



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

Why Standards for Data Collaboration ?



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

“ 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

OGC[®]

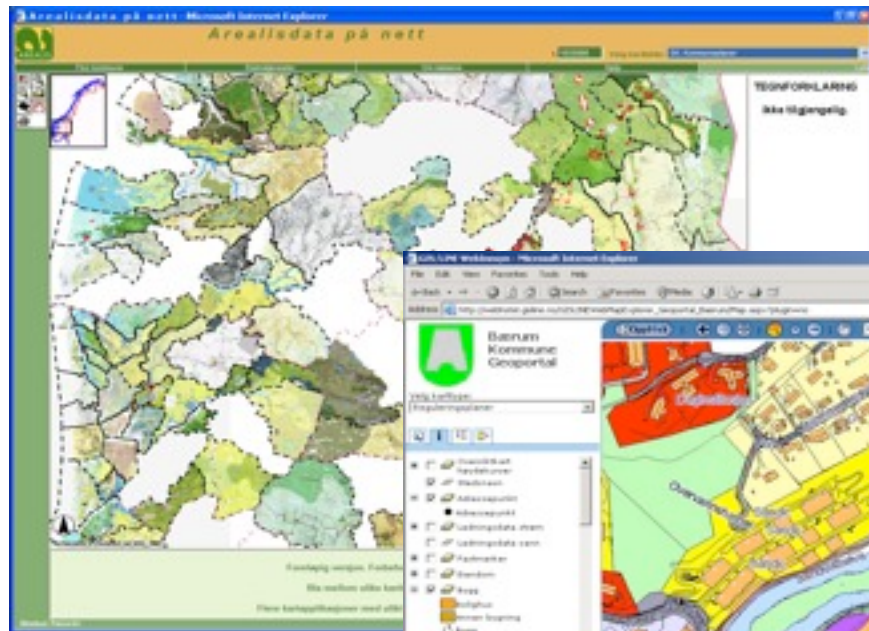


© 2011 Open Geospatial Consortium, Inc.

Digital Norway – Land Use

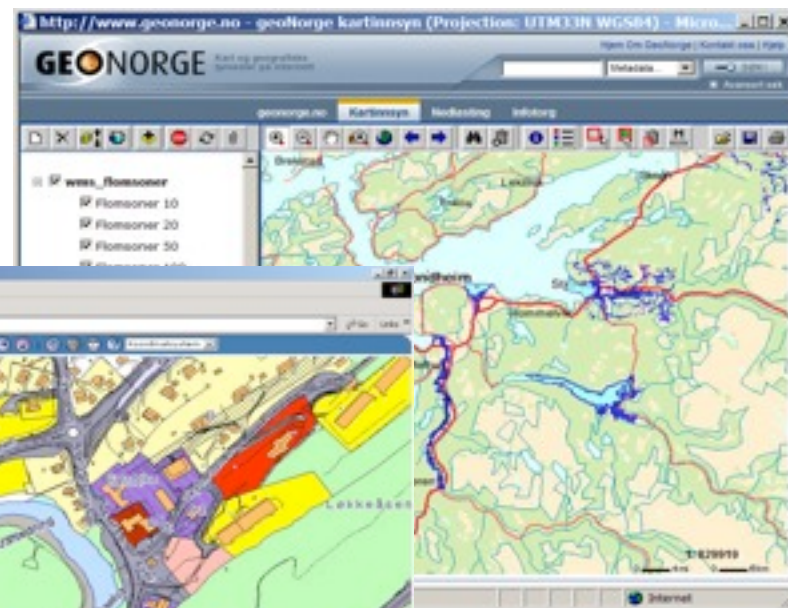


Municipal Areas

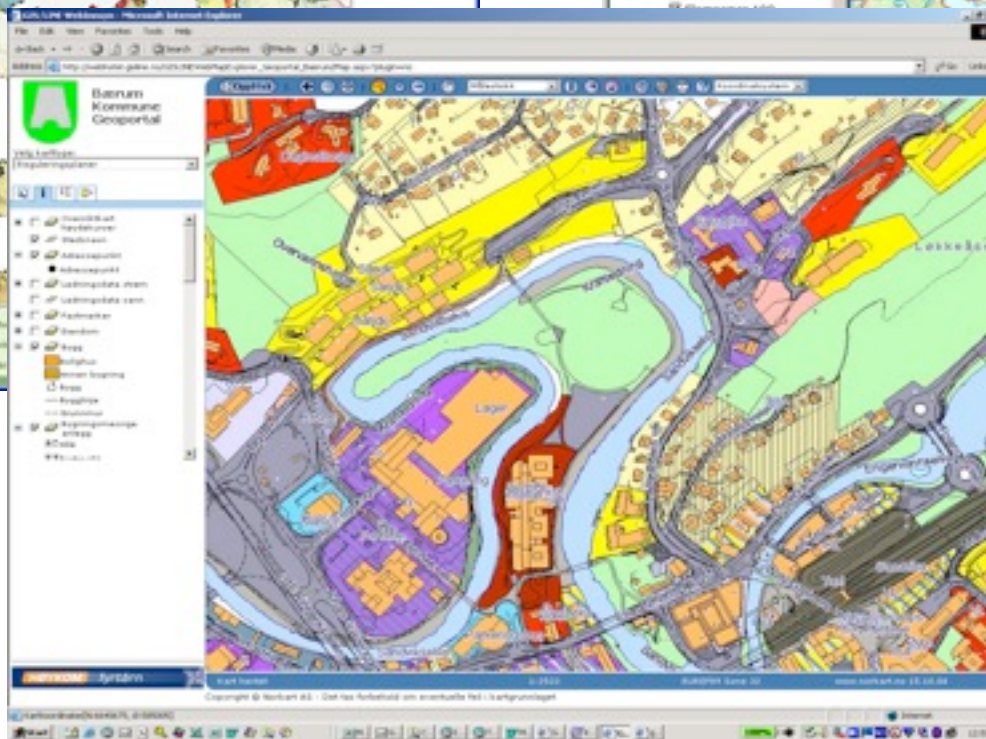


- Fisheries
- Waste Water Outflow
- Water Supply

Flood Risk Areas



- Demography
- Biodiversity
- Agriculture and Forestry



Land Use

OGC®

© 2011 Open Geospatial Consortium, Inc.

Debris Flow Monitoring - Taiwan



Debris Flow Monitoring - Taiwan



OGC[®]

© 2011 Open Geospatial Consortium, Inc.



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.



OGC®

© 2011 Open Geospatial Consortium, Inc.



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



OGC®

© 2011 Open Geospatial Consortium, Inc.



Sensor Web Enablement Standards Application Ocean Observation



NOAA IOOS
INTEGRATED OCEAN OBSERVING SYSTEM

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.



OGC®

© 2011 Open Geospatial Consortium, Inc.

Friday, May 20, 2011

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.

OGC[®]

© 2011 Open Geospatial Consortium, Inc.

9

OGC From an Organizational Perspective



Over 420 Member Organizations



NORTHROP GRUMMAN



ORACLE

Google



BAE SYSTEMS



SAIC
From Science to Solutions



INTERGRAPH



National Geographic Information Institute
Ministry of Land, Transport and Maritime Affairs

Text



SAMSUNG SDS



SEJONG UNIVERSITY



Microsoft



OGC[®]

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



Health



Emergency Services



Consumer Services



Energy



Geosciences



E -Government



OGC®

© 2011 Open Geospatial Consortium, Inc.

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.

OGC Alliance Partners



... and others

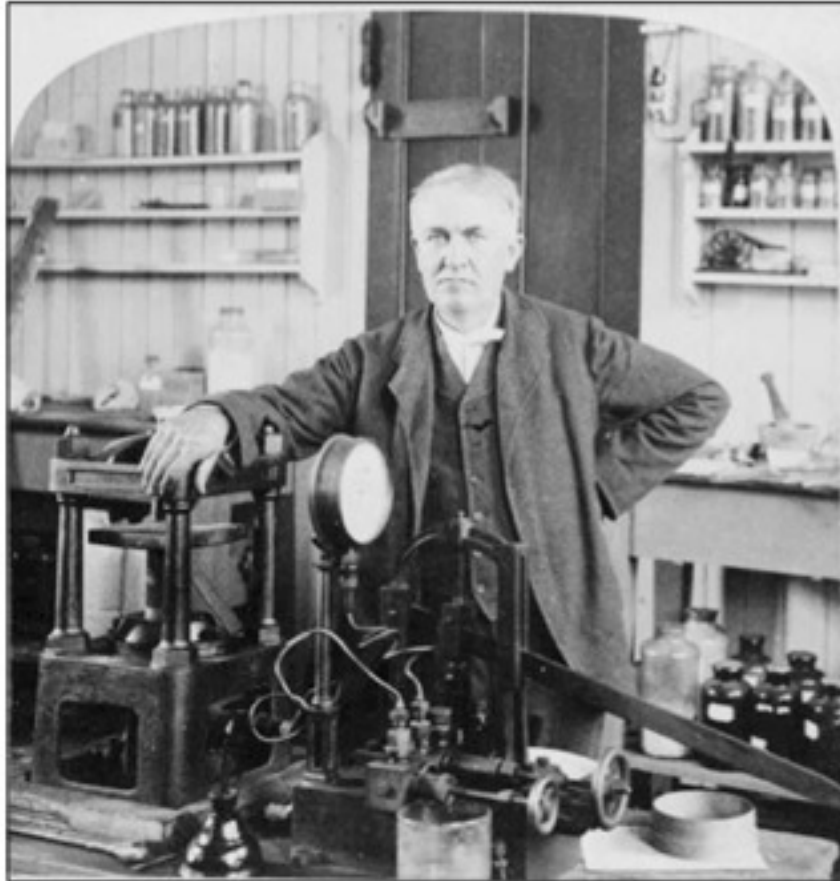
OGC®

www.opengeospatial.org/ogc/alliancepartners

© 2011 Open Geospatial Consortium, Inc.

18

..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

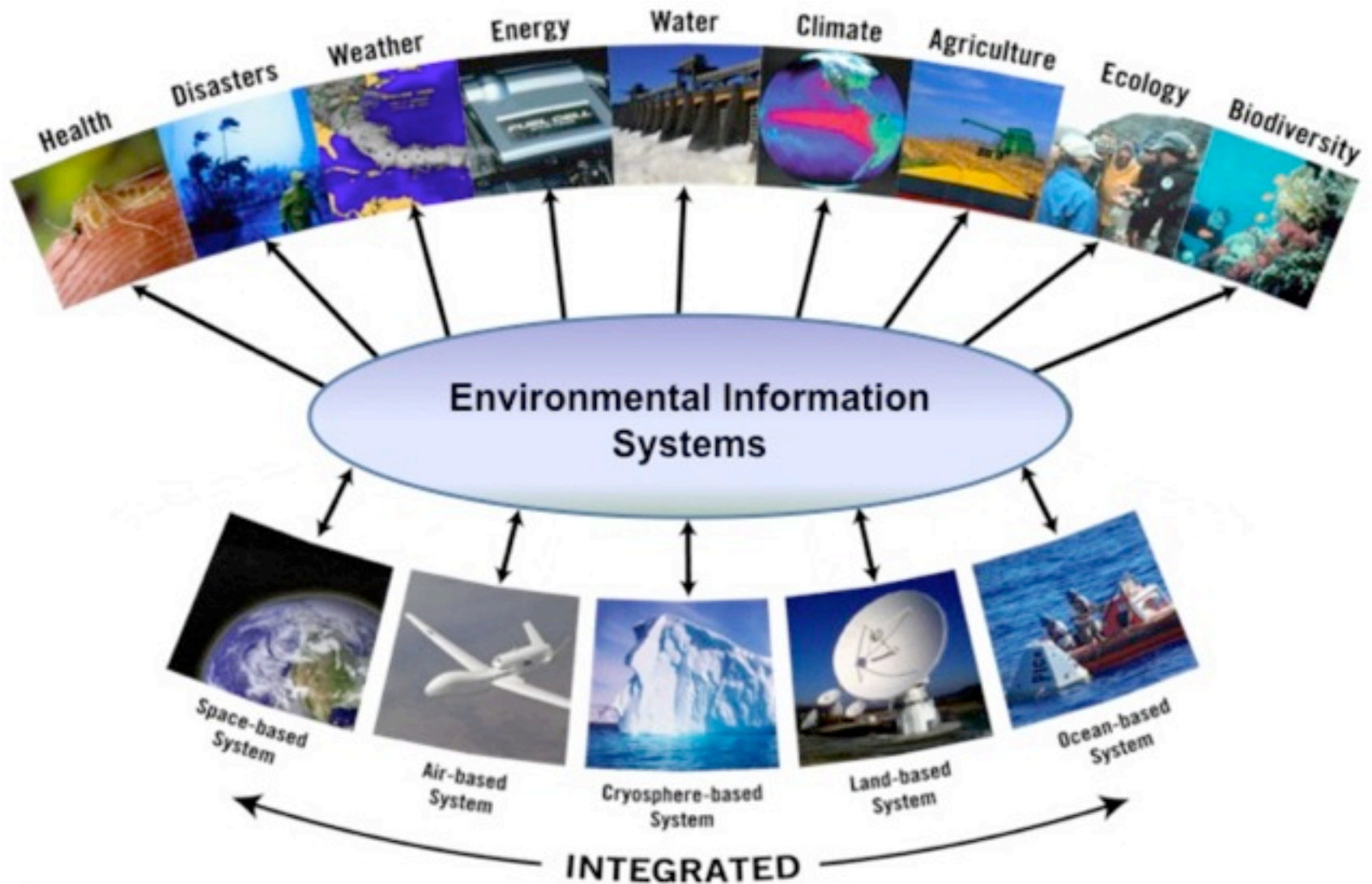


Interoperability Program

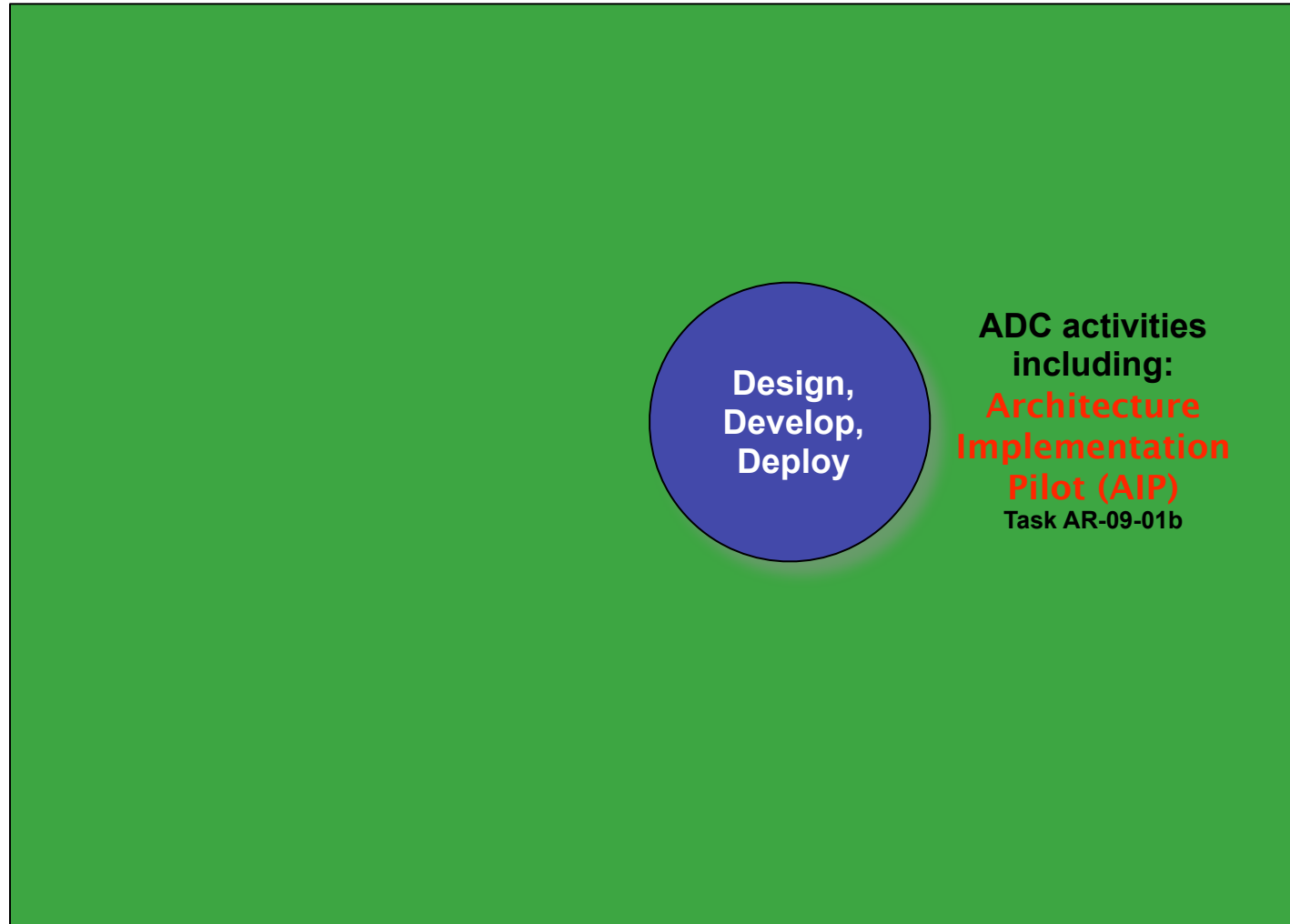


40+ Interoperability Program initiatives since 1999.

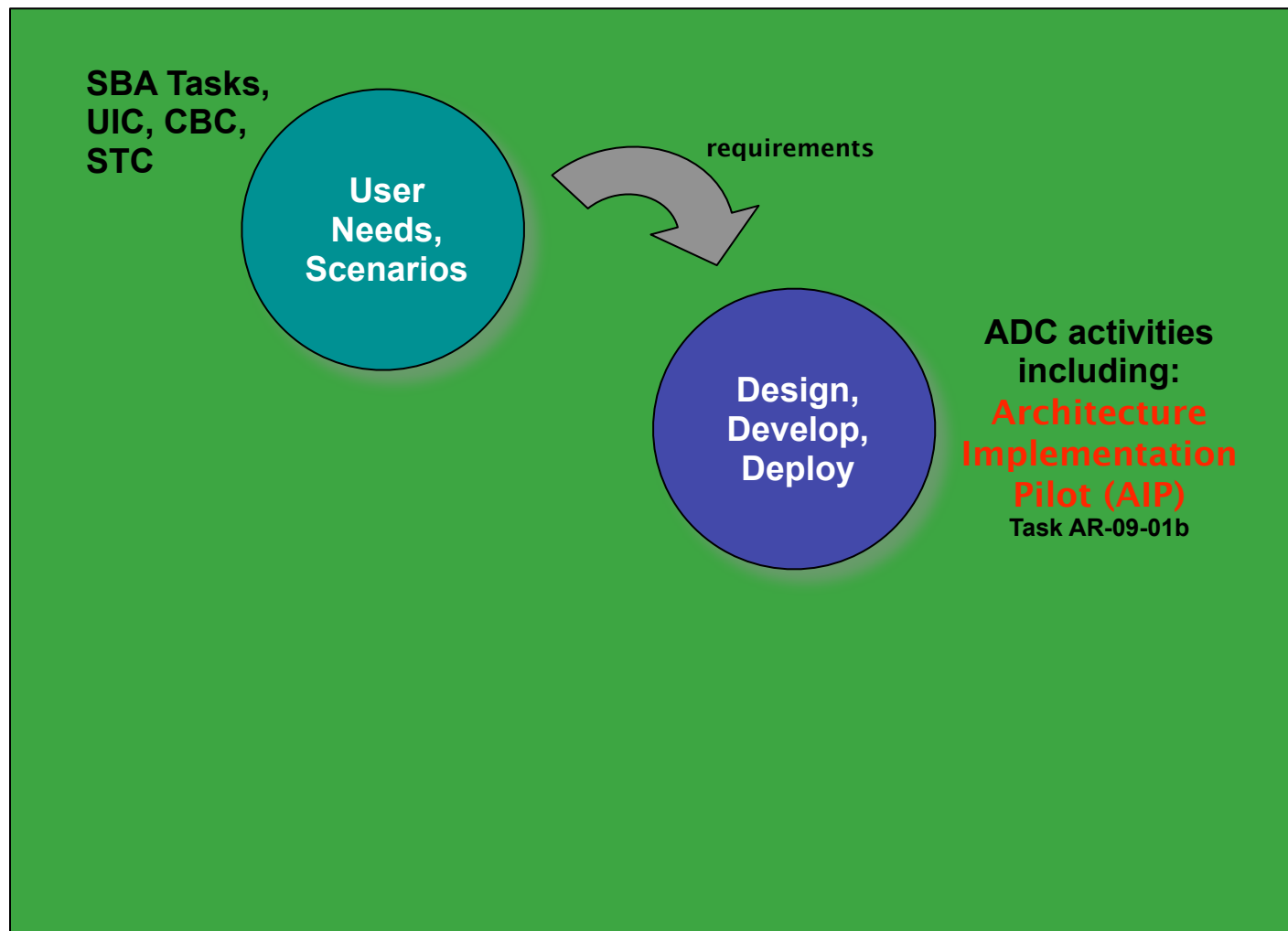
GEOSS



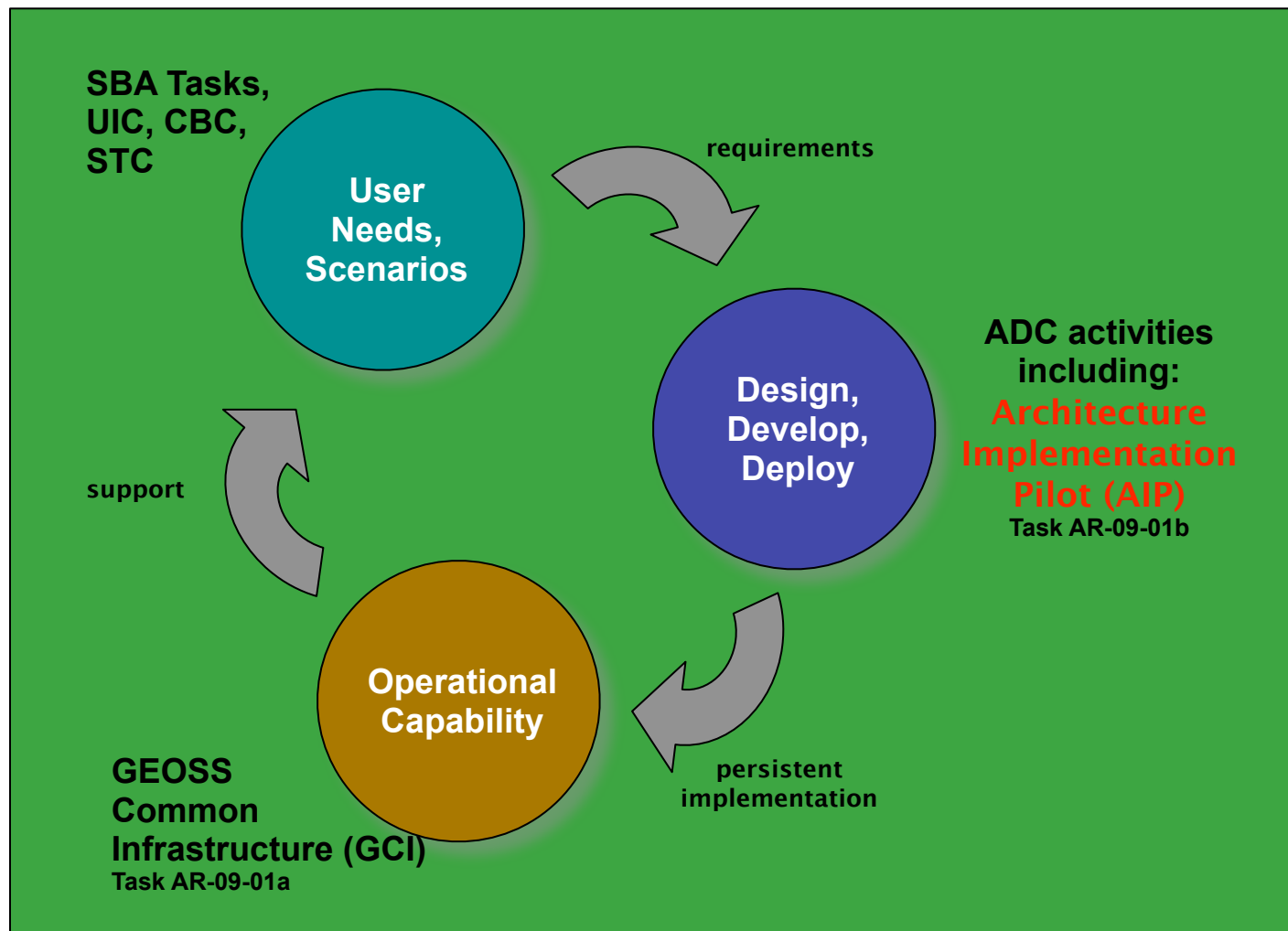
GEOSS Architecture Implementation Pilot



GEOSS Architecture Implementation Pilot



GEOSS Architecture Implementation Pilot



Need for Compliance Testing



**Directions Magazine**
All Things Location

Hello. [Login](#) | [Register](#)

[→SEARCH](#)

**Free ArcGIS App for Windows Phone 7**
[Download Now](#)**ABOUT DIRECTIONS**

Follow Us

Subscribe
Advertise
Publications
Conferences

HOMECHANNELSNEWSRESOURCESBLOG/OPINIONPODCASTS/VIDEOSWEBINARSMAP GALLERY

Map Suite
For .NET GIS Developers[Free Download!](#)**Press Releases**
[Home](#) | [Submit Press Release](#)**SuperGIS Server 3 Passes OGC Compliance Test**
Tuesday, May 17th 2011

SuperGeo Technologies Inc. | Taiwan

Read More About: [GEOSPATIAL DATA](#), [GIS](#), [GIS SOFTWARE](#), [MAP SERVICE](#), [OGC](#), [SUPERGIS APPLICATION](#), [SUPERGIS DESKTOP 3](#), [SUPERGIS SERVER 3](#), [SUPERPAD 3](#), [SUPERWEBGIS 3](#), [WCS](#), [WFS](#), [WMS](#)**Content Engineering Services**
Transformation | Optimization**Infotech**
Creating Business Impact

SuperGeo Technologies, the leading global provider of complete GIS software and solutions, is delighted to announce that SuperGIS Server 3, its comprehensive server-based GIS software, has passed OGC official compliance test.

NEWSLETTER
Get the Daily Newsletter with the latest technology headlines and feature**Don't get caught with your maps DOWN!**
[geoxmf.com](#)**CoreLogic.**
For highly accurate, up-to-date tax information. »

OGC®

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

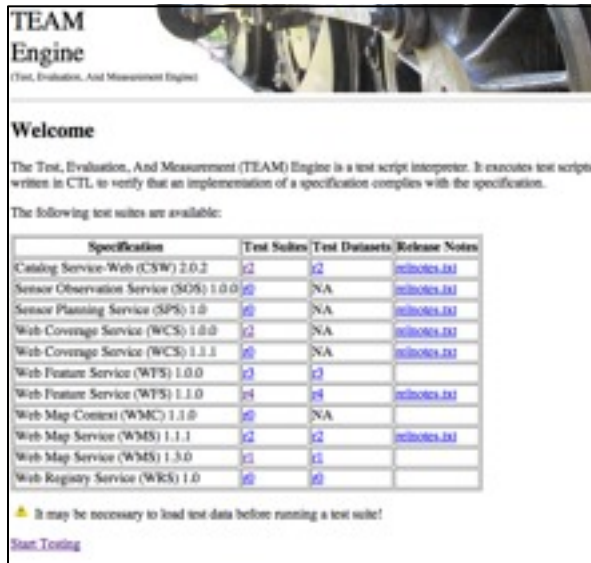
ESRI	
Product Name	OGC Spec
ArcGIS 8.1	
ArcGIS Server 9.3	
ArcGIS Server 9.3.1	
ArcGIS Server 9.2	
lat/ion GmbH	
Product Name	OGC Spec
degree Sensor	
Observation Service	SOS 1.0.0 (compliant)
degree Web	
Coverage Service	WCS 1.0 (compliant)
Roita India Ltd.	
Product Name	OGC Spec
Rolta OnPoint 6.4	WMS 1.3.0 (server compliant), CAT 2.0.2, WFS 1.0.0 (server compliant)
KSIC(Korea Geospatial Information & Communication Co., LTD.)	
Product Name	OGC Spec
IntraMap/Web v5.6	GML 3.0, WCS 1.0.0, WFS 1.1.0, SLD 1.0, WMS 1.1.1, WMS 1.3.0 (compliant)
IntraMap/Web 6.0	WMS 1.3.0 (compliant)
Oracle Corporation	
Product Name	OGC Spec
Oracle Application Server MapViewer, 10g Release 2 (10.1.2)	WMS 1.1.1 (server compliant)
Oracle Locator 11g, Release 11.1.0.7	SFS(TF) 1.1 (compliant)

Compliance Procedure

1) Developers go to online test engine

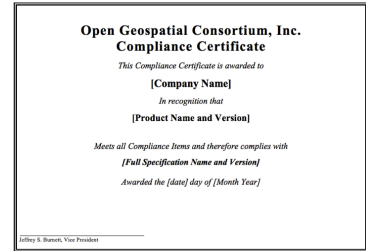
2) Fill the Test Results (TSR) form

4) Get compliance certificate



3) Pay License Fee

Annual Fee per Product Version per Implemented Standard Version		
Licensee Total Gross Annual Revenue	Non Member	Member
\$0M - < \$2M	\$100	\$80
\$2M - < \$3M	\$250	\$200
\$3M - < \$10M	\$500	\$400
\$10M - < \$20M	\$750	\$600
\$20M - < \$50M	\$1,200	\$960
\$50M - < \$100M	\$2,000	\$1,600
\$100M - < \$500M	\$4,500	\$3,600
\$500M+	\$7,000	\$5,600



5) Use certification mark



More information:
<http://bit.ly/gTmmSo>

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

OGC[®]

Agenda



- **Overview OGC Testing Facility - TEAM Engine**

Online Facility TEAM Engine

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

TEAM Engine

(Test, Evaluation, And Measurement Engine)

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:

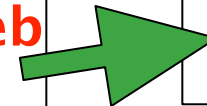
Specification	Test Suites	Test Datasets	Release Notes
Catalog Service-Web (CSW) 2.0.2	r2	r2	relnotes.txt
Sensor Observation Service (SOS) 1.0.0	r0	NA	relnotes.txt
Sensor Planning Service (SPS) 1.0	r0	NA	relnotes.txt
Web Coverage Service (WCS) 1.0.0	r2	NA	relnotes.txt
Web Coverage Service (WCS) 1.1.1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

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

[Start Testing](#)

MOZILLA PUBLIC
LICENSE
Version 1.1

Java - Web
Server



- ▼ apache-tomcat-5.5.28
 - ▶ bin
 - ▶ common
 - ▶ conf
 - LICENSE
 - ▶ logs
 - NOTICE
 - RELEASE-NOTES
 - RUNNING.txt
 - ▶ server
 - ▶ shared
 - ▶ temp
 - ▼ webapps
 - ▶ balancer
 - ▶ jsp-examples
 - ▶ ROOT
 - ▶ servlets-examples
 - ▶ teamengine
 - teamengine.war

TEAM Engine

(Test, Evaluation, And Measurement Engine)



Select test suite:

Organization	Standard	Version	Test Suite Rev
OGC 	WMS 	1.3.0 	r1 

Select Profile(s):

Enter Session Description (Optional):

Test 1 - Feb 01 2011 - Data one

Start a new test session

Tester
selects test
suite

Tester
names
session

OGC Production release. Built 2010-11-16. Problems? Email the [webmaster](#).

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`

Capabilities URL

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

☒ Automatic - The updateSequence tests will use the values supplied below
☐ Manual - The updateSequence tests will use the values supplied below

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

A value that is less than or equal to the updateSequence value
 A value that is less than or equal to the updateSequence value

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 raster dataset.
- ☐ VECTOR ELEVATION - Test the elevation dimension using the cite:Lakes vector dataset.
- ☐ TIME - Test the time dimension using the cite:Autos dataset.
- ☐ RECOMENDATIONS - Test functionality which is recommended in the specification.

Console

```
Testing suite wms-1.3.0:compliance_suite...
Testing main:main (s0001)...
Assertion: The implementation under test complies with the WMS 1.3.0 specification.
Testing interactive:main (s0001/d3607e344_1)...
Assertion: The tests that require user interaction behave properly.
Testing interactive:basic-polygons-sanity-check (s0001/d3607e344_1/d3402e17_1)...
Assertion: The diamond from the cite:BasicPolygons layer displays correctly.
Test interactive:basic-polygons-sanity-check Failed
Testing interactive:blue-lake-sanity-check (s0001/d3607e344_1/d3402e19_1)...
Assertion: The layers from the Blue Lake dataset display correctly.
Test interactive:blue-lake-sanity-check Failed
Testing interactive:layer-order (s0001/d3607e344_1/d3402e21_1)...
Assertion: When a GetMap request contains multiple layers, then the response render
Test interactive:layer-order Failed
Testing interactive:aspect-ratio (s0001/d3607e344_1/d3402e23_1)...
Assertion: When a GetMap request is made where the aspect ratio of the BBOX and the
```

Tester provides end point of the service

Tester Selects Options

Console provides feedback on test assertions

Results for session s0006

Test Suite: Web Map Service (WMS) 1.3.0



Summary
of
Results

- ☐ ☒ [Test main:main \(View Details\)](#): Passed
 - ☐ ☒ [Test interactive:main \(View Details\)](#): Passed
 - ☒ [Test interactive:basic-polygons-sanity-check \(View Details\)](#): Passed
 - ☒ [Test interactive:blue-lake-sanity-check \(View Details\)](#): Passed
 - ☒ [Test interactive:layer-order \(View Details\)](#): Passed
 - ☒ [Test interactive:aspect-ratio \(View Details\)](#): Passed
 - ☒ [Test interactive:exceptions-inimage \(View Details\)](#): Passed
 - ☒ [Test interactive:fees-and-access-constraints \(View Details\)](#): Passed
 - ☐ ☒ [Test main:options-requirements \(View Details\)](#): Passed
 - ☒ [Test main:gif-or-png \(View Details\)](#): Passed
 - ☒ [Test main:std-data-present \(View Details\)](#): Passed
 - ☒ [Test main:getfeatureinfo-supported \(View Details\)](#): Passed
 - ☒ [Test main:std-data-queryable \(View Details\)](#): Passed
 - ☐ ☒ [Test basic_elements:main \(View Details\)](#): Passed
 - ☐ ☒ [Test basic_elements:version-negotiation \(View Details\)](#): Passed
 - ☒ [Test basic_elements:negotiate-no-version \(View Details\)](#): Passed
 - ☒ [Test basic_elements:negotiate-basic_elements-version \(View Details\)](#): Passed
 - ☒ [Test basic_elements:negotiate-higher-version \(View Details\)](#): Passed
 - ☒ [Test basic_elements:negotiate-lower-version \(View Details\)](#): Passed
 - ☐ ☒ [Test basic_elements:reserved-chars \(View Details\)](#): Passed
 - ☒ [Test basic_elements:escaped-chars \(View Details\)](#): Passed



Online Facility TEAM Engine

[tion:core-SOS.GetObservation-ResponseAdvertisedEventData.1 \(View Details\)](#): Passed
[tion:core-SOS.GetObservation-ResponseMatchingEventData.1 \(View Details\)](#): Passed
[tion:core-SOS.GetObservation-ResponseMatchingFeatureOfInterestData.1 \(View Details\)](#): Passed
[tion:core-SOS.GetObservation-ResponseMatchingResultData.1 \(View Details\)](#): Passed
[tion:core-SOS.GetObservation-ResponseMatchingResponseFormatData.1 \(View Details\)](#): Passed

2  Warning: 0  Fail: 0

[Delete this session](#) [Download log Files](#) [Create execution log report file](#) [Email log Files](#)



**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

Open Geospatial Consortium Inc.

Date: 2010-11-23

Reference number of this OGC® document: OGC 06-126r4

Version: 0.8

Category: OGC® Best Practices Document

Editor: Chuck Morris

Compliance Test Language (CTL) Best Practice





CITE Navigation

- About CITE
- Start Testing
- Beta Testing
- Build for Local Testing
- TEAM Engine & CTL Quick Start
- ▶ Standards Available for Testing
- Reference Implementations
- Developer Information
- Frequently Asked Questions

Writing a simple Hello World Test Script

The following is a simple Hello World CTL test script.

hello.ctl

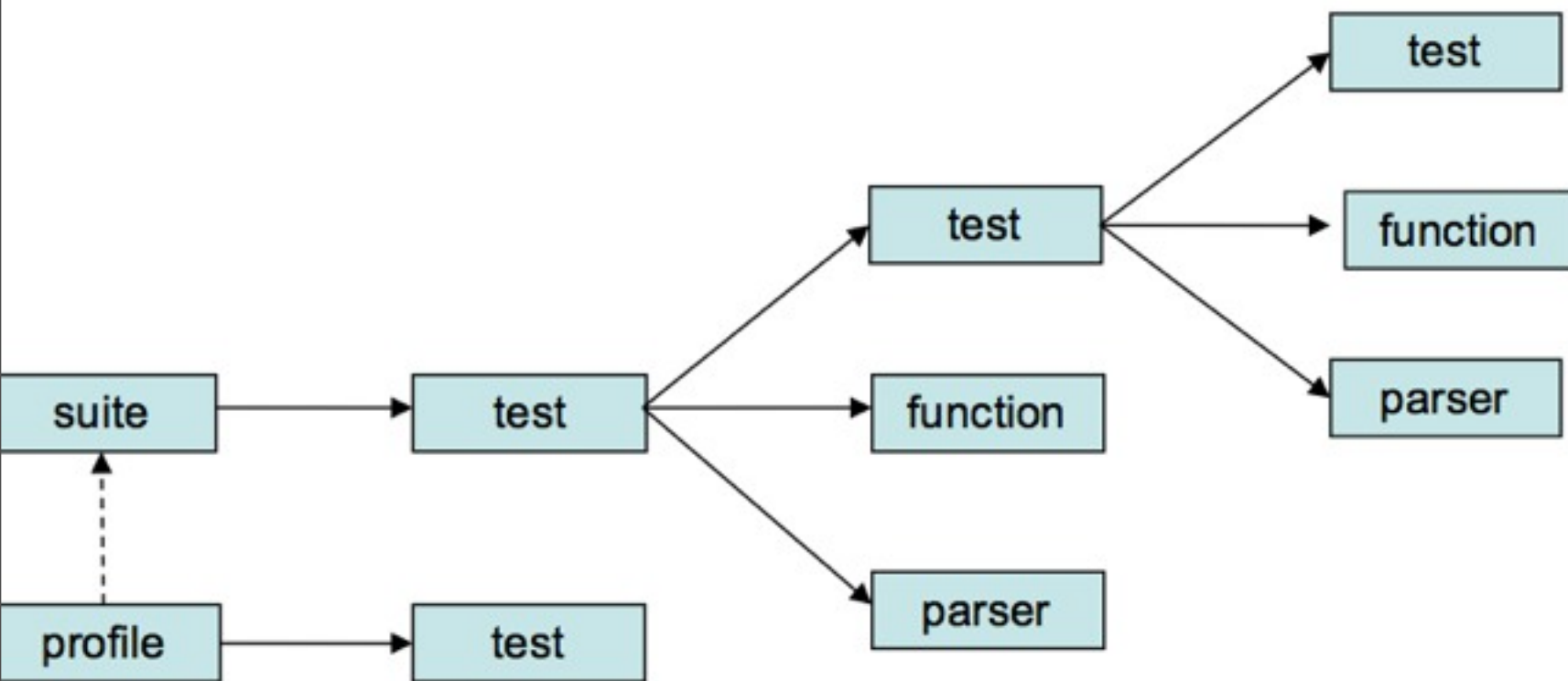
```
<ctl:test
  xmlns:ctl="http://www.occamlab.com/ctl"
  xmlns:my="urn://mynamespace"
  name="my:hello"
>
  <ctl:assertion>This test displays a greeting.</ctl:assertion>
  <ctl:code>
    <ctl:message>Hello, world!</ctl:message>
  </ctl:code>
</ctl:test>
```

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



CTL Features and Examples



```

<ctl:suite name="test:base">
  <ctl:title>Base profile test suite</ctl:title>
  <ctl:description></ctl:description>
  <ctl:starting-test>test:base_main</ctl:starting-test>
  <ctl:form>
    Enter a value for X:
    <xhtml:input type="text" name="x"/>
    <xhtml:br/>
    <xhtml:input type="submit"/>
  </ctl:form>
</ctl:suite>

```

Suite and Profiles

```

<ctl:profile name="test:profile_a">
  <ctl:title>Profile A</ctl:title>
  <ctl:description></ctl:description>
  <ctl:base>test:base</ctl:base>
  <ctl:exclude>/test:base_main/test:fail</ctl:exclude>
  <ctl:starting-test>test:profile_a_main</ctl:starting-test>
</ctl:profile>

```

```

<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: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: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>

```

Suite and Profiles

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

```
<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

```
<ctl:form>
  <p>
    <br/>
    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:value-of select="name($form-values/values/value[@key='myupload'])" />
</ctl:message>
```

HTTP Requests

```
<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

```
<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

```
<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:call-function name="sosFunctions:xmlValidatingParser">
      <ctl:with-param name="schemaFile" select="$exceptionReportSchema"/>
    </ctl:call-function>
  </ctl:request>
</xsl:variable>
```

Demonstration WFS testing



Agenda



- **Resources and Getting Started**

CITE WIKI

<http://cite.opengeospatial.org/>



[OGC® Home](#) | [OGC Network™](#) | [OGC Forum](#)

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



Search

Search this site:

Search

TEAM Engine - CSW, WFS, & WMS compliance testing along with WMC validation. GeoRSS Validator - Validate your GeoRSS feed. GML 2.1.2 Validator - Validate your GML schema or instance documents.


Visit the [OGC Website](#) for a listing of certified OGC® Compliant products.



Building instructions



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



Open Geospatial Consortium, Inc.

OGC® Home | OGC Network™ | OGC Forum

Compliance and Interoperability Testing Initiative (CITE)

Search

Search this site:

CITE Navigation

- About CITE
- Start Testing
- Beta Testing
- Build for Local Testing
- TEAM Engine & CTL Quick Start
- ▶ Standards Available for Testing
- Reference Implementations
- Developer Information
- Frequently Asked Questions
- Discussion / Issues
- Service Status

Build Instructions

Page *Build Instructions* has been updated.

Submitted by Luis Bermudez on Mon, 2010-12-13 20:28

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
<https://teamengine.svn.sourceforge.net/svnroot/teamengine/trunk>.

You should have a layout similar to the figure bellow.

Developers Mailing List



<https://lists.opengeospatial.org/mailman/listinfo/cite-forum>
cite-forum@lists.opengeospatial.org

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.

Agenda

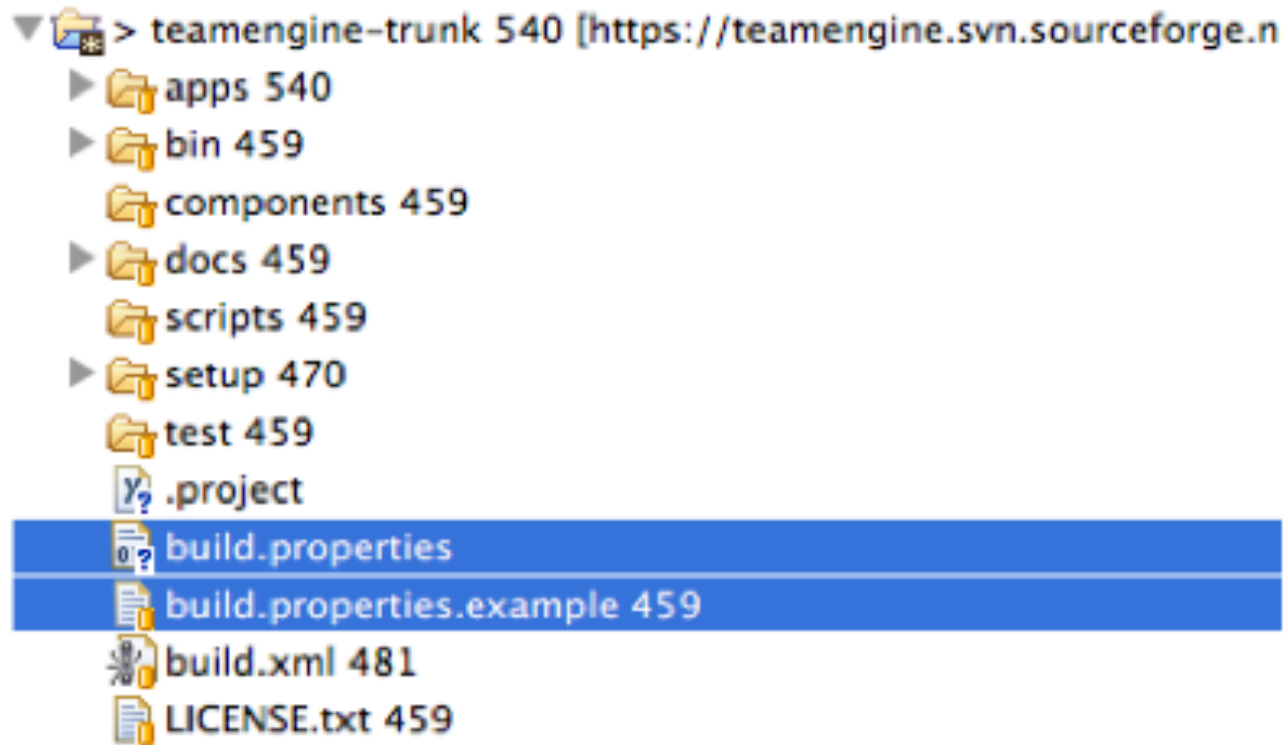


- **Running TEAM Engine deploying a TEAM Engine WAR**

1) Checkout team engine from Sourceforge



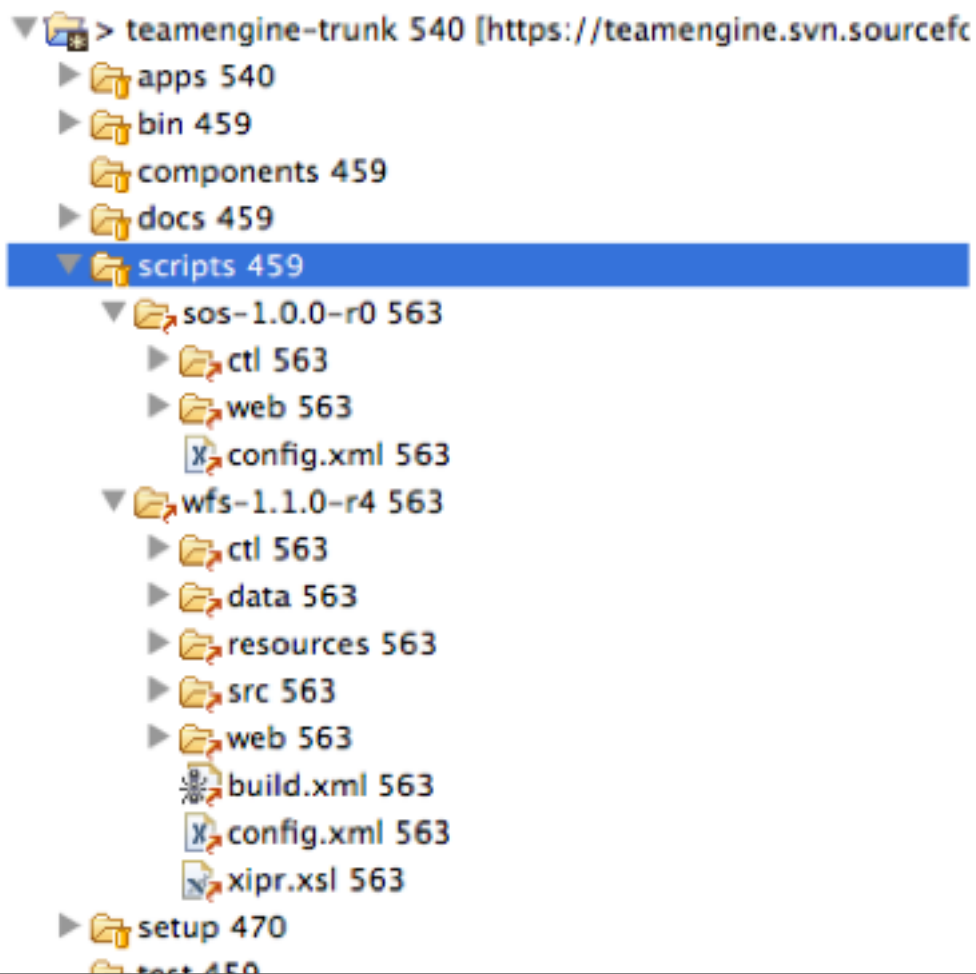
<https://teamengine.svn.sourceforge.net/svnroot/teamengine/trunk>.



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



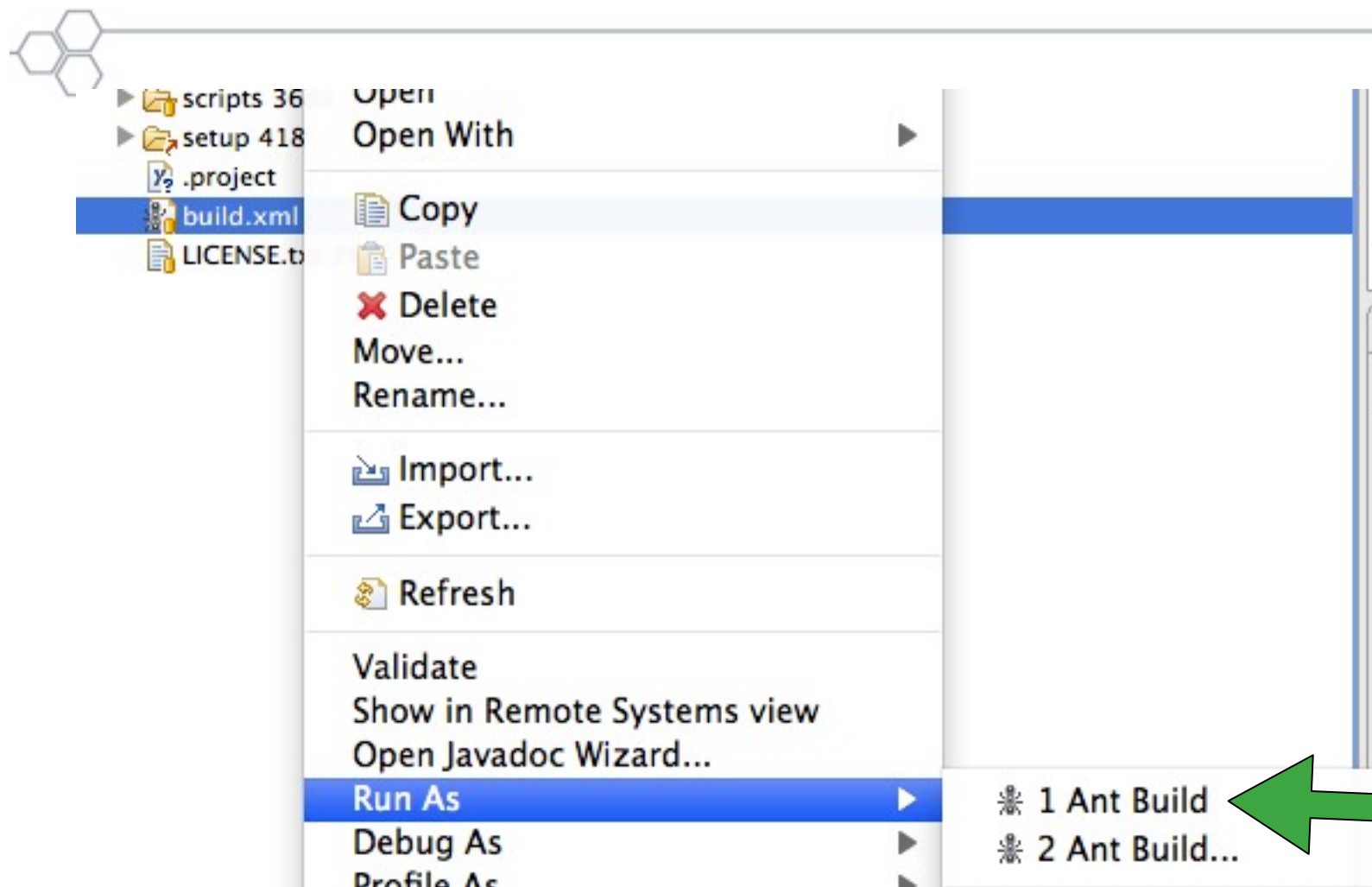
3) Create a build.properties file



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
```

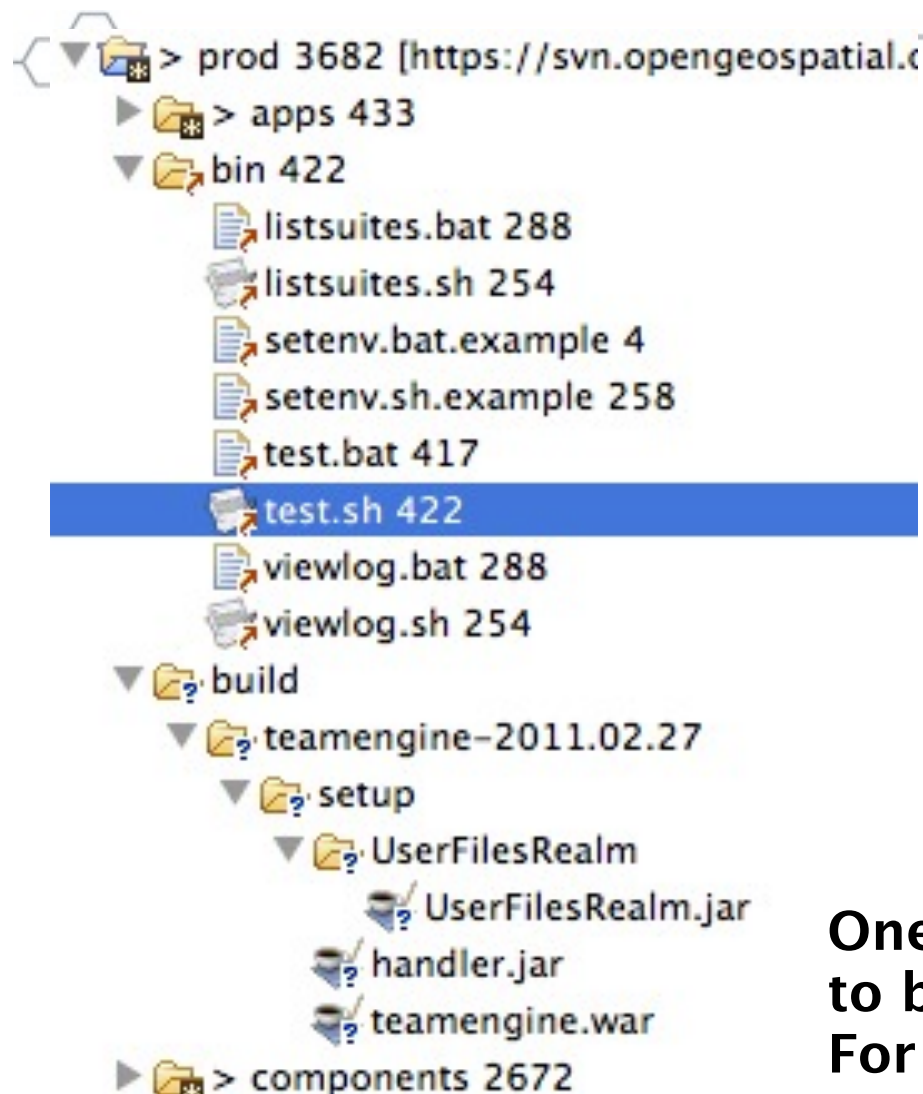
```
[copy] Copying 2 files to /Users/bermud/Documents/workspace/teamengi
```

```
[zip] Building zip: /Users/bermud/Documents/workspace/teamengine-tr
```

```
BUILD SUCCESSFUL
```

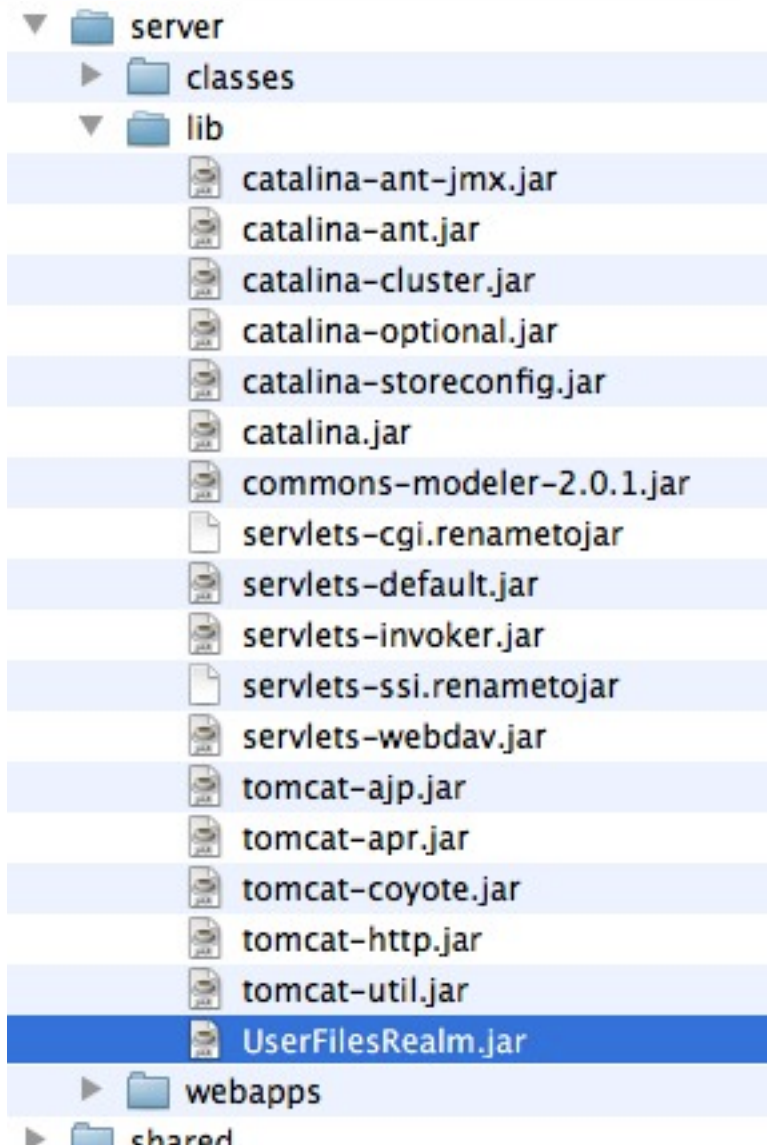
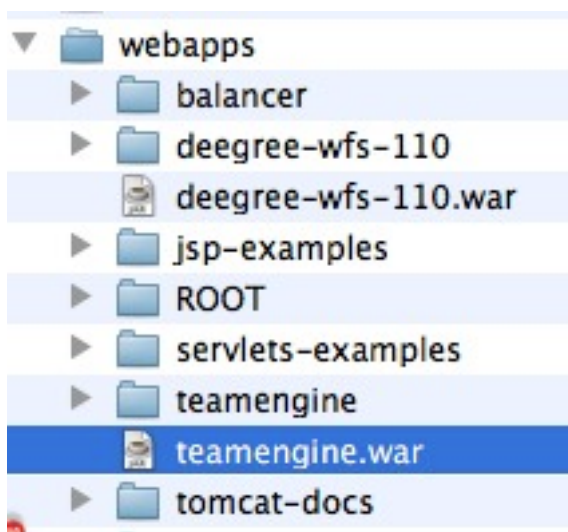
```
Total time: 9 seconds
```

After the build

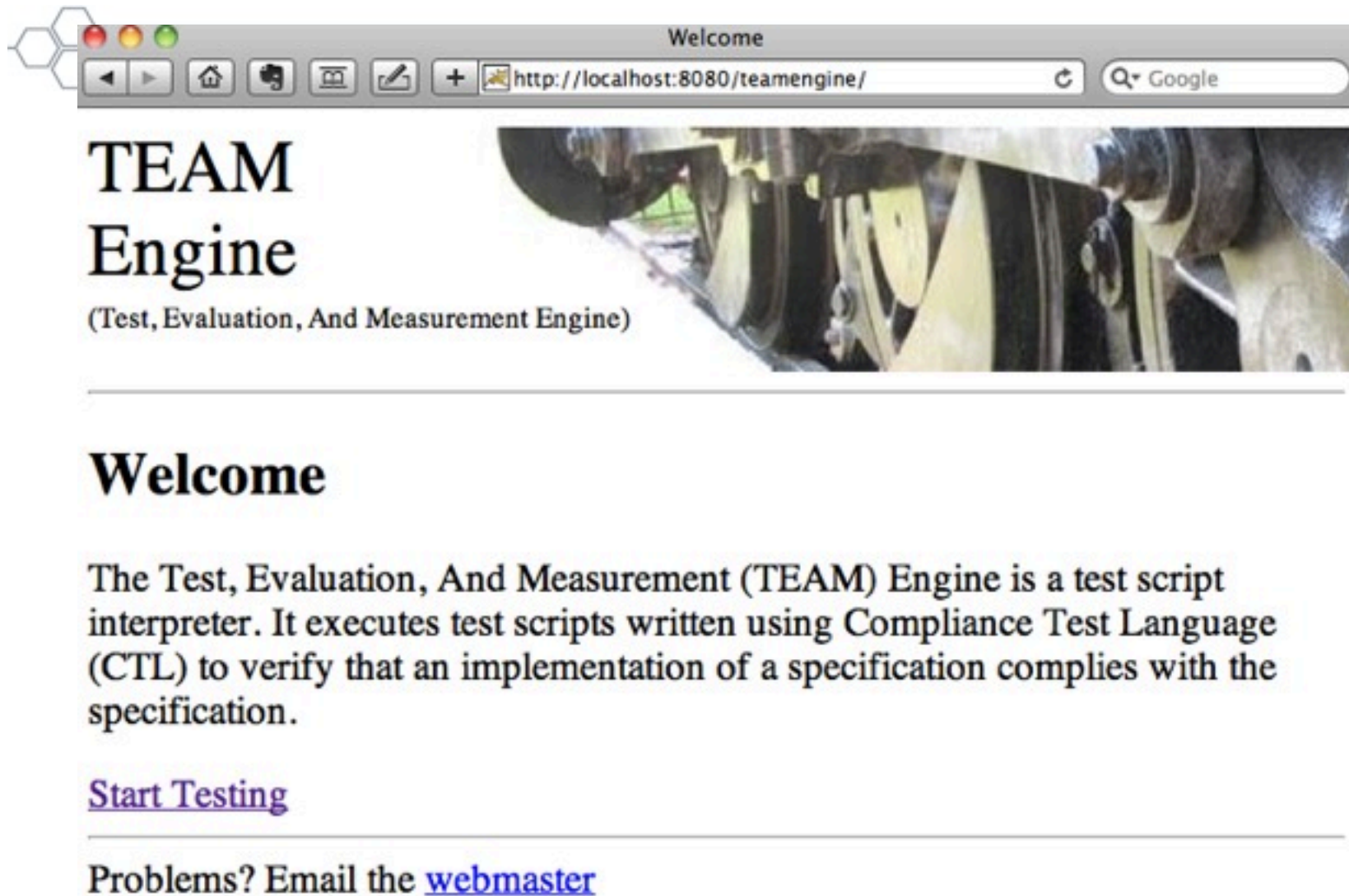


One WAR is created and 2 jars. Need to be moved to a servlet container. For example TOMCAT

Tomcat directories for the build artifacts



After deploying the WAR file you should see:



Agenda



- **Running TEAM Engine via command line.**

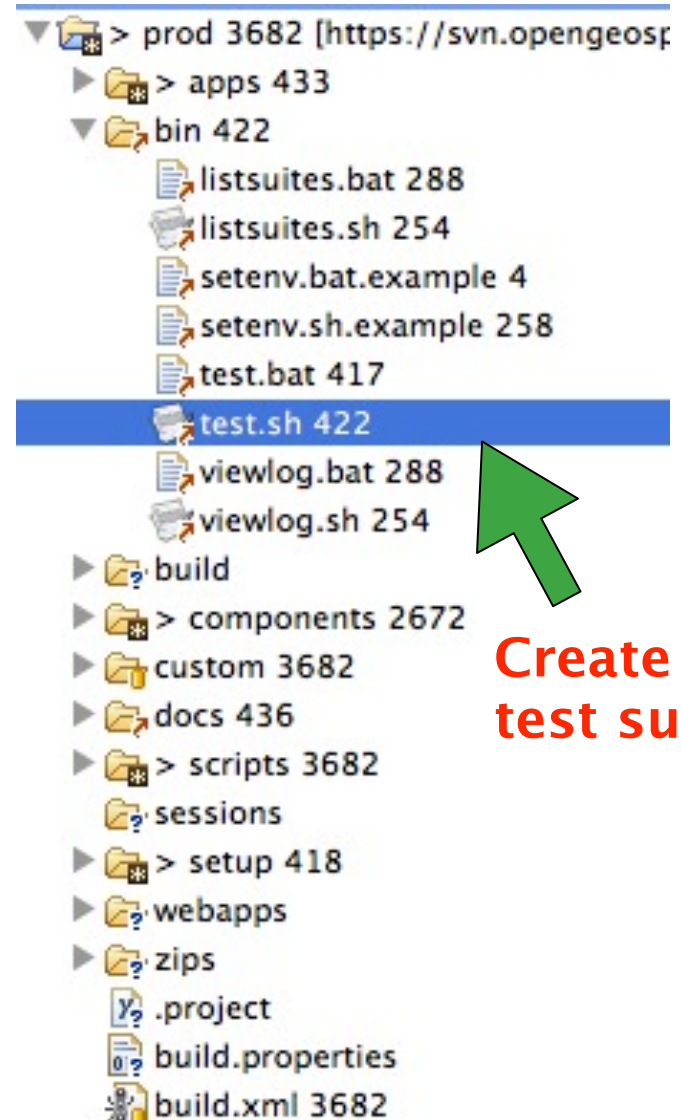
Build Engine



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
Feb 27, 2011 5:39:58 PM com.occamlab.te.Generator generateXsl
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
■
```

JAVA window will pop up for WMS 1.3.0



TEAM Engine

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`

Capabilities URL

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