Collaborative, Open Source Web Capability for Testing Compliance to Geospatial Web Services Standards

The 2011 International Conference on Collaboration Technologies and Systems (CTS 2011)

Luis Bermudez
lbermudez@opengeospatial.org
May 23, 2011
Agenda

• Why Open Geospatial Standards
• Introduction to the Open Geospatial Consortium (OGC)
• Need for Compliance Testing
• Compliance Program
• Overview of OGC Testing Facility - TEAM Engine
• Introduction to the Compliance Test Language (CTL)
• Demonstration of Web Feature Service Testing
• Resources and Getting Started
• Running TEAM Engine in a Local Environment
• Setting TEAM Engine in a Development Environment
• Getting Involved
“When you are delivering spatial web services on behalf of 20 government agencies to more than a 1000 organizations running their own spatial systems, you need standards.”
Why Standards for Data Collaboration?

Kylie Armstrong
Business Development
Western Australian Land Information Authority
Landgate, Australia

“Using the internationally recognized OGC and ISO standards for both the architecture and web services has been essential to our success.”
Policy Requirements for Open Standards

- Global Earth Observation System of Systems (GEOSS)
- NATO C3
- US NGA
- US Federal Geographic Data Committee
- European INSPIRE Directive
- European Space Agency
- Local, national, regional government
- Science and Research
Digital Norway – Land Use

Municipal Areas

- Fisheries
- Waste Water Outflow
- Water Supply

Flood Risk Areas

- Demography
- Biodiversity
- Agriculture and Forestry

Land Use

© 2011 Open Geospatial Consortium, Inc.

Friday, May 20, 2011
Debris Flow Monitoring - Taiwan
Debris Flow Monitoring - Taiwan
Debris Flow Monitoring - Taiwan

• Typhoons and earthquakes trigger landslides and flooding on a frequent basis
Debris Flow Monitoring - Taiwan

- Typhoons and earthquakes trigger landslides and flooding on a frequent basis
- OGC standards used with an array of spatial data and sensors to forecast, detect, alert and respond to debris flow situations.
Debris Flow Monitoring - Taiwan

• Typhoons and earthquakes trigger landslides and flooding on a frequent basis

• OGC standards used with an array of spatial data and sensors to forecast, detect, alert and respond to debris flow situations.

• Rapidly deployed network of debris flow sensors, and distributed services performing sensor data analysis and processing
Sensor Web Enablement Standards Application
Ocean Observation

Core DIF Standards
These are some of the basic standards and specifications adopted by the NOAA IOOS Data Integration Framework.

- OGC Sensor Observation Service (SOS) specification: NOAA IOOS uses this service type to provide access to in-situ oceanographic data in an XML encoding defined by the GML application schema referenced above.
- OGC Web Coverage Service (WCS) specification: this service type to provide access to gridded data in binary formats such as NetCDF and GeoTIFF.
- OPeNDAP Information: This service type is used to provide access to gridded remotely sensed data such as NetCDF and GeoTIFF.
- OGC Web Map Service (WMS) specification: used to provide georeferenced images of data.

Real-Time Data from an OGC Sensor Web:
This interoperability demonstration represents an effort to develop a Web Services Infrastructure for Ocean Observing that is enabling observing systems, to more closely to the vision of “network as platform.” We are seeking participants who would like to share their in-situ observation data via ISO 19115 based Web Services. To learn more, visit the OGC 11IAA28 website.

2005 Platforms reporting Click the station icon on the map for the latest observations.

© 2011 Open Geospatial Consortium, Inc.
Open Geospatial Consortium (OGC)

To serve as a global forum for and lead the development, promotion and harmonization of open and freely available geospatial standards.
Over 420 Member Organizations
Significant Government Participation

• US DHS
• US EPA
• US FAA
• US NASA
• USGS
• US NGA
• US Census
• US NOAA
• JPEO
• Oakridge National Lab
• Natural Resources Canada
Over 35 Adopted Standards

• Data Services
  – Sensor Observation Service (SOS)
  – Web Coverage Service
  – Web Feature Service
  – Web Map Service ..

• Catalogue Services
  – Catalogue Service

• Processing Services
  – Open Location Services (OpenLS)
  – Coordinate Transformation Service
  – Sensor Planning Service (SPS)
  – Web Processing Service (WPS)
Over 35 Adopted Standards

- **Encodings**
  - Geography Markup Language (GML)
  - Styled Layer Descriptor (SLD)
  - Transducer Markup Language (TML)
  - Sensor Model Language (SensorML)
  - CityGML
  - Web Map Context (WMC)
  - Observations & Measurements (O&M)
  - Filter Encoding
  - KML
  - Symbology Encoding
  - GML in JPEG 2000
  - ....
OGC Activities Driven by Community Needs

- Education & Research
- Sustainable Development
- Utilities
- Emergency Services
- Consumer Services
- Energy
- Geosciences
- Health
- E-Government

Global Spatial Data Infrastructure
Domain Working Groups

Meteorology Domain Working Group
• Advancing Practices to share met/ocean data
• Lead by WMO
Domain Working Groups

Meteorology Domain Working Group
• Advancing Practices to share met/ocean data
• Lead by WMO
Domain Working Groups

Emergency and Disaster Management DWG

• Provide requirements and Best practices for web service interfaces, models and schemas
• for enabling the discovery, access, sharing, analysis, visualization and processing of information to
• the forecasting, prevention, response to and recovery from emergency and disaster situations.
..how do we know if a standard works?

I have not failed, I’ve just found 10,000 ways that won’t work.

Thomas Edison
Interoperability Program

OGC Organization
Committees
Board
Staff

Specification Program
DWG
SWG
OAB
CITE SC

Outreach and Community Adoption Program (OCAP)

Compliance & Testing
Team Engine

Interoperability Program
Testbeds, pilots, experiments
IP Management team

Plan
Do
Check

Act

Friday, May 20, 2011
40+ Interoperability Program initiatives since 1999.
ADC activities including: Architecture Implementation Pilot (AIP)
Task AR-09-01b
GEOSS Architecture Implementation Pilot

User Needs, Scenarios

requirements

Design, Develop, Deploy

ADC activities including: Architecture Implementation Pilot (AIP)
Task AR-09-01b

SBA Tasks, UIC, CBC, STC
GEOSS Architecture Implementation Pilot

ADC activities including: Architecture Implementation Pilot (AIP)
Task AR-09-01b

User Needs, Scenarios

Design, Develop, Deploy

Operational Capability

GEOSS Common Infrastructure (GCI)
Task AR-09-01a

SBA Tasks, UIC, CBC, STC

requirements

persistent implementation

support
Need for Compliance Testing
Compliance Program Goals

• Provide robust standard compliance solutions for communities applying/using geospatial software/technologies

• Provide a process whereby compliance for OGC specifications can be tested. Validate certified product compliance with OGC standards and provide Seal of Approval.

• Increase systems interoperability

• Reduce technology risks
Compliance Numbers (Mar 2011)

- More than 10 years providing certification
- Web Testing Engine - open source - operational since 2007
- More than 600 implementing products in the market
- More than 260 compliant products in the market
Compliance Procedure

1) Developers go to online test engine

2) Fill the Test Results (TSR) form

3) Pay License Fee

4) Get compliance certificate

5) Use certification mark

More information: http://bit.ly/gTmmSo
Agenda

- Overview OGC Testing Facility - TEAM Engine
Online Facility TEAM Engine

http://cite.opengeospatial.org/teamengine/

---

**Welcome**

The Test, Evaluation, And Measurement (TEAM) Engine is a test script interpreter. It executes test scripts written in CTL to verify that an implementation of a specification complies with the specification.

The following test suites are available:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Test Suites</th>
<th>Test Datasets</th>
<th>Release Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog Service-Web (CSW) 2.0.2</td>
<td>r2</td>
<td>r2</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Sensor Observation Service (SOS) 1.0.0</td>
<td>r0</td>
<td>NA</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Sensor Planning Service (SPS) 1.0</td>
<td>r0</td>
<td>NA</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Coverage Service (WCS) 1.0.0</td>
<td>r2</td>
<td>NA</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Coverage Service (WCS) 1.1.1</td>
<td>r0</td>
<td>NA</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Feature Service (WFS) 1.0.0</td>
<td>r3</td>
<td>r3</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Feature Service (WFS) 1.1.0</td>
<td>r4</td>
<td>r4</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Map Context (WMC) 1.1.0</td>
<td>r0</td>
<td>NA</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Map Service (WMS) 1.1.1</td>
<td>r2</td>
<td>r2</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Map Service (WMS) 1.3.0</td>
<td>r1</td>
<td>r1</td>
<td>relnotes.txt</td>
</tr>
<tr>
<td>Web Registry Service (WRS) 1.0</td>
<td>r0</td>
<td>r0</td>
<td>relnotes.txt</td>
</tr>
</tbody>
</table>

⚠️ It may be necessary to load test data before running a test suite!

Start Testing

---

MOZILLA PUBLIC LICENSE
Version 1.1

---

Java – Web Server
Tester selects test suite

Tester names session
Tester provides end point of the service

Console provides feedback on test assertions

UpdateSequence Values

The WMS spec allows servers to use an UpdateSequence specification. If the server advertises an UpdateSequence the UpdateSequence behavior automatically. However, tests may not always be correct. If you suspect a problem:

(Fill in these boxes if the Manual option is selected)

Options

- BASIC: Test basic functionality that depends on the CITE dataset. This option is required for certification.
- QUERYABLE: Test GetFeatureInfo functionality that depends on the CITE dataset.
- RASTER ELEVATION: Test the elevation dimension using the cite:Terrain dataset.
- VECTOR ELEVATION: Test the elevation dimension using the cite:Lakes vector dataset.
- TIME: Test the time dimension using the cite:Autos dataset.
- RECOMMENDATIONS: Test functionality which is recommended in the specification.

Tester Selects Options
## Results for session s0006

### Test Suite: Web Map Service (WMS) 1.3.0

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test main:main (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:main (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:basic-polygons-sanity-check (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:blue-lake-sanity-check (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:layer-order (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:aspect-ratio (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:exceptions-inimage (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test interactive:fees-and-access-constraints (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test main:options-requirements (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test main:gif-or-png (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test main:std-data-present (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test main:getfeatureinfo-supported (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test main:std-data-queryable (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:main (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:version-negotiation (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:negotiate-no-version (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:negotiate-basic_elements-version (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:negotiate-higher-version (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:negotiate-lower-version (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:reserved-chars (View Details)</td>
<td>Passed</td>
</tr>
<tr>
<td>Test basic_elements:escaped-chars (View Details)</td>
<td>Passed</td>
</tr>
</tbody>
</table>
Online Facility TEAM Engine

No failures or warnings.

Email Results to Compliance Program
Agenda

- Introduction to the Compliance Test Language (CTL)
XML grammar for documenting and scripting suites of tests for verifying that an implementation of a specification complies with the specification.
Writing a simple Hello World Test Script

The following is a simple Hello World CTL test script.

```xml
<ctl:test
 xmlns:ctl="http://www.occamlab.com/ctl"
 xmlns:my="urn://mynamespace"
 name="my:hello"
>
 <ctl:assertion>This test displays a greeting.</ctl:assertion>
 <ctl:code>
   <ctl:message>Hello, world!</ctl:message>
 </ctl:code>
</ctl:test>
```

This is one of the simplest test scripts that can be written. It contains a single test. Tests are identified by a namespace qualified name attribute. In this case, we have made up our own namespace "urn://mynamespace" identified by the prefix "my" which we are using in the test name. A test has an assertion, which is a statement that is true if what is being tested is compliant. It also has code that determines whether the assertion is true or false. This test isn't really testing anything. The code merely displays a greeting on the console, and the assertion just describes what the code does.

To execute the test, use the TEAM Engine test utility.

```
C:\teamengine2.0>bin\test -source=hello.ctl -test=my:hello
```
Structure CTL Tests
Suite and Profiles
<ctl:test name="test:base_main">
  <ctl:param name="x"/>
  <ctl:assertion>Base Main</ctl:assertion>
  <ctl:code>
    <ctl:message>X: <xsl:value-of select="$x"/></ctl:message>
    <ctl:call-test name="test:pass"/>
    <ctl:call-test name="test:fail"/>
  </ctl:code>
</ctl:test>

<ctl:test name="test:pass">
  <ctl:assertion>Pass</ctl:assertion>
  <ctl:code/>
</ctl:test>

<ctl:test name="test:fail">
  <ctl:assertion>Fail</ctl:assertion>
  <ctl:code>
    <ctl:fail/>
  </ctl:code>
</ctl:test>

<ctl:test name="test:profile_a_main">
  <ctl:param name="x"/>
  <ctl:assertion>Profile A Main</ctl:assertion>
  <ctl:code>
    <ctl:message>X: <xsl:value-of select="$x"/></ctl:message>
  </ctl:code>
</ctl:test>
Test and Assertions

<ctl:test name="example:zulu">
  <ctl:param name="time">time string</ctl:param>
  <ctl:assertion>
    If the hours field is included in ${time},
    the suffix Z (for zulu) is required.
  </ctl:assertion>
  <ctl:link title="WMS 1.1.1 Section B.2.1">wms111.html#b_2_1</ctl:link>
  <ctl:code>
    <xsl:if test="contains(${time}, 'T')">
      <xsl:variable name="len" select="string-length(${time})"/>
      <xsl:if test="not(substring(${time}, $len) = 'Z')">
        <ctl:fail/>
      </xsl:if>
    </xsl:if>
  </ctl:code>
</ctl:test>
Functions

```xml
<function name="example:add">  
  <param name="num1">First Number</param>  
  <param name="num2">Second Number</param>  
  <return>num1 + num2</return>  
  <description>Adds two numbers</description>  
  <code>  
    <xsl:value-of select="$num1 + $num2"/>  
  </code>  
</function>

<function name="example:sqrt">  
  <param name="num"/>  
  <return>the square root of num</return>  
  <description>Calculates a square root</description>  
  <java class="java.lang.Math" method="sqrt"/>  
</function>
```
XHTML Forms

```xml
<ctl:form>
  <p>
    Do you see the Google logo?<br/>
    <input type="submit" name="answer" value="yes"/>
    <input type="submit" name="answer" value="no"/>
  </p>
</ctl:form>
```

```
<values>
  <value key="answer">yes</value>
</values>
```
XHTML Forms - Uploading File

<ctl:variable name="form-values">
  <ctl:form>
    <p>Select an XML file:</p>
    <input name="myupload" type="file" />
    <br />
    <input type="submit" value="OK" />
    <ctl:parse file="myupload" />
  </ctl:form>
</ctl:variable>

<ctl:message>
  <xsl:text>The root element is named <xsl:text></xsl:text></xsl:text>
  <xsl:value-of select="name($form-values/values/value[@key='myupload']/*)" />
</ctl:message>
HTTP Requests

```xml
<request>
  <url>http://www.somewms.com</url>
  <method>get</method>
  <param name="SERVICE">WMS</param>
  <param name="REQUEST">GetCapabilities</param>
  <param name="VeRsIoN">1.1.1</param>
</request>
```
Parsers

```xml
<xsl:variable name="results">
  <request>
    <url>http://www.example.com/example.xml</url>
    <method>get</method>
    <parsers:XMLValidatingParser>
      <parsers: schemas>
        <parsers:schema type="url">
          http://www.example.com/example.xsd
        </parsers:schema>
      </parsers: schemas>
    </parsers:XMLValidatingParser>
  </request>
</xsl:variable>
<xsl:if test="not($results/*)">
  <message>Parsing or validation failed.</message>
</xsl:if>
```

CDataParser
HTTPParser
XMLValidatingParser
SOAPParser
XSLTransformationParser
Example - SOS DescribeSensor request

```xml
<xsl:variable name="noOutputFormatRequest">
  <ctl:request>
    <ctl:url>
      <xsl:value-of select="$postURL"/>
    </ctl:url>
    <ctl:method>post</ctl:method>
    <ctl:body>
      <DescribeSensor
        service="SOS"
        xmlns="http://www.opengis.net/sos/1.0"
      >
        <xsl:attribute name="version">
          <xsl:value-of select="$describeSensorVersion"/>
        </xsl:attribute>
        <procedure>
          <xsl:value-of select="$procedure"/>
        </procedure>
      </DescribeSensor>
    </ctl:body>
  </ctl:request>
  <ctl:call-function name="sosFunctions:xmlValidatingParser">
    <ctl:with-param name="schemaFile" select="$exceptionReportSchema"/>
  </ctl:call-function>
</xsl:variable>
```
Demonstration WFS testing
Agenda

- Resources and Getting Started
CITE WIKI

http://cite.opengeospatial.org/

Compliance and Interoperability Testing Initiative (CITE)

CITE Navigation
- About CITE
- Start Testing
- Beta Testing
- TEAM Engine Quick Start
- Build Instructions
- Standards Available for Testing
- Reference Implementations
- Developer Information
- Frequently Asked Questions
- Discussion / Issues
- Service Status
- Contact Us
- Recent posts
- Tests Available in Future

Compliance & Interoperability Testing & Evaluation Initiative

Submitted by webmaster on Tue, 2007-04-03 18:06

Compliance & Interoperability Testing & Evaluation (CITE), also known as the OGC Compliance Testing Program, is an ongoing initiative that develops tests for OGC standards, and makes those tests available for online testing. The goal of CITE is to increase systems interoperability while reducing technology risks by providing a process whereby compliance for OGC specifications can be tested.

The Compliance Testing Program provides confidence to technology vendors and buyers. Vendors feel confident that they are providing a product compliant with OGC standards, which will be easier to integrate and easier to market. Buyers feel confident that a compliant product will work with another compliant product based on the same OGC specification, regardless of which
Building instructions

http://cite.opengeospatial.org/node/65

Build instructions for performing local test

Build prerequisites

- JAVA JDK 1.6 or greater. You can download JAVA from here.
- Have Tomcat Installed.
- Be able to run Apache Ant builds. More information about ANT here. Most of the IDE tools like Eclipse and Netbeans already have built in this capability.

Get the Source Code

Checkout the TEAM Engine code from Sourceforge

You should have a layout similar to the figure bellow.
The cite-forum public mailing list provides CITE and OGC compliant software developers the means to discuss issues and solutions related to OGC tests. This list is used for:

- Discussions of problems found when using TEAM Engine or the test scripts (for example apparently inexplicable failing tests)
- Discussions of new features for TEAM Engine
- Discussions of new tests enhancements
- Announcing of new beta and production releases of TEAM Engine
- Submission of bugs when using TEAM Engine or test scripts
CITE Subcommittee Mailing List

https://lists.opengeospatial.org/mailman/listinfo/cite

cite.sc@lists.opengeospatial.org

This list is used for:

- Discussion of Policies
- Discussion of Road map
- Discussions of Strategies
- Discussion of the Agenda for the TC meetings
Issue Tracker

- Is in the CITE SC Project Web Site.
• Running TEAM Engine deploying a TEAM Engine WAR
1) Checkout team engine from Sourceforge

2) Copy the tests in the scripts folder

- Tests are available in one zip file (~ 10 MB): http://portal.opengeospatial.org/files/?artifact_id=44173
3) Create a build.properties file

```
build.properties

usersdir /local/srv/teamengine2.dat/te2.users
workdir /local/srv/teamengine2.dat/te2.work
tomcat.dir /Applications/apache-tomcat-5.5.28
```
4) Run the build
5) Get Successful Build Result

```
zip.manager.bin:
[delete] Deleting directory /Users/bermud/Documents/workspace/teamengine-tr
[mkdir] Created dir: /Users/bermud/Documents/workspace/teamengine-tr
[mkdir] Created dir: /Users/bermud/Documents/workspace/teamengine-tr
[copy] Copying 2 files to /Users/bermud/Documents/workspace/teamengine-tr

BUILD SUCCESSFUL
Total time: 9 seconds
```
After the build

One WAR is created and 2 jars. Need to be moved to a servlet container. For example TOMCAT
Tomcat directories for the build artifacts
After deploying the WAR file you should see:

Welcome

The Test, Evaluation, And Measurement (TEAM) Engine is a test script interpreter. It executes test scripts written using Compliance Test Language (CTL) to verify that an implementation of a specification complies with the specification.

Start Testing

Problems? Email the webmaster
Agenda

• Running TEAM Engine via command line.
Build Engine
Create a test suite

- Go to the project root
  ```
  cd /Users/bermud/Documents/workspace/prod
  ```
- Create a directory to keep information about the sessions
  ```
  mkdir sessions
  ```
- Run
  ```
  ./bin/test.sh -source=scripts/wms-1.3.0-r1/ctl -logdir=sessions
  ```
8) Output from your run

INFO: Validating /Users/bermud/Documents/workspace/prod/apps/engine/resources/com/occamlab/te/scri
Feb 27, 2011 5:39:56 PM com.occamlab.te.Generator generateXsl
INFO: Validating /Users/bermud/Documents/workspace/prod/apps/engine/resources/com/occamlab/te/scri
Feb 27, 2011 5:39:56 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/basic_elements.xml
Feb 27, 2011 5:39:56 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/dimensions.xml
Feb 27, 2011 5:39:57 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/functions.xml
Feb 27, 2011 5:39:57 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/getcapabilities.xml
Feb 27, 2011 5:39:57 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/getfeatureinfo.xml
INFO: Validating scripts/wms-1.3.0-r1/ctl/getmap.xml
Feb 27, 2011 5:39:59 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/interactive.xml
Feb 27, 2011 5:39:59 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/main.xml
Feb 27, 2011 5:39:59 PM com.occamlab.te.Generator generateXsl
INFO: Validating scripts/wms-1.3.0-r1/ctl/recommendations.xml
Feb 27, 2011 5:39:59 PM com.occamlab.te.Generator generateXsl
JAVA window will pop up for WMS 1.3.0

Web Map Service 1.3.0

Capabilities Setup

Enter a capabilities document URL below. main may be the URL to a static capabilities document, or a GetCapabilities request from a WMS. A typical GetCapabilities request will take main form:

http://hostname/path?SERVICE=WMS&REQUEST=GetCapabilities&VERSION=1.3.0

UpdateSequence Values

The WMS spec allows servers to use an UpdateSequence value for maintaining cache consistency as described in Section 7.2.3.5 of the specification. If the server advertises an UpdateSequence value and the Automatic option is selected below, the test suite will attempt to test the UpdateSequence behavior automatically. However, the lexical ordering of UpdateSequence values is determined by the server, so the tests may not always be correct. If you suspect a problem, select the Manual option and enter the updateSequence values requested below.

Automatic - The updateSequence tests will use automatically generated updateSequence values
Manual - The updateSequence tests will use the values supplied below
Setting up TEAM Engine in a Development Environment

• Create scripts to run via command line.
• Integration with MAVEN and JUNIT coming soon. Experimental branch is here:
  https://teamengine.svn.sourceforge.net/svnroot/teamengine/branches/maven
Getting Involved

• Get in the mailing list:
  https://lists.opengeospatial.org/mailman/listinfo/cite-forum
• If you are interested in advancing TEAM Engine let us know
• Any other comments and suggestions:
  Luis Bermudez
  lbermudez@opengeospatial.org