Warning

This document is not an OGC Standard. This document presents a discussion of technology issues considered in an initiative of the OGC Interoperability Program. This document does not represent an official position of the OGC. It is subject to change without notice and may not be referred to as an OGC Standard. However, the discussions in this document could very well lead to the definition of an OGC Standard.
Abstract

This Engineering Report was created as a deliverable of the CITE Thread as part of the OGC Web Services, Phase 9 (OWS-9) initiative of the OGC Interoperability Program. This OGC® Engineering Report describes how to execute the CITE tests, how to select the conformance classes, how to access and download the TEAM Engine, Test Scripts and Reference Implementations, how to insert your own data and/or schemas along with a section that provides other community help resources.

This is not a normative document.

Keywords

ogcdoc, OGC document, cite, testing, TEAM, ows9, ows-9

What is OGC Web Services 9 (OWS-9)?

OWS-9 builds on the outcomes of prior OGC interoperability initiatives and is organized around the following threads:

- **Aviation**: Develop and demonstrate the use of the Aeronautical Information Exchange Model (AIXM) and the Weather Exchange Model (WXXM) in an OGC Web Services environment, focusing on support for several Single European Sky ATM Research (SESAR) project requirements as well as FAA (US Federal Aviation Administration) Aeronautical Information Management (AIM) and Aircraft Access to SWIM (System Wide Information Management) (AAtS) requirements.

- **Cross-Community Interoperability (CCI)**: Build on the CCI work accomplished in OWS–8 by increasing interoperability within communities sharing geospatial data, focusing on semantic mediation, query results delivery, data provenance and quality and Single Point of Entry Global Gazetteer.


- **OWS Innovations**: Explore topics that represent either new areas of work for the Consortium (such as GPS and Mobile Applications), a desire for new approaches to existing technologies to solve new challenges (such as the OGC Web Coverage Service (WCS) work), or some combination of the two.
- **Compliance & Interoperability Testing & Evaluation (CITE):** Develop a suite of compliance test scripts for testing and validation of products with interfaces implementing the following OGC standards: Web Map Service (WMS) 1.3 Interface Standard, Web Feature Service (WFS) 2.0 Interface Standard, Geography Markup Language (GML) 3.2.1 Encoding Standard, OWS Context 1.0 (candidate encoding standard), Sensor Web Enablement (SWE) standards, Web Coverage Service for Earth Observation (WCS-EO) 1.0 Interface Standard, and TEAM (Test, Evaluation, And Measurement) Engine Capabilities.

**The OWS-9 sponsors are:** AGC (Army Geospatial Center, US Army Corps of Engineers), CREALF-GeoViQua-EC, EUROCONTROL, FAA (US Federal Aviation Administration), GeoConnections - Natural Resources Canada, Lockheed Martin Corporation, NASA (US National Aeronautics and Space Administration), NGA (US National Geospatial-Intelligence Agency), USGS (US Geological Survey), UK DSTL (UK MoD Defence Science and Technology Laboratory).
License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.


## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Scope</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Revision history</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Future work</td>
<td>1</td>
</tr>
<tr>
<td>1.4 Foreword</td>
<td>1</td>
</tr>
<tr>
<td>2 References</td>
<td>2</td>
</tr>
<tr>
<td>3 Background</td>
<td>2</td>
</tr>
<tr>
<td>3.1 OGC CITE program</td>
<td>2</td>
</tr>
<tr>
<td>3.2 Composition of a test suite and conformance classes</td>
<td>2</td>
</tr>
<tr>
<td>3.3 Introduction to TEAM Engine</td>
<td>3</td>
</tr>
<tr>
<td>3.4 Test scripts</td>
<td>3</td>
</tr>
<tr>
<td>3.4.1 CTL</td>
<td>4</td>
</tr>
<tr>
<td>3.4.2 TestNG</td>
<td>5</td>
</tr>
<tr>
<td>3.5 Reference Implementations</td>
<td>6</td>
</tr>
<tr>
<td>4 How to setup TEAM Engine</td>
<td>7</td>
</tr>
<tr>
<td>4.1 Requirements</td>
<td>7</td>
</tr>
<tr>
<td>4.2 Setting up via MAVEN</td>
<td>8</td>
</tr>
<tr>
<td>4.3 Installation and Configuration</td>
<td>12</td>
</tr>
<tr>
<td>4.4 Template Configuration File for New Tests</td>
<td>14</td>
</tr>
<tr>
<td>4.5 Running as a web server</td>
<td>15</td>
</tr>
<tr>
<td>4.5.1 Create a CATALINA_BASE directory</td>
<td>15</td>
</tr>
<tr>
<td>4.5.2 Configure Java Virtual Machine</td>
<td>15</td>
</tr>
<tr>
<td>4.5.3 Create contexts</td>
<td>16</td>
</tr>
<tr>
<td>4.5.4 Deploy the teamengine.war component and start the Tomcat instance</td>
<td>16</td>
</tr>
<tr>
<td>5 Downloading OGC Test Scripts</td>
<td>16</td>
</tr>
<tr>
<td>5.1 Accessing and installing OGC Tests</td>
<td>16</td>
</tr>
<tr>
<td>5.2 Running OGC Test via command line</td>
<td>17</td>
</tr>
<tr>
<td>6 How to test profiles</td>
<td>17</td>
</tr>
<tr>
<td>7 How to test your own data</td>
<td>18</td>
</tr>
<tr>
<td>8 How to access test reports</td>
<td>18</td>
</tr>
<tr>
<td>9 Support and additional help resources</td>
<td>19</td>
</tr>
<tr>
<td>9.1.1 Wiki</td>
<td>19</td>
</tr>
<tr>
<td>9.1.2 CITE Forum</td>
<td>19</td>
</tr>
</tbody>
</table>
**Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OGC Compliant logo.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Example of Test Suite Options</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Structure CTL script</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>List of Reference Implementations as of Dec 2012</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>TEAM Engine Code structure at Sourceforge</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Example MAVEN artifacts being downloaded</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>MAVEN Build Success Message</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>teamengine-core build</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>teamengine-core-distribution unzipped</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>TE BASE directory</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>Test Suites with revisions available at TEAM Engine Beta 4 (stage 2)</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>Scope GML 3.2 Test</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>Test Report</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>User Test Folder</td>
<td>19</td>
</tr>
</tbody>
</table>
OGC® OWS-9 CITE Help Guide Engineering Report

1 Introduction

1.1 Scope

This OGC® Engineering Report describes how to execute the CITE tests, how to select
the conformance classes, how to access and download the TEAM Engine, Test Scripts
and Reference Implementations, how to insert your own data and/or schemas along with
a section that provides other community help resources.

All questions regarding this document should be directed to the editor or the contributors:

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luis Bermudez (LB)</td>
<td>OGC</td>
</tr>
<tr>
<td>David Wesloh (DW)</td>
<td>NGA</td>
</tr>
</tbody>
</table>

1.2 Revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Editor</th>
<th>Primary clauses modified</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012.11.30</td>
<td>1</td>
<td>LB</td>
<td>All</td>
<td>Initial Document</td>
</tr>
<tr>
<td>2012.12.20</td>
<td>2</td>
<td>DW</td>
<td>ALL</td>
<td>Comments on the sections an minor editing</td>
</tr>
<tr>
<td>2012.12.26</td>
<td>3</td>
<td>DW</td>
<td>ALL</td>
<td>Added more content and figures to sections an minor editing.</td>
</tr>
<tr>
<td>2013.01.13</td>
<td>4</td>
<td>LB</td>
<td>Minor</td>
<td>Minor edits and removed comments and tracked changes.</td>
</tr>
</tbody>
</table>

1.3 Future work

1.4 Foreword

Attention is drawn to the possibility that some of the elements of this document may be
the subject of patent rights. The Open Geospatial Consortium shall not be held
responsible for identifying any or all such patent rights.
Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

Compliance Test Language (CTL) Best Practice: OGC 06-126
http://portal.opengeospatial.org/files/?artifact_id=33085

CITE Policies and Procedures: OGC 08-134r9
http://portal.opengeospatial.org/files/49237

3 Background

3.1 OGC CITE program

The OGC Compliance (CITE) Program provides the resources, procedures, and policies for improving software implementations' compliance with OGC standards. The Compliance Program provides an online free testing facility, a process for certification and branding of compliant products, and coordination of a vibrant community that develops and supports test scripts. The Compliance Program also runs plugfests, which are short term events for increasing interoperability among vendors' products.

3.2 Composition of a test suite and conformance classes

A Compliance Test Package is composed of an Abstract Test Suite (ATS) and an Executable Test Suite (ETS). An ATS is a set of testable assertions about the functionality of a standard, which an implementation must support in order to achieve compliance to the standard. An ATS is based on the conformance classes defined in the standard in accordance with the latest revision of “OGC 08-131r3 The Specification Model — A Standard for Modular specifications”. An ETS is a set of code (e.g. Java and CTL) that provides runtime tests for the assertions defined by the ATS. Test data required to do the tests are part of the ETS.
3.3 Introduction to TEAM Engine

The OGC’s accepted Compliance Test Engine is an open source product called TEAM Engine. TEAM Engine (Test, Evaluation, And Measurement Engine) is a software program for testing web services, clients and encodings. It executes test scripts written in Compliance Test Language (CTL), TestNG and other languages. It is lightweight and easy to run as a command line or to setup as a service. It can be used to test any type of service or encoding, not only OGC. It is also the official tool used by the Open Geospatial Consortium (OGC) for compliance testing. It helps verify that a software implementation is conformant with a standard in order for the implementation to get certified. Once the software implementation gets certified, the organization owner of the software can use the compliance OGC logo in marketing materials.

![OGC Compliant logo](image)

Figure 1 – OGC Compliant logo.

3.4 Test scripts

The Test Scripts composition is explained in 3.2. It is a set of code that implements the ATS. The tests can be organized as a set of conformance classes or it can be presented with options that the user running the test can select. Figure 2 shows an example of possible selections for WFS 1.1.0.
Compliance test suite for Web Feature Service (WFS) 1.1.0

Service metadata

Please provide a URL from which a capabilities document can be retrieved. Modify the URL template below to specify the location of an OGC capabilities document for the WFS implementation under test (this can refer to a static document or to a service endpoint).

Capabilities URL:  http://hostname:port/path/to/file

Supported conformance classes

A conformance class denotes a set of functional capabilities provided by the WFS under test.

<table>
<thead>
<tr>
<th>Conformance Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFS-Basic (Required)</td>
<td>Implements GetCapabilities, DescribeFeatureType and GetFeature requests</td>
</tr>
<tr>
<td>WFS-Transaction</td>
<td>Implements the Transaction request (LockFeature and GetFeatureWithLock are optional)</td>
</tr>
<tr>
<td>WFS-XLink</td>
<td>Implements the GetGmlObject request and supports (local) XLink processing in GetFeature requests</td>
</tr>
</tbody>
</table>

GML Simple Features (GMLSF) compliance level

This indicates the scope of GML 3.1 support, as documented in OGC 06-049. Test data for levels SF-0 and SF-1 are currently available. This setting is ignored when assessing XLink conformance because it requires support at the SF-2 level.

- SF-0: Level 0 (only simple non-spatial property types; Curve and Surface geometries are excluded)
- SF-1: Level 1 (complex non-spatial property types, plus Curve and Surface geometries)

WARNING: Don't forget to add the test data!

Figure 2 – Example of Test Suite Options

The Figure shows that there are three conformance classes in the test: WFS-Basic, WFS-Transaction and WFS-Link. Note that the basic level cannot be pre-selected because it is the required conformance class to get WFS 1.1.0 certification. The user can also select the level of GML.

Tests that ran in TEAM Engine can be encoded in CTL and TestNG. These are described in the following sections.

3.4.1 CTL

Compliance Test Language or CTL is an XML grammar, based on XSLT, for documenting and scripting suites of tests for verifying that an implementation of a specification complies with the specification. A more complete description of CTL is available as an OGC Discussion Paper (OGC Document 06-126).

A test suite in CTL consists of a set of objects. The initial object is a suite, which identifies a starting test and may also include a form with instructions. The starting test is called using the values the user enters in the form as parameters and contains instructions that may call other tests or use functions and parsers.
A suite may optionally be extended by a profile to test functionality that is not mandatory in the base test suite. A profile identifies the base test suite it is extending and a starting profile test. When a profile is executed, the starting tests from both the base test suite and the profile are called, and both are passed the values from the base test suite’s form as parameters. See Figure 3.

Figure 3 – Structure CTL script

A CTL file is an XML file that contains a CTL object or a package element as its root element. The package element is a container for multiple CTL objects. Each CTL object is identified by a unique, namespace qualified, name, so the set of objects in a test suite may span several files. Test objects contain programmatic code that consists of XSL instructions and/or CTL instructions. Some parser (e.g. SOAP) objects are built-in to CTL, and may be used without declaring a parser object.

3.4.2 TestNG

(From the http://testng.org/doc/index.html)

TestNG is a testing framework based on JUnit and NUnit. It provides the following new functionalities:

- Use of Annotations.
- Allow running tests in in arbitrarily big thread pools with various policies available (all methods in their own thread, one thread per test class).
- Tests are multithread safe.
- Support for data-driven testing.
- Support for parameters.
- Powerful execution model (no more TestSuite).
- Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc.).
- Embeds BeanShell for further flexibility.
- Default JDK functions for runtime and logging (no dependencies).
- Dependent methods for application server testing.

### 3.5 Reference Implementations

A Reference Implementation (OGC 08-134r9) is a fully functional, licensed copy of a tested, branded software that has passes the test for an associated conformance class in a version of an Implementation Standard. It is free and publicly available for testing as a web service or via download. The Reference Implementation does not need to pass all the conformance classes within the standard. In most of the cases, the Reference Implementation will pass at least the core and possibly some number of extension conformance classes. Multiple reference implementations can exist for an associated version of an Implementation Standard. The coordinator after reviewing the results and checking the public interface of the software will determine if the implementation can be a reference implementation.

OGC will maintain a page that explains the scope of each reference implementation. Currently it is located at: [http://cite.opengeospatial.org/reference](http://cite.opengeospatial.org/reference). OGC will also provide this information in the public data base at the main OGC portal.
As of December 2012, all the listed reference implementations, fully implement all the tests. Meaning that the implementation passes all the advertised tests in TEAM Engine.

OGC will make its best effort to host Reference Implementations on an OGC server to help others in the community to develop compliant implementations.

4 How to setup TEAM Engine

TEAM Engine can be setup to run via a web server or be invoked via command line. For developers invoking TEAM Engine via command line is very convenient since the user can very fast test changes in the implementation code. Setting it up via a Web Service can help communities making available a central web test facility to test standards of interest to the community.

4.1 Requirements

TEAM engine requires JAVA 1.6. For web installation it is recommended a dedicated TOMCAT server. The installation server at OGC is as follows:
- 3GHz (4 core) Xeon processor
- 2 GB RAM
- Debian Linux
- 3 GB of data storage for 800 sessions. Each session is a test run stored as uncompressed xml files.

### 4.2 Setting up via MAVEN

TEAM Engine is available via MAVEN, which is easy for developers to integrate and easy to build. MAVEN is a software project management tool that helps to integrate distributed code. MAVEN provides a common framework to build and release code. Via MAVEN the executable TEAM Engine code is created. Apache Maven 3.0 is required to build the teamengine code base. MAVEN is supported by most Integration Development Environments (IDE) tools such as Eclipse. MAVEN can be downloaded at: [http://maven.apache.org](http://maven.apache.org)

TEAM Engine code is available at Sourceforge and can be downloaded via SVN [http://sourceforge.net/projects/teamengine/](http://sourceforge.net/projects/teamengine/).

The code is available when selecting the **Code** option or typing the following

[http://sourceforge.net/p/teamengine/code/]
To connect via an SVN client you can use the following URL

svn://svn.code.sf.net/p/teamengine/code/trunk

When opening the root the following folder and files will be shown.
TEAM Engine consists in the following MAVEN modules:

- teamengine-core: Main CTL script processor
- teamengine-web: Web interface
- teamengine-resources: Shared resources (stylesheets, schemas, etc.)
- teamengine-config: Configuration utility
- teamengine-realm: Custom Tomcat user realm
- teamengine-spi: Extensibility framework and REST API

Once downloaded, simply run `mvn package` in the root project directory to generate all build artifacts, or mvn install to add the artifacts to the local repository. When installing, all needed code is downloaded from various MAVEN repositories. Depending on the Internet connection this process may take 1-5 minutes.
Once all the code is downloaded you should see a BUILD SUCCESS message as shown in Figure 7.

```
[INFO] Reactor Summary:
[INFO]
[INFO] TEAM Engine .................................... SUCCESS [9.22s]
[INFO] TEAM Engine - Tomcat Realm .................. SUCCESS [23.55s]
[INFO] TEAM Engine - Shared Resources ............. SUCCESS [5.65s]
[INFO] TEAM Engine - Service Providers .......... SUCCESS [13.67s]
[INFO] TEAM Engine - Configuration Utility ....... SUCCESS [1.42s]
[INFO] TEAM Engine - Core Module .................. SUCCESS [24.61s]
[INFO] TEAM Engine - Web Module ................. SUCCESS [15.31s]
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total Time: 1:34.014s
[INFO] Finished at: Wed Dec 26 00:47:36 EST 2012
[INFO] Final Memory: 22M/81M
```

Figure 7 – MAVEN Build Success Message

The main build artifacts include:

- `teamengine-core-${version}-distribution.zip`
  - The core binary distribution for command line usage.
- `teamengine-core-${version}-base.[zip | tar.gz]`
  - Content of the main configuration directory (TE_BASE).
teamengine.war
    The JEE web (Servlet) application.

teamengine-common-libs.[zip | tar.gz]
    Common runtime dependencies (e.g. JAX-RS 1.1).

The generated files (after running mvn package) are found under the target folder. For example under the teamengine-core the were various directories and zip files created as shown in Figure 8.

![Figure 8 – teamengine-core build](image)

4.3 Installation and Configuration

To run via command line you need the teamengine-core-xxx-distribution.zip generated in the previous step (See Figure 8).

Unzip the file (e.g. teamengine-core-4.0-beta1-distribution.zip) and you will be ready to run team engine. The unzip directory should like Figure 9
Figure 9 – teamengine-core-distribution unzipped

The bin folder contains all the command executable. The docs folder contains documentation, and the lib and resources folders contain the code needed to run TEAM Engine via command line.

The next step is to setup a directory where all the tests, session and temporary files will be stored. This is called TE_BASE (Figure 10). The value of the TE_BASE system property or environment variable specifies the location of the main configuration directory that contains several essential sub-directories. The process includes:

- Creation of a TE_BASE directory in the server (e.g. /srv/teamengine)
- Unpacking the contents of the teamengine-core-*/base archive into this location.
- Ensuring that users (including the Tomcat user) have write access.

Figure 10 – TE BASE directory
CTL test scripts are placed in the TE_BASE/scripts directory and the TE_BASE/config.xml file is updated accordingly.

The scripts are available at the OGC SVN repository, which has anonymous read access

https://svn.opengeospatial.org/ogc-projects/cite/scripts/

Figure 11 shows the tests available by Dec 2012 in the TEAM Engine beta 4.

<table>
<thead>
<tr>
<th>Available test suites</th>
<th>Version</th>
<th>Test Suite Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>OGC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Feature Service (WFS)</td>
<td>2.0</td>
<td>2.0-beta1</td>
</tr>
<tr>
<td>Web Feature Service (WFS)</td>
<td>1.1.0</td>
<td>r10</td>
</tr>
<tr>
<td>Geography Markup Language (GML)</td>
<td>3.2.1</td>
<td>3.2-beta1</td>
</tr>
<tr>
<td>Web Map Service (WMS)</td>
<td>1.3.0</td>
<td>r4</td>
</tr>
<tr>
<td>Web Map Service (WMS)</td>
<td>1.1.1 (deprecated)</td>
<td>r2</td>
</tr>
<tr>
<td>Web Map Service (WMS) - Client</td>
<td>1.3.0</td>
<td>r1</td>
</tr>
<tr>
<td>Web Coverage Service (WCS)</td>
<td>2.0.1</td>
<td>r1</td>
</tr>
<tr>
<td>Web Coverage Service - Earth Observation Profile</td>
<td>1.0 (pending)</td>
<td>r2</td>
</tr>
<tr>
<td>Sensor Observation Service (SOS)</td>
<td>2.0</td>
<td>r3</td>
</tr>
<tr>
<td>Sensor Planning Service (SPS)</td>
<td>2.0</td>
<td>r3</td>
</tr>
<tr>
<td>Sensor Planning Service (SPS)</td>
<td>1.0 (deprecated)</td>
<td>r2</td>
</tr>
<tr>
<td>OWS Context (OWC)</td>
<td>1.0 (pending)</td>
<td>1.0-SNAPSHOT</td>
</tr>
<tr>
<td>Sensor Model Language (SensorML)</td>
<td>1.0.1</td>
<td>r4</td>
</tr>
<tr>
<td>Catalogue Service - Web (CSW)</td>
<td>2.0.2</td>
<td>r4</td>
</tr>
</tbody>
</table>

Figure 11 – Test Suites with revisions available at TEAM Engine Beta 4 (stage 2)

The table shows the specification name, the version and the test suite revision. For example, WFS 1.1.0 has had 10 revisions. To run the latest revision of WFS 1.1.0 the test that needs to be downloaded and copied into the TE_BASE/scripts is:

https://svn.opengeospatial.org/ogc-projects/cite/scripts/wfs/1.1.0/tags/r10/

### 4.4 Template Configuration File for New Tests

A new TEAM Engine installation contains an entry for a sample test script ("XML Note"). Every OGC test suite should include a configuration file. If a new test suite is to be create this file can be copied and edited. An entry for the WFS 2.0 test suite is shown below.
4.5 Running as a web server

Apache Tomcat 7.0 (with JDK 6 or later) is a supported servlet container. It is strongly recommended that a dedicated Tomcat instance be created to host the teamengine application. Create one as suggested below.

4.5.1 Create a CATALINA_BASE directory

Under Windows do the following:

> mkdir base-1 & cd base-1
> xcopy %CATALINA_HOME%\conf conf\n
Under GNU/Linux do the following

$ sudo mkdir -p /srv/tomcat/base-1; cd /srv/tomcat/base-1
$ sudo cp -r $CATALINA_HOME/conf .
$ sudo mkdir lib logs temp webapps work

4.5.2 Configure Java Virtual Machine

These JVM options are recommended for the Tomcat instance:

CATALINA_OPTS="-server -Xmx1024m -XX:MaxPermSize=128m -DTE_BASE=$TE_BASE -Dderby.system.home=$DERBY_DATA"
4.5.3 Create contexts

Unpack the contents of the teamengine-common-libs archive into the CATALINA_BASE/lib directory. Add the following contexts for static web content to CATALINA_BASE/conf/Catalina/localhost/.

docs.xml

```xml
<Context path="/docs" docBase="${TE_BASE}/resources/docs" />
```

reports.xml

```xml
<Context path="/reports" docBase="${TE_BASE}/users" />
```

about.xml

```xml
<Context path="/about" docBase="${TE_BASE}/scripts" />
```

4.5.4 Deploy the teamengine.war component and start the Tomcat instance.

Once the war is installed, start Tomcat. The following URIs provide starting points for discovering and executing test suites:

- /teamengine - Home page for selecting and running CTL test suites
- /teamengine/rest/suites - Shows a listing of available (TestNG) test suites. See CiteTestNG for details.

5 Downloading OGC Test Scripts

5.1 Accessing and installing OGC Tests

OGC Tests are available via OGC SVN:
Download the scripts you want to test and copy them under the scripts directory (See section 4.3). Then follow the instructions in 4.3.

### 5.2 Running OGC Test via command line

Once TEAM Engine and tests are correctly installed, following instructions in section 4.2 and 4.3, the command line can be used to run the tests. For example:

```
C:\teamengine2.0>bin\test -source=hello.ctl
```

### 6 How to test profiles

As of December 2012 only profiles related to GML can be tested. The test verifies conformance of GML data and application schemas with respect to ISO 19136:2007 (GML 3.2.1). As shown in Figure 12, a conforming GML data instance must refer to the relevant GML application schema, which in turn imports the complete GML schema.

![Figure 12 – Scope GML 3.2 Test](image-url)
7 How to test your own data

Test Suites to test encodings will be available in 2013. This will include GML, SensorML and KML. The procedure is the same as testing a server. TEAM Engine will provide an HTML form to capture the information about the location of the file. Then the file will be uploaded to the engine and tested.

8 How to access test reports

Results for session s0055

Test Suite: OGC WMS 1.1.1 Test Suite

- Test wms:wms_main (View Details): Failed (Inherited Failure)
  - Test wms:wmsops-getmap-params-exceptions-3 (View Details): Passed
  - Test wms:basic_elements-param_rules-order_and_case-3 (View Details): Passed
  - Test wms:basic_elements-version-negotiation-2 (View Details): Passed
  - Test wms:basic_elements-version-negotiation-4 (View Details): Passed
  - Test wms:dims-declaring-3 (View Details): Passed
  - Test wms:dims-declaring-4 (View Details): Passed
  - Test wms:profiles-queryable-1 (View Details): Failed
  - Test wms:profiles-queryable-2 (View Details): Passed
  - Test wms:profiles-queryable-3 (View Details): Failed
  - Test wms:wmsops-getcapabilities-output_formats-1 (View Details): Passed
  - Test wms:wmsops-getcapabilities-params-request-1 (View Details): Passed
  - Test wms:wmsops-getcapabilities-params-updateSequence-1 (View Details): Passed
  - Test wms:wmsops-getcapabilities-params-version-1 (View Details): Passed
  - Test wms:wmsops-getcapabilities-params-version-2 (View Details): Passed
  - Test wms:wmsops-getcapabilities-params-version-3 (View Details): Passed

Figure 13 – Test Report

The example test report shows the result of a OWS WMS 1.1.1 test run. It shows that the main test failed, because sub-tests failed (e.g. Test wms:profiles-queryable-1). The user has the option to view the details of what was tested on each step.

The reports are stored under the users folder in TE_BASE. An example of the structure of the folder for the user ogctest is presented in Figure 14.
Figure 14 - User Test Folder

For each session there is folder created. Under each folder there are xml files that provide the information of the run. The log.xml has information about the test that was run, information about the implementation (e.g. URL of capabilities document) and contains an xml tag endtest, where if result =0, it means it passed the test.

<endtest result="0"/>

9 Support and additional help resources

There are various resources provided by the CITE program to support testers, the TEAM Engine community and test developers. These are as follows:

9.1.1 Wiki

The wiki provides public information and its being updated constantly with information about how better use TEAM Engine. Link is here:

http://cite.opengeospatial.org

9.1.2 CITE Forum

The CITE forum link is as follows:

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

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

The CITE Forum is supported by Test Leads, which are individuals from organizations that have an agreement with OGC to maintain the test. This agreement is via in-kind contribution, as an exchange for membership or paid via the OGC interoperability program activities. The Leads make their best effort to respond to any posted message within 24 hours.