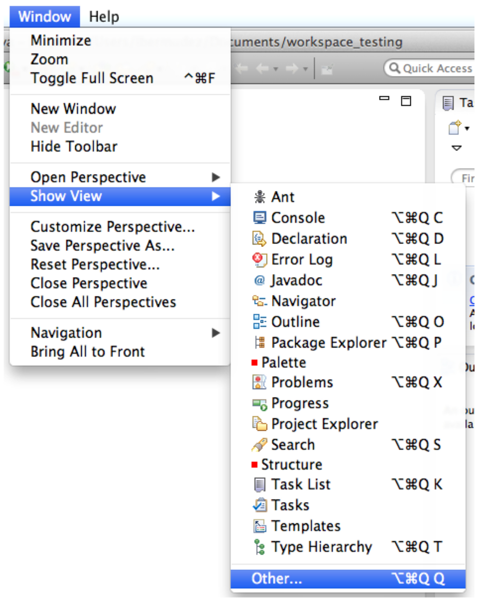
Name	Version	Id
✓ Subversive SVN Connectors	3.0.0.I20130122-1…	org.polarion.eclipse.team.svn.conn
✓ SVNKit 1.7.8 Implementation (Optional)	3.0.0.I20130122-1…	org.polarion.eclipse.team.svn.conn

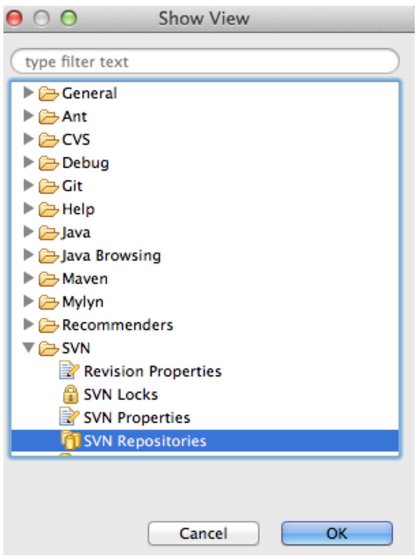
3.3.5 Restart Eclipse

Eclipse will prompt you to restart.

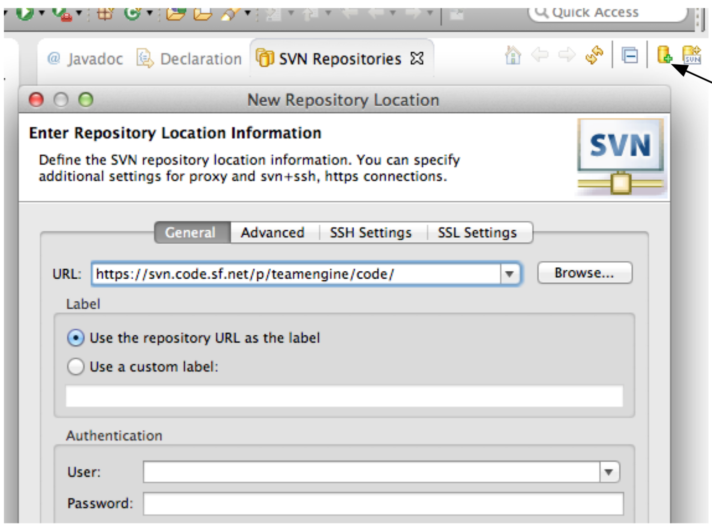
3.4 Connect to Repository

3.4.1 Open SVN Repository view





3.4.2 Type SVN URL



New Repository Location

Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.

General | Advanced | SSH Settings | SSL Settings

URL: https://svn.code.sf.net/p/teamengine/code/

Label

(●) Use the repository URL as the label

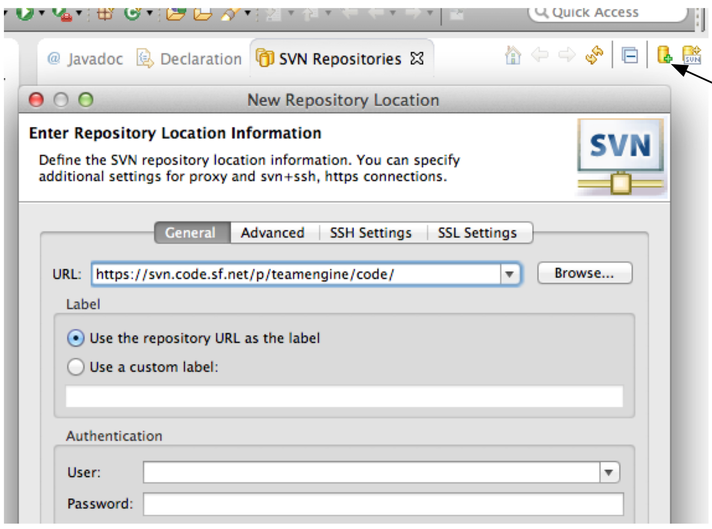
( ) Use a custom label:

Authentication

User:

Password:

3.4.3 Install MVN Subeclipse connector



New Repository Location

Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.

General | Advanced | SSH Settings | SSL Settings

URL: https://svn.code.sf.net/p/teamengine/code/

Label

(●) Use the repository URL as the label

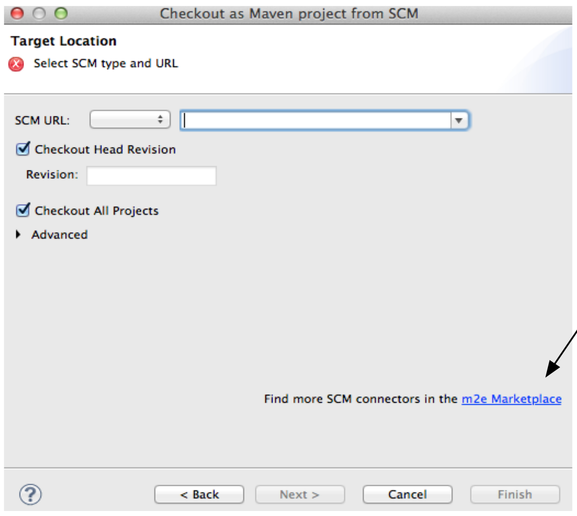
( ) Use a custom label:

Authentication

User:

Password:

3.4.4 Install SVN Maven Connector



Checkout as Maven project from SCM

Target Location

Select SCM type and URL

SCM URL:

☑ Checkout Head Revision

Revision:

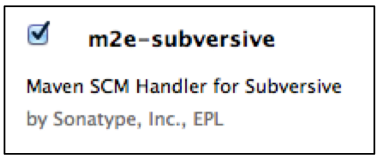
☑ Checkout All Projects

Advanced

Find more SCM connectors in the m2e Marketplace

< Back | Next > | Cancel | Finish

1

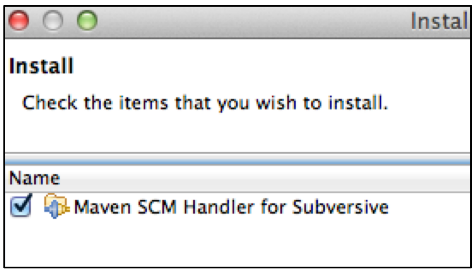


☑ m2e-subversive

Maven SCM Handler for Subversive

by Sonatype, Inc., EPL

2



Instal

Install

Check the items that you wish to install.

Name

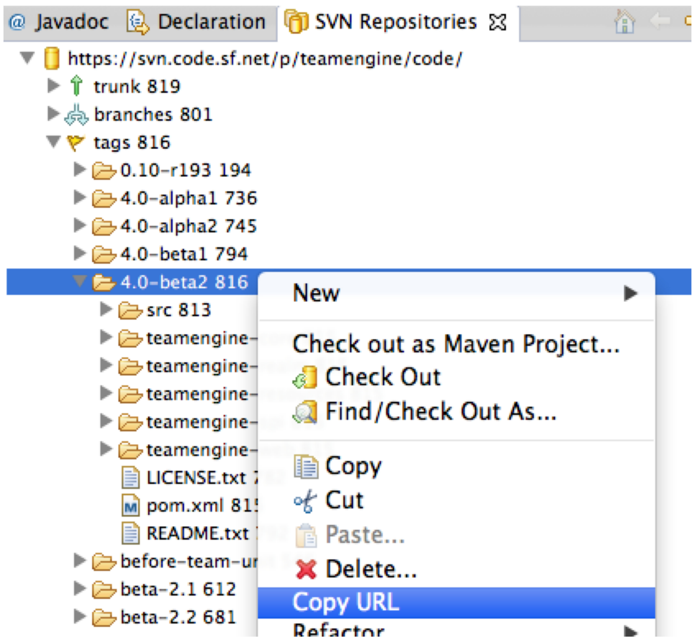
☑ Maven SCM Handler for Subversive

3

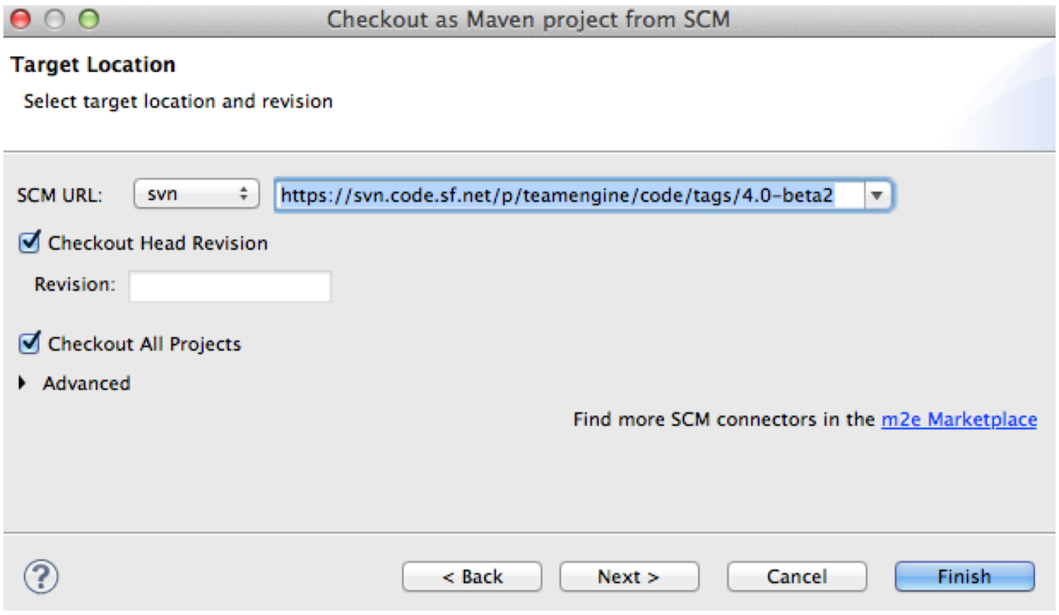
3.5 Download as a Maven project

Copy URL of the branch: For example:

```
https://svn.code.sf.net/p/teamengine/code/tags/4.0-beta2
```



Checkout as Maven from SCM


Checkout as Maven project from SCM

Target Location

Select target location and revision

SCM URL: svn https://svn.code.sf.net/p/teamengine/code/tags/4.0-beta2

☑ Checkout Head Revision

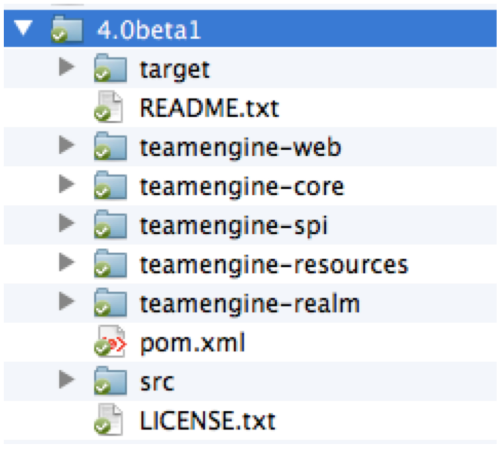
Revision:

☑ Checkout All Projects

Advanced

Find more SCM connectors in the m2e Marketplace

3.6 Package should look like the following


```
▼ 4.0beta1
    ▶ target
      README.txt
    ▶ teamengine-web
    ▶ teamengine-core
    ▶ teamengine-spi
    ▶ teamengine-resources
    ▶ teamengine-realm
      pom.xml
    ▶ src
      LICENSE.txt
```

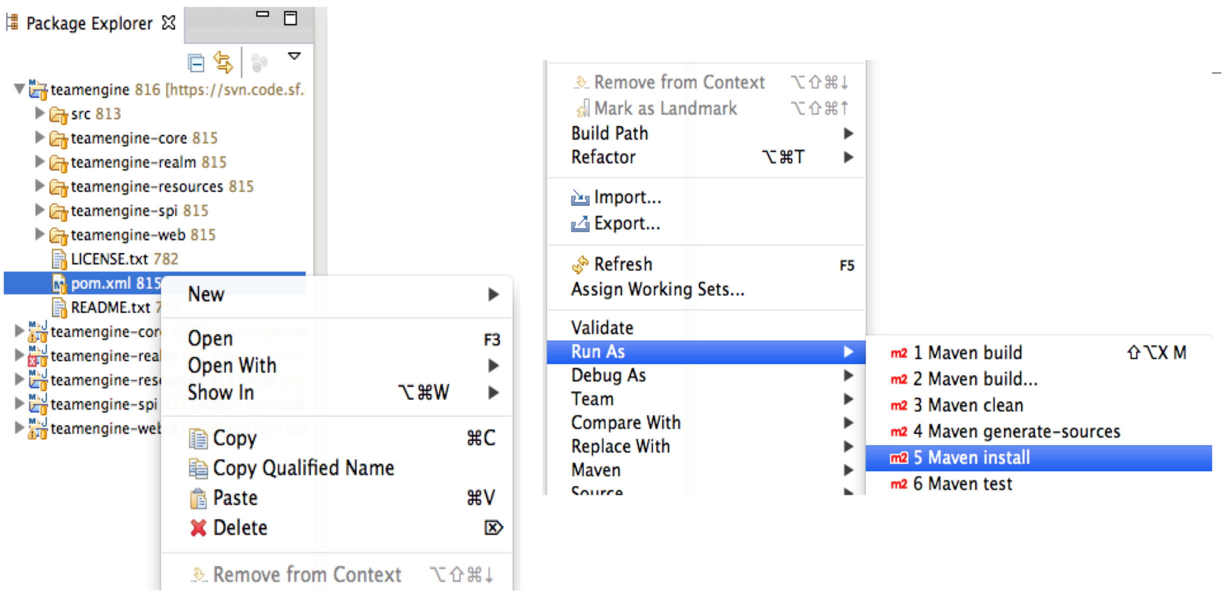
4 Build

4.1 Build code with MAVEN

Execute maven build command from the root of the source code:

```
cd .../teamengine
mvn clean install
```

Can also build via Eclipse, doing right click on the main pom, or the main folder `teamengine`.

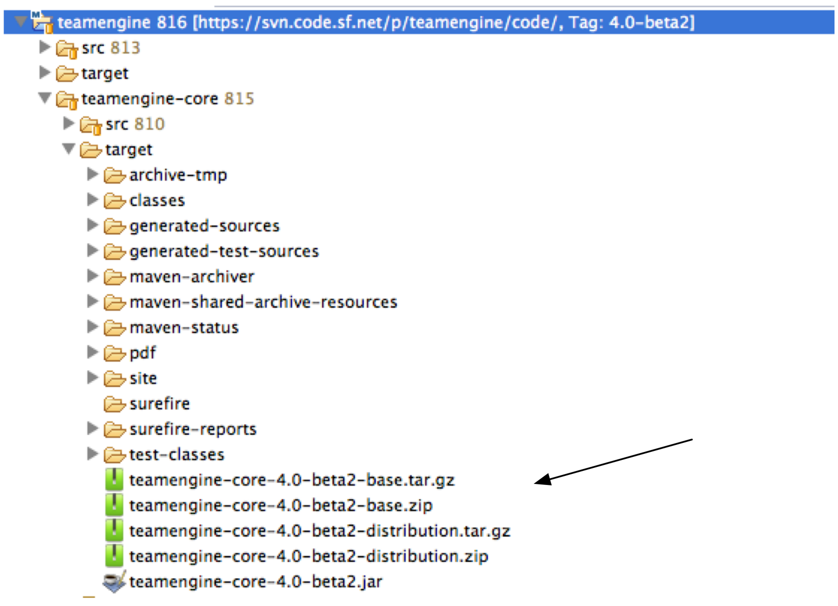


A successful build should look like the following:

```
[INFO] Relativizing decoration links with respect to project URL: http://sourceforge.net/projects/teamengine/
[INFO] Rendering site with org.apache.maven.skins:maven-fluido-skin:jar:1.3.0 skin.
[INFO]
[INFO] --- maven-pdf-plugin:1.2:pdf (pdf) @ teamengine-web ---
[INFO] Skipped report generation.
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] TEAM Engine ....................................... SUCCESS [15.912s]
[INFO] TEAM Engine - Tomcat Realm ........................ SUCCESS [0.617s]
[INFO] TEAM Engine - Shared Resources .................... SUCCESS [0.317s]
[INFO] TEAM Engine - Service Providers ................... SUCCESS [0.901s]
[INFO] TEAM Engine - Core Module ......................... SUCCESS [0.666s]
[INFO] TEAM Engine - Web Module .......................... SUCCESS [0.731s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 20.151s
[INFO] Finished at: Wed Apr 17 06:42:15 EDT 2013
[INFO] Final Memory: 20M/81M
[INFO] ------------------------------------------------------------------------
```

4.2 Check for the created artifacts

After building MAVEN, artifacts are created in the target folders. The zip files created under `teamengine-core` will be used to setup and configure TEAM Engine.



```
teamengine 816 [https://svn.code.sf.net/p/teamengine/code/, Tag: 4.0-beta2]
  src 813
  target
  teamengine-core 815
    src 810
    target
      archive-tmp
      classes
      generated-sources
      generated-test-sources
      maven-archiver
      maven-shared-archive-resources
      maven-status
      pdf
      site
      surefire
      surefire-reports
      test-classes
      teamengine-core-4.0-beta2-base.tar.gz
      teamengine-core-4.0-beta2-base.zip
      teamengine-core-4.0-beta2-distribution.tar.gz
      teamengine-core-4.0-beta2-distribution.zip
      teamengine-core-4.0-beta2.jar
```

4.3 Configure TE_BASE

- Create a TE_BASE directory
- Unpack the content of **teamengine-core-4.0-betaX-base.zip**
- Make `TE_BASE` and environmental variable. Information about setting up environmental variables **can be found here:**
  - For MAC
  - For Linux
  - Window and more information at wikipedia
- **Concretely for MAC::**

  export TE_BASE=/Users/bermudez/Documents/Dropbox/software/te/TE_BASE

4.4 Understand TE_BASE

TE_Base structure is as follows:

```
TE_BASE
  |-- config.xml   # main configuration file
  |-- resources/   # shared test suite resources
  |-- scripts/     # CTL test scripts
  |-- work/        # teamengine work directory
  +-- users/       # user account details and test run outputs
     |-- {user1}/
     |-- {user2}/
     +-- ...
```

4.5 Locate a simple ctl Test

TE_BASE comes with a simple ctl script, **note.ctl**. It is located under the scripts directory:

```
|-- scripts/
   note.ctl
```

4.6 Simple Run TEAM Engine

The build created a file `teamengine-core-4.0-beta2-distribution.zip`. Unpack the file. Should look like the following:

```
teamengine-core-4.0-beta2-distribution
  |-- bin/  # shell scripts (windows, unix)
    |-- unix
    |-- windows
  |-- lib/   # supporting libraries
  |-- resources/ # classpath resources (stylesheets, schemas, etc.)
```

Go the bin folder and select either unix or windows to run test command: `unix/test.sh` or `windows/test.bat`.

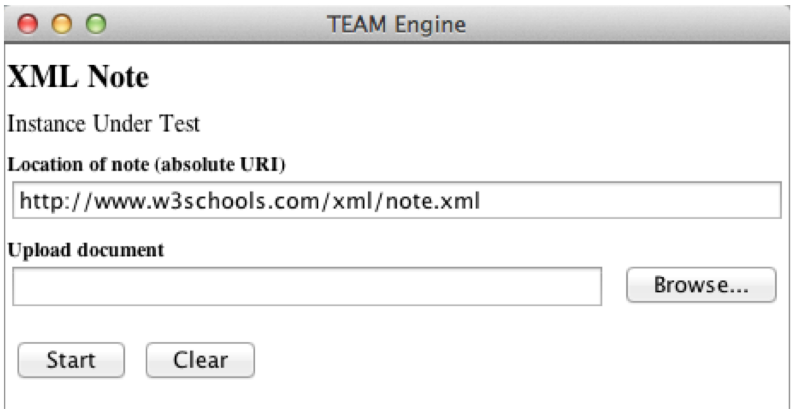
To run the command it is necessary to provide a parameter `-script`. For example:

```
./test.sh -source=note.ctl
```

If the TEAN Engine has properly being installed, the command prompt should show a message like the following:

```
  Luiss-MacBook-Pro:unix lbermudez$ ./test.sh -source=note.ctl
  Testing suite note:note-test in Test Mode with defaultResult of Pass ...
  Testing note:main type Mandatory in Test Mode with defaultResult Pass (s0004)...
  Assertion: The note is valid.

And a popup window should appear
```



XML Note

Instance Under Test

Location of note (absolute URI)

http://www.w3schools.com/xml/note.xml

Upload document

The popup window contains a default link to note to be tested located in this URL: http://www.w3schools.com/xml/note.xml. Clicking on Start will start the test. The test should failed and the terminal should provide the following:

```
Testing suite note:note-test in Test Mode with defaultResult of Pass ...
Testing note:main type Mandatory in Test Mode with defaultResult Pass (s0004)...
      Assertion: The note is valid.
Testing note:check-heading type Mandatory in Test Mode with defaultResult Pass (s0004/d1e97_1)...
         Assertion: The heading contains more than whitespace.
      Test note:check-heading Passed
Testing note:check-user type Mandatory in Test Mode with defaultResult Pass (s0004/d1e102_1)...
         Assertion: The 'to' user is valid.
      Test note:check-user Passed
Testing note:check-user type Mandatory in Test Mode with defaultResult Pass (s0004/d1e107_1)...
         Assertion: The 'from' user is valid.
      Test note:check-user Failed
   Test note:main Failed - Inherited)
Suite note:note-test Failed
```

5 OGC tests scripts

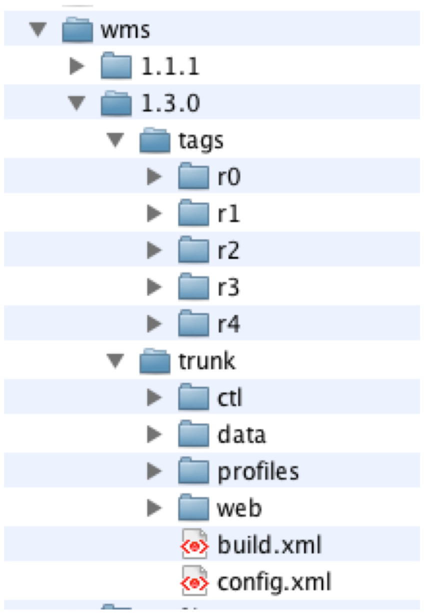
5.1 Understanding OGC Tests Structure

OGC Tests can be written either in CTL (Compliance Test Language) or TestNG. Tests are located at the public OGC SVN Repository:

CTL tests are located at https://svn.opengeospatial.org/ogc-projects/cite/scripts/ TestNG test are located at https://svn.opengeospatial.org/ogc-projects/cite/ets

5.1.1 CTL tests structure

The CTL tests are structured as follows:

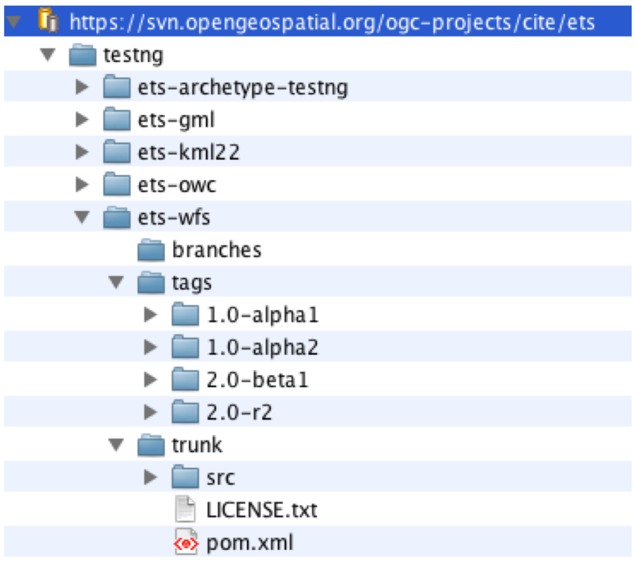


```
▼ wms
    ▶ 1.1.1
    ▼ 1.3.0
        ▼ tags
            ▶ r0
            ▶ r1
            ▶ r2
            ▶ r3
            ▶ r4
        ▼ trunk
            ▶ ctl
            ▶ data
            ▶ profiles
            ▶ web
              build.xml
              config.xml
```

The trunk contains the latest version and versions are tagged for deployment in teamengine OGC web site.

5.1.2 TestNG test structure

The TestNG tests are structured as follows:



```
https://svn.opengeospatial.org/ogc-projects/cite/ets
  testng
    ets-archetype-testng
    ets-gml
    ets-kml22
    ets-owc
    ets-wfs
      branches
      tags
        1.0-alpha1
        1.0-alpha2
        2.0-beta1
        2.0-r2
      trunk
        src
        LICENSE.txt
        pom.xml
```

The trunk contains the latest version and versions are tagged for deployment in teamengine OGC web site. The TestNG tests also follows a MAVEN structure and have a pom with the main configuration.

5.2 Configuring CTL test in TEAM Engine

To make available the OGC tests in TEAM Engine the tests need to be placed at the TE_BASE/scripts directory and the TE_BASE/config.xml file is needs to be updated accordingly.

5.2.1 Copying CTL Tests in TEAM Engine

The tests can be copied manually or using a script.

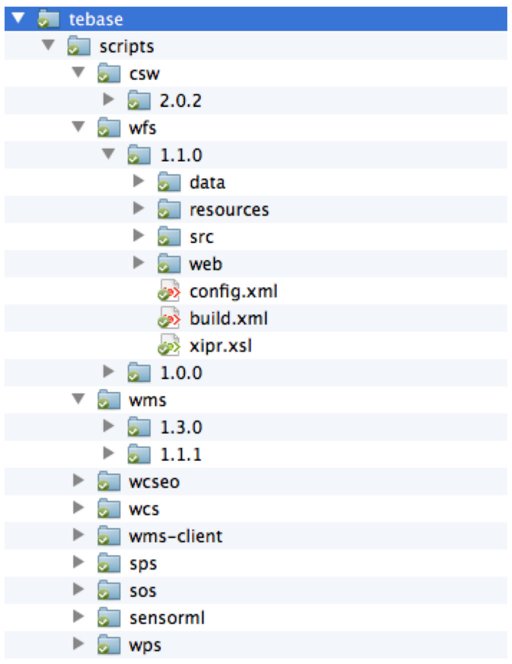
As a convenience, the shell script `export-ctl` may be run to export CTL test suites from the official OGC repository. The location of a CSV file is passed as the first argument to the script. Each record in the file should contain two fields: a Subversion URL, and a local path name relative to TE_BASE/scripts. For example one row of the file might be as follows:

```
https://svn.opengeospatial.org/ogc-projects/cite/scripts/sensorml/1.0.1/trunk,sensorml/1.0.1
```

The `ctl-suites-dev.csv` file can be found in the same directory as the shell scripts; it includes entries for the latest development versions of several OGC test suites. Running the following command will populate the `TE_BASE/scripts` directory with these test suites:

```
$ export TE_BASE=/some/path
$ ./export-ctl.sh ctl-suites-dev.csv
```

After running the command the TE_BASE should look like the following:



```
tebase
  scripts
    csw
      2.0.2
    wfs
      1.1.0
        data
        resources
        src
        web
        config.xml
        build.xml
        xipr.xsl
      1.0.0
    wms
      1.3.0
      1.1.1
    wcseo
    wcs
    wms-client
    sps
    sos
    sensorml
    wps
```

Once the tests are properly installed in the local TEAM Engine, it is possible to run OGC test suites. For example:

```
./test.sh -source=csw/2.0.2/src/main.xml
```

5.2.2 Building and Copying TestNG Tests in TEAM Engine

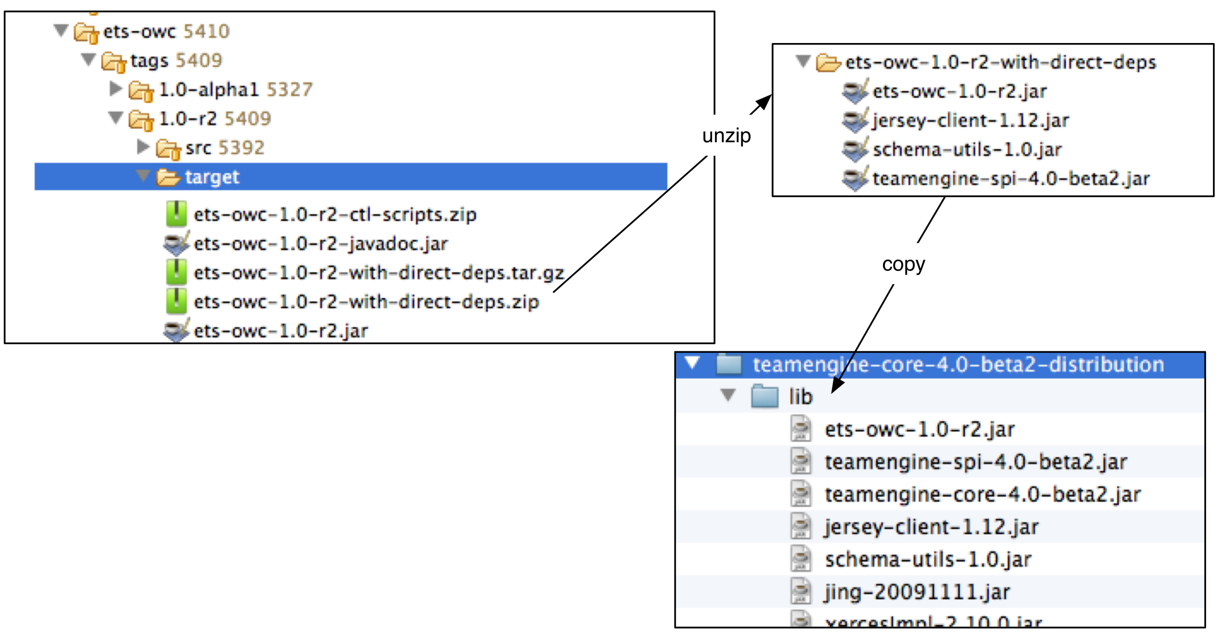
For TestNG Tests, go under the folder of the test and run:

```
mvn install
```

For example to build 1.0-alpha1 version of ets-owc::

$ cd ets-owc/tags/1.0-alpha1 $ mvn package

It will create a target folder with the builds. Unzip the zip files with the test binaries. The binaries and dependencies are found in the zip file that has the name of the test the version number and "-with-direct-depd". Place the jars under the teamengine core lib directory (e.g. `teamengine-core-4.0-beta2-distribution`).



```
ets-owc 5410
  tags 5409
    1.0-alpha1 5327
    1.0-r2 5409
      src 5392
      target
        ets-owc-1.0-r2-ctl-scripts.zip
        ets-owc-1.0-r2-javadoc.jar
        ets-owc-1.0-r2-with-direct-deps.tar.gz
        ets-owc-1.0-r2-with-direct-deps.zip
        ets-owc-1.0-r2.jar
```

```
ets-owc-1.0-r2-with-direct-deps
  ets-owc-1.0-r2.jar
  jersey-client-1.12.jar
  schema-utils-1.0.jar
  teamengine-spi-4.0-beta2.jar
```

```
teamengine-core-4.0-beta2-distribution
  lib
    ets-owc-1.0-r2.jar
    teamengine-spi-4.0-beta2.jar
    teamengine-core-4.0-beta2.jar
    jersey-client-1.12.jar
    schema-utils-1.0.jar
    jing-20091111.jar
```

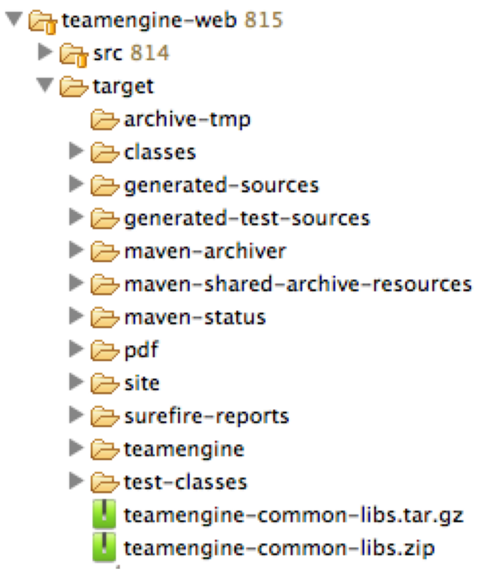
The target folder also created a *.ctl-scripts.zip file. Unzip the files and copy them under the TE_BASE/scripts.

When builduing to code, this is running at the root level (teamengine):

```
mvn install
```

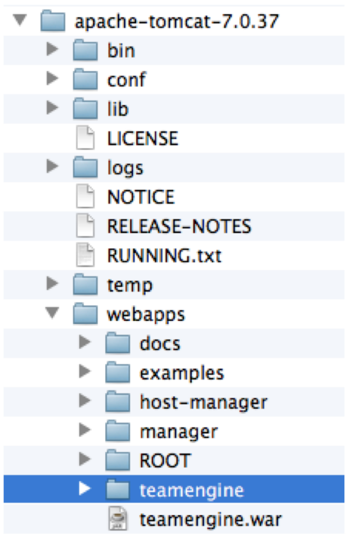
5.3 Install war

Various files under `target` were created. Under the folder `teamengine-web/target` a war file was created.



```
teamengine-web 815
  src 814
  target
    archive-tmp
    classes
    generated-sources
    generated-test-sources
    maven-archiver
    maven-shared-archive-resources
    maven-status
    pdf
    site
    surefire-reports
    teamengine
    test-classes
    teamengine-common-libs.tar.gz
    teamengine-common-libs.zip
```

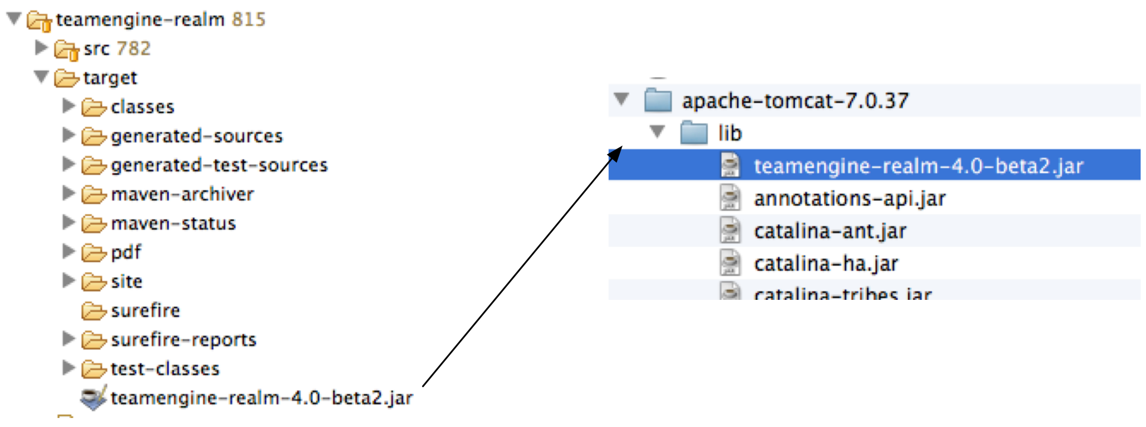
Copy the war file under webapps in tomcat.



```
apache-tomcat-7.0.37
  bin
  conf
  lib
  LICENSE
  logs
  NOTICE
  RELEASE-NOTES
  RUNNING.txt
  temp
  webapps
    docs
    examples
    host-manager
    manager
    ROOT
    teamengine
    teamengine.war
```

5.4 Install Realm

Under the folder `teamengine-realm/target` a jar file was created. This jar manages a simple authentication and management of users using TEAM Engine. Copy this file under `lib` in the web server.



5.5 Configure Tomcat

Open bin/startup.sh (or startup.bat if running in windows). And before PRGDIR=`dirname "$PRG"` add the following two first lines:

```
# define CATALINA_OPTS for TEAM Engine
export CATALINA_OPTS="-server -Xmx1024m -XX:MaxPermSize=128m -DTE_BASE=$TE_BASE -Dderby.system.home=$DERBY_DATA"


PRGDIR=`dirname "$PRG"`
EXECUTABLE=catalina.sh
```

5.6 Configure TE_BASE scripts

Register the tests that will appear in the web interface in `TE_BASE/config.xml`. For example add the following inside <stripts></scripts>:

```
<organization>
 <name>OGC</name>
 <standard>
   <name>OGC KML</name>
   <version>
     <name>2.2</name>
     <suite>
       <namespace-uri>http://www.opengis.net/cite/kml22</namespace-uri>
       <prefix>tns</prefix>
       <local-name>ets-kml22-2.2-r1</local-name>
       <title>KML 2.2 Validator</title>
       <description>Verifies the structure and content of KML 2.2 documents.</description>
     </suite>
     <revision>
       <name>2.2-r1</name>
       <status>Alpha</status>
       <sources>
```

```
          <source>kml22/2.2-r1/kml22-suite.ctl</source>
        </sources>
        <webdir>kml22/2.2-r1/web</webdir>
      </revision>
    </version>
  </standard>
</organization>
```

5.7 Add the test libraries

For TestNG tests copy the test jars under lib.

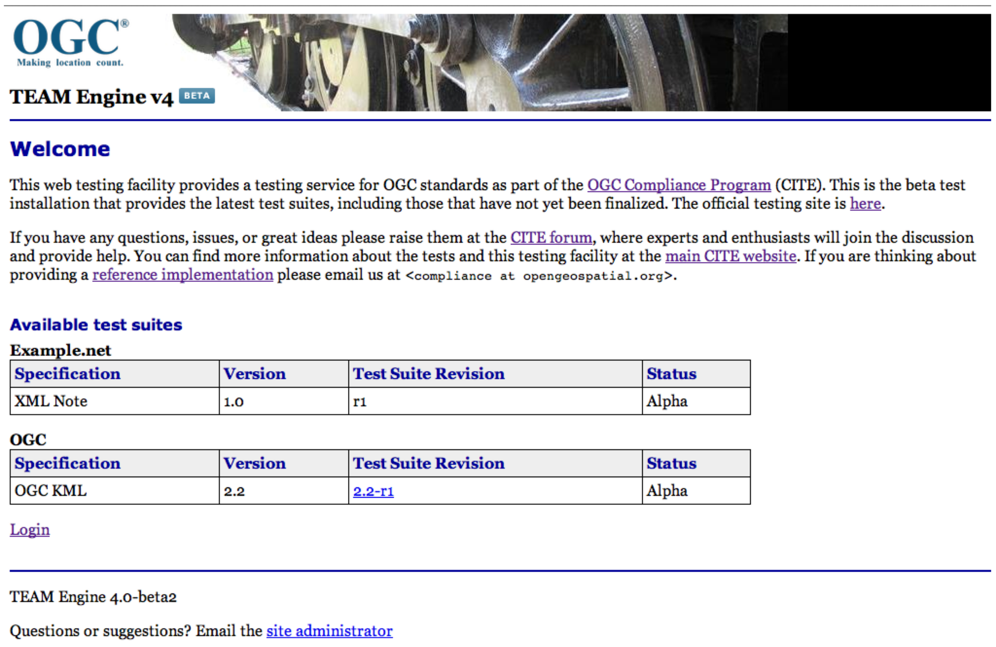
5.8 Start Tomcat

For example:

```
$ cd /Applications/apache-tomcat-7.0.37
$ cd bin
$ ./startup.sh
```

When typing: http://localhost:8080/teamengine/

The TEAM Engine Web Inerface should appear like the following:



Welcome

This web testing facility provides a testing service for OGC standards as part of the OGC Compliance Program (CITE). This is the beta test installation that provides the latest test suites, including those that have not yet been finalized. The official testing site is here.

If you have any questions, issues, or great ideas please raise them at the CITE forum, where experts and enthusiasts will join the discussion and provide help. You can find more information about the tests and this testing facility at the main CITE website. If you are thinking about providing a reference implementation please email us at <compliance at opengeospatial.org>.

Available test suites

Example.net

Specification	Version	Test Suite Revision	Status
XML Note	1.0	r1	Alpha

OGC

Specification	Version	Test Suite Revision	Status
OGC KML	2.2	2.2-r1	Alpha

Login

TEAM Engine 4.0-beta2

Questions or suggestions? Email the site administrator