



TestCases for the SCA Web Service Binding Specification Version 1.1

Committee Specification Draft 012 /
Public Review ~~01~~Draft 02

~~14~~ July 2010~~1~~

Specification URIs:

This ~~V~~version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csprd02.pdf>
(Authoritative)

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csprd02.html>

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csprd02.odt>

Previous ~~V~~version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-cd01.pdf>~~N/A~~

~~Latest Version:~~

(Authoritative)

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-cd01.html>

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-cd01.odt>

Latest version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases.pdf> (Authoritative)

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases.html>

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases.odt>

Technical Committee:

OASIS Service Component Architecture / Bindings (SCA-Bindings) TC

~~OASIS Service Component Architecture / Bindings (SCA-Bindings) TC~~

Chair(s)::

Simon Holdsworth (simon_holdsworth@uk.ibm.com, ~~IBM~~) IBM

Editor(s)::

Mike Edwards (mike_edwards@uk.ibm.com, ~~IBM~~) IBM

Related Additional Work Product artifacts:

This prose specification is one component of a Work Product which also includes:

- Test Suite artifacts (provided in two formats)

- ZIP archive:

<http://docs.oasis-open.org/opencsa/sca-bindings/TestCases-BWS-csprd02.zip>

- Online resource:

<http://docs.oasis-open.org/opencsa/sca-bindings/TestCases-BWS-csprd02/document>

Related work:

This specification is related to:

- *Service Component Architecture Web Services Service Binding Specification Version 1.1. Latest version.* <http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec-cd04.pdf.html>

Declared XML Namespace(s):namespaces:

<http://docs.oasis-open.org/ns/opencsa/scatests/200903>
<http://docs.oasis-open.org/ns/opencsa/scatests/2009032>
<http://test.sca.oasisopen.org/>

Abstract:

This document defines the TestCases for the SCA Web Service Binding ~~s~~Specification. The TestCases represent a series of tests that an SCA runtime must pass in order to claim conformance to the requirements of the SCA Web Service Binding specification.

Status:

This document was last revised or approved by the OASIS Service Component Architecture / Bindings (SCA-Bindings) TC on the above date. The level of approval is also listed above. Check the "Latest ~~Version~~" or "~~Latest Approved Version~~version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "~~Send A Comment~~" "~~Send A Comment~~" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/sca-bindings/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/sca-bindings/ipr.php>).

[The non-normative errata page for this specification is located at\).](#)

Citation format:

[When referencing this specification the following citation format should be used:](#)

[SCA-WSBindingTestCases]

[TestCases for the SCA Web Service Binding Specification Version 1.1. 14 July 2011. OASIS Committee Specification Draft 02 / Public Review Draft 02.](#)
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csprd02.html>

Notices

Copyright © OASIS ~~©2010~~, [Open 2011](#). All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "~~OASIS is a trademark~~", "[SCA](#)", and "[Service Component Architecture](#)" are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	6
1.1	TestCase Structure.....	6
1.1.1	Test Client	6
1.1.2	Test Application.....	6
1.1.3	Test Artifacts Organization	7
1.2	Namespaces and Java Package Names.....	8
1.2.1	SCA Artifact Namespaces.....	8
1.2.2	WSDL Namespace.....	8
1.2.3	Java Package name.....	8
1.3	Terminology	8
1.4	Normative References.....	8
2	TestCases	9
2.1	TestCases for Web Service Binding Specification Section 2	9
2.1.1	BWS_2001_TestCase.....	9
2.1.2	BWS_2002_TestCase.....	9
2.1.3	BWS_2003_TestCase.....	10
2.1.4	BWS_2004_TestCase.....	10
2.1.5	BWS_2005_TestCase.....	11
2.1.6	BWS_2006_TestCase.....	11
2.1.7	BWS_2007_TestCase.....	12
2.1.8	BWS_2008_TestCase.....	12
2.1.9	BWS_2009_TestCase.....	13
2.1.10	BWS_2010_TestCase.....	13
2.1.11	BWS_2011_TestCase.....	14
2.1.12	BWS_2012_TestCase.....	15
2.1.13	BWS_2013_TestCase.....	15
2.1.14	BWS_2014_TestCase.....	16
2.1.15	BWS_2015_TestCase.....	16
2.1.16	BWS_2016_TestCase.....	17
2.1.17	BWS_2017_TestCase.....	17
2.1.18	BWS_2018_TestCase.....	18
2.1.19	BWS_2019_TestCase.....	19
2.1.20	BWS_2020_TestCase.....	19
2.1.21	BWS_2021_TestCase.....	19
2.1.22	BWS_2022_TestCase.....	20
2.1.23	BWS_2023_TestCase.....	21
2.2	TestCases for Web Service Binding Specification Section 4	21
2.2.1	BWS_4001_TestCase.....	21
2.2.2	BWS_4002_TestCase.....	22
2.2.3	BWS_4003_TestCase.....	22
2.2.4	BWS_4004_TestCase.....	22
2.2.5	BWS_4005_TestCase.....	23
2.2.6	BWS_4006_TestCase.....	23
2.2.7	BWS_4007_TestCase.....	24

2.2.8 BWS_4008_TestCase.....	24
2.3 TestCases for Web Service Binding Specification Section 5	25
2.3.1 BWS_5001_TestCase.....	25
2.3.2 BWS_5002_TestCase.....	26
2.3.3 BWS_5003_TestCase.....	26
2.3.4 BWS_5004_TestCase.....	27
2.3.5 BWS_5005_TestCase.....	27
2.3.6 BWS_5006_TestCase.....	28
3 Catalog of Test Artifacts.....	29
3.1 Composite Files - lower level.....	29
3.2 Java Interfaces	29
3.3 Java Implementation Classes.....	30
3.4 WSDL Interface Files	30
4 Conformance.....	34
Appendix A. Test Assertions for the Web Service Binding	35
A.1. Example Test Assertion	35
A.2. Test Assertions for Web Service Binding Specification Section 2	35
A.3. Test Assertions for Web Service Binding Specification Section 4	45
A.4. Test Assertions for Web Service Binding Specification Section 5	46
Appendix B. Cross Mapping of Normative Statements to Assertions.....	50
Appendix C. Cross Mapping of Test Assertions to TestCases	52
Appendix D. Acknowledgments	55
Appendix E. Revision History.....	56

1 Introduction

This document defines the testcases for the SCA Web Service Bindings specification **[SCA-WSBinding]**. The tests described in this document are derived from the normative statements in SCA Web Service Binding specification via Test Assertions which are described in Appendix A, "Web Service Binding Test Assertions" [related to the Test Assertions described i.](#)

1.1 TestCase Structure

The SCA Bindings testcases follow a standard structure. They are divided into two main parts:

1. Test Client, which drives the test and checks that the results are as expected
2. Test Application, which forms the bulk of the testcase and which consists of Composites, WSDL files, XSDs and code artifacts such as Java classes, organized into a series of SCA contributions

The basic idea is that the Test Application runs on the SCA runtime that is under test, while the Test Client runs as a standalone application, invoking the Test Application through one or more service interfaces.

1.1.1 Test Client

The test client is designed as a standalone application. The version built here is a Java application which uses the JUnit test framework, although in principle, the client could be built using another implementation technology.

The test client is structured to contain configuration information about the testcase, which consists of:

1. metadata identifying the Test Application in terms of the SCA Contributions that are used and the Composites that must be deployed and run
2. data indicating which service operation(s) must be invoked with input data and expected output data (including exceptions for expected failure cases)

The Java test client consists of a base runtime class, BaseJAXWSTestCase.java. Each actual testcase is implemented by a small class which extends the base runtime class. The bulk of the code required to run a test is held in the base runtime class. The small testcase class contains the configuration for the specific test, which it provides to the code in the base runtime class through a standard interface.

The Java test client base runtime class is structured so that there is a replaceable class called the RuntimeBridge, which is used to communicate with the SCA runtime under test, for the purposes of deploying and running the test application. Each SCA runtime provider can produce a version of this class. The code within the runtime bridge is likely to be highly proprietary and specific to the SCA runtime for which it is written. Which runtime bridge class is used at runtime is controlled by an environment variable or system variable with the name "OASIS_TESTENV_RUNTIME_BRIDGE_CLASS", which is read by the code in BaseJAXWSTestCase.

The Test Client defaults to using Web services to communicate with the test application.

1.1.2 Test Application

Each Test Application consists of one top level SCA Composite file and one or more other SCA Composite files and their associated artifacts (implementations, interface files), plus test client invocation application described above.

A typical test application has a design where the top level composite offers a single service to the client application over a Web services binding. The top level composite contains one component which offers the service that is used by the client application. The top level composite then contains one or more other components which are used by the first component.

All of the components in the top level composite are implemented by composites. These second level composites then contain typically one component, implemented using a specific technology such as Java POJO or a BPEL process. In some cases the implementation may be a third level composite.

The application is structured so that alternative technologies can be used. For example, replacing the contents of the second-level or third-level composites allows different implementation technologies to be tested – eg BPEL scripts or C++ classes may be used instead of Java classes. Similarly, the binding used to connect from the top level composite to the client application may be changed from Web services to JMS if required, simply by changing the binding on the <service/> of the top level composite.

Which implementation language to use for test artifacts is controlled by a system variable or environment variable which is read by the test client application, with the name "OASIS_TESTENV_IMPL_LANG".

This variable can have one of the following values:

- "Java" - for Java implementations
- "BPEL" - for WS_BPEL process implementations
- "CPP" - for C++ implementations

The testcases are designed so that the range of implementation types can be expanded

1.1.3 Test Artifacts Organization

Note that the design of these testcases promotes reuse of artifacts between testcases, so that many testcases share components. For example, components implementing simple invocable services are all implemented using a single parameterized implementation artifact.

All the test artifacts are contained in a number of Contributions, which are simply filesystem directories which are all peers in the filesystem hierarchy. The names of the directories are the names of the Contributions and the names are significant. The names of Contributions containing implementation type specific artifacts (such as Java classes) are also specially structured to allow for replacement of one type of implementation artifact with another.

Broadly, Contribution names are as follows:

- BWS_nnnn - a contribution that is specific for a particular testcase, where "nnnn" is the number of the testcase. Often this is required because a particular testcase involves artifacts that contain errors that are statically checkable - an SCA runtime is permitted to reject such artifacts when they are contributed and deployed and it is important to ensure that contributions containing deliberate errors for one testcase do not interfere with the operation of other testcases.
- BWS_nnnn_Java - a contribution for a specific testcase where there is a need for language specific artifacts that relate to that testcase alone
- BWS_General - a shared contribution containing implementation type independent artifacts that can be used by many testcases.
- BWS_General_Java - a shared contribution containing implementation type dependent artifacts for Java POJOs. These artifacts can include both Java classes and also SCA composites that directly use Java classes.
- Contribution1, Contribution2, etc - contributions that are used by various testcases that are testing the handling of SCA contributions

Note that the names of Contributions containing implementation specific artifacts ends with a name that is specific to the implementation type - so "_Java" is used for Java implementations, "_BPEL" is used for BPEL implementations, "_CPP" is used for C++ implementations (and so on). Note that the name following the underscore matches the name used in the "OASIS_TESTENV_IMPL_LANG" variable used to control execution of the test client. The concept is that where there is an implementation type specific contribution, each implementation type must provide its own versions of the same basic artifacts. Typically, this means that each contribution must contain the same set of Composites, but that the implementation type dependent artifacts that these composites use will differ from implementation type to implementation type.

Basically, the setting of the variable is used to select the suffix used for implementation type dependent contributions. If the variable is set to "Java" then the contribution "General_Java" is selected, whereas if the variable is set to "BPEL", the contribution "General_BPEL" is selected.

1.2 Namespaces and Java Package Names

The SCA Web Service Binding testcase suite makes use of some XML namespaces and Java package names, as follows:

1.2.1 SCA Artifact Namespaces

These apply to artifacts such as Composites

<http://docs.oasis-open.org/ns/opencsa/scatests/200903>

<http://docs.oasis-open.org/ns/opencsa/scatests/2009032>

1.2.2 WSDL Namespace

<http://test.sca.oasisopen.org/>

1.2.3 Java Package name

For Java interface classes and for Java implementation classes

`org.oasisopen.sca.test`

1.3 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 [RFC 2119]

1.4 Normative References

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997.
<http://www.ietf.org/rfc/rfc2119.txt>.

[SCA-WS-
TA-GUIDE] OASIS Committee Draft 045, "~~Test Assertions for the SCA Web Service Binding Guidelines~~ **Version 1.1 Specification**", July 0, August 2010
<http://docs.oasis-open.org/tag/guidelines/v1.0/cd05/testassertionsguidelines-cd-05.pdf>.

[SCA-WSBinding] OASIS Committee Draft 04, "Service Component Architecture Web Service Binding Specification Version 1.1," May 2010.
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec-cd04.pdf>

2 TestCases

2.1 [TestCases for Web Service Binding Specification](#) Section 2

2.1.1 BWS_2001_TestCase

Testcase ID	BWS_2001_TestCase
Test Assertion	BWS-TA-20001
Description	Tests that the value of the @uri attribute of a <binding.ws/> subelement of a <component/> <reference/> element is an absolute URI.
Artifacts	BWS_2001_TestCase.java Test_BWS_2001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

2.1.2 BWS_2002_TestCase

Testcase ID	BWS_2002_TestCase
Test Assertion	BWS-TA-20002
Description	Tests that a binding.ws/@wsdlElement attribute value points to an existing WSDL 1.1 element
Artifacts	BWS_2002_TestCase.java Test_BWS_2002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

2.1.3 BWS_2003_TestCase

Testcase ID	BWS_2003_TestCase
Test Assertion	BWS-TA-20003
Description	Tests that a binding.ws/@wsdlElement attribute of a <service/> element of a <component/> does not point to a WSDL 1.1 service element
Artifacts	BWS_2003_TestCase.java Test_BWS_2003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

2.1.4 BWS_2004_TestCase

Testcase ID	BWS_2004_TestCase
Test Assertion	BWS-TA-20004
Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL service element, that the set of available WSDL ports is not empty.
Artifacts	BWS_2004_TestCase.java Test_BWS_2004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Positive test: "BWS_2004 request service1 operation1 invoked service2 operation1 invoked"

2.1.5 BWS_2005_TestCase

Testcase ID	BWS_2005_TestCase
Test Assertion	BWS-TA-20005 BWS_TA_20010

Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL service element, and the WSDL service element does not have any ports that are supported by the SCA runtime, the SCA runtime raises an error.
Artifacts	BWS_2005_TestCase.java Test_BWS_2005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2005 sca-variables.dtd in contribution BWS_2005
Expected output	Negative test: "exception"

2.1.6 BWS_2006_TestCase

Testcase ID	BWS_2006_TestCase
Test Assertion	BWS-TA-20006
Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL service element, and the WSDL service element has 2 or more WSDL Ports available, that the SCA Runtime uses exactly one port for each invocation through the reference
Artifacts	BWS_2006_TestCase.java Test_BWS_2006.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Positive test - 2 alternative outcomes: 1) "BWS_2006 request service1 operation1 invoked service2 operation1 invoked" 2) "BWS_2006 request service1 operation1 invoked service3 operation1 invoked"

2.1.7 BWS_2007_TestCase

Testcase ID	BWS_2007_TestCase
-------------	-------------------

Test Assertion	BWS-TA-20007
Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL port element, that the WSDL portType element associated with the WSDL port element is compatible with the service interface.
Artifacts	BWS_2007_TestCase.java Test_BWS_2007.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1InvalidService.wsdl
Expected output	Negative test: "exception"

2.1.8 BWS_2008_TestCase

Testcase ID	BWS_2008_TestCase
Test Assertion	BWS-TA-20008
Description	Tests that where a <component/> <service/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL port element, that the endpoint specified by the port element is exposed by the SCA runtime
Artifacts	BWS_2008_TestCase.java Test_BWS_2008.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1WithPort.wsdl
Expected output	Positive test: "BWS_2008 request service1 operation1 invoked service2 operation1 invoked"

2.1.9 BWS_2009_TestCase

Testcase ID	BWS_2009_TestCase
Test Assertion	BWS-TA-20009

Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL port element, that the portType specified by the WSDL port is a compatible superset of the interface on the reference
Artifacts	BWS_2009_TestCase.java Test_BWS_2009.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite9.composite Service1.wsdl Service1Superset.wsdl sca_variables.dtd in BWS_2009 contribution
Expected output	Positive test: "BWS_2009 request service1 operation1 invoked service2 operation1 invoked"

2.1.10 BWS_2010_TestCase

Testcase ID	BWS_2010_TestCase
Test Assertion	BWS-TA-20011
Description	Tests that where a <component/> <service/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL binding element, that the portType associated with the WSDL port is compatible with the interface on the service
Artifacts	BWS_2010_TestCase.java Test_BWS_2010.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl Service1WithBinding.wsdl
Expected output	Positive test: "BWS_2010 request service1 operation1 invoked service2 operation1 invoked"

2.1.11 BWS_2011_TestCase

Testcase ID	BWS_2011_TestCase
Test Assertion	BWS-TA-20012

Description	Tests that where a <component/> <service/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL binding element and the WSDL binding element is not supported by the SCA runtime, an error is raised
Artifacts	BWS_2011_TestCase.java Test_BWS_2011.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl Service1WithBinding.wsdl
Expected output	Negative test: "exception"

2.1.12 BWS_2012_TestCase

Testcase ID	BWS_2012_TestCase
Test Assertion	BWS-TA-20013 BWS-TA-20015 (this testcase uses the @uri to identify the target service and so covers that aspect of that test assertion)
Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL binding element, that the portType specified by the WSDL binding is a compatible superset of the interface on the reference
Artifacts	BWS_2012_TestCase.java Test_BWS_2012.composite TestInvocation.wsdl TestClient_0002.composite TestComposite4.composite TestComposite9.composite Service1.wsdl Service1Superset.wsdl in BWS_2012 contribution sca_variables.dtd in BWS_2012 contribution
Expected output	Positive test: "BWS_2012 request service1 operation1 invoked service2 operation1 invoked"

2.1.13 BWS_2013_TestCase

Testcase ID	BWS_2013_TestCase
Test Assertion	BWS-TA-20014

Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL binding element, and the WSDL binding element is not supported by the SCA runtime, the SCA runtime raises an error.
Artifacts	BWS_2013_TestCase.java Test_BWS_2013.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2013 sca-variables.dtd in contribution BWS_2013
Expected output	Negative test: "exception"

2.1.14 BWS_2014_TestCase

Testcase ID	BWS_2014_TestCase
Test Assertion	BWS-TA-20015 (this testcase uses the EndpointReference element to identify the target service and so covers that aspect of that test assertion)
Description	Tests that where a <component/> <reference/> has a <binding.ws/> element with @wsdlElement attribute referencing a WSDL binding element, that when the endpoint address of the target service is supplied as an <wsa:EndpointReference/> element, the service can be successfully invoked
Artifacts	BWS_2014_TestCase.java Test_BWS_2014.composite TestInvocation.wsdl TestClient_0002.composite TestComposite4.composite TestComposite9.composite Service1.wsdl Service1Superset.wsdl in BWS_2014 contribution sca_variables.dtd in BWS_2014 contribution
Expected output	Positive test: "BWS_2014 request service1 operation1 invoked service2 operation1 invoked"

2.1.15 BWS_2015_TestCase

Testcase ID	BWS_2015_TestCase
Test Assertion	BWS-TA-20016
Description	Tests that where a <binding.ws/> element has a @wsdli:wSDLLocation attribute specified that it also has a @wsdlElement attribute specified
Artifacts	BWS_2015_TestCase.java Test_BWS_2015.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2015 sca-variables.dtd in contribution BWS_2015
Expected output	Negative test: "exception"

2.1.16 BWS_2016_TestCase

Testcase ID	BWS_2016_TestCase
Test Assertion	BWS-TA-20017
Description	Tests that where a <binding.ws/> element has a @wsdli:wSDLLocation attribute specified, that the value of the attribute points to a WSDL 1.1 document
Artifacts	BWS_2016_TestCase.java Test_BWS_2016.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2016 sca-variables.dtd in contribution BWS_2016
Expected output	Negative test: "exception"

2.1.17 BWS_2017_TestCase

Testcase ID	BWS_2017_TestCase
Test Assertion	BWS-TA-20018

Description	Tests that where a <binding.ws/> element has a @uri attribute referencing the target service it does not also have a wsa:EndpointReference subelement referencing the target service
Artifacts	BWS_2017_TestCase.java Test_BWS_2017.composite TestInvocation.wsdl TestClient_0002.composite TestComposite9.composite TestComposite4.composite Service1.wsdl Service1Superset.wsdl in contribution BWS_2017 sca-variables.dtd in contribution BWS_2017
Expected output	Negative test: "exception"

2.1.18 BWS_2018_TestCase

Testcase ID	BWS_2018_TestCase
Test Assertion	BWS-TA-20019
Description	Tests that where a <binding.ws/> element is a subelement of a <service> <callback> element, it has no @uri attribute specified
Artifacts	BWS_2018_TestCase.java Test_BWS_2018.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite TestComposite55.composite Service1.wsdl Service5.wsdl Service5Callback.wsdl
Expected output	Negative test: "exception"

2.1.19 BWS_2019_TestCase

Testcase ID	BWS_2019_TestCase
Test Assertion	BWS-TA-20023
Description	Tests that a <binding.ws/> element conforms to the schema defined in sca-binding-webservice-1.1.xsd

Artifacts	BWS_2019_TestCase.java Test_BWS_2019.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

2.1.20 BWS_2020_TestCase

Testcase ID	BWS_2020_TestCase
Test Assertion	BWS-TA-20024
Description	Tests that where a <binding.ws/> subelement of a component <reference/> does not have a target defined, that the SCA runtime raises an error
Artifacts	BWS_2020_TestCase.java Test_BWS_2020.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

2.1.21 BWS_2021_TestCase

Testcase ID	BWS_2021_TestCase
Test Assertion	BWS-TA-20025
Description	Tests that where a <binding.ws/> subelement of a component <reference/> has a valid target defined by means of its @uri attribute, that the SCA runtime uses that target address when the component invokes an operation of the reference
Artifacts	BWS_2021_TestCase.java Test_BWS_2021.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite

	TestComposite4.composite Service1.wsdl
Expected output	Positive test: "BWS_2021 request service1 operation1 invoked service2 operation1 invoked"

2.1.22 BWS_2022_TestCase

Testcase ID	BWS_2022_TestCase
Test Assertion	BWS-TA-20030
Description	Tests that where a <binding.ws/> subelement of a component <reference/> has a @wsdlElement attribute which references a Port element of a WSDL document and the Port element has an associated portType which contains an sca @callback extension attribute, but where the <reference/> interface declaration does not have a callback, that the portType is treated as being not compatible with the interface for the reference and that an error is raised.
Artifacts	BWS_2022_TestCase.java Test_BWS_2022.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl Service1WithCallback.wsdl
Expected output	Negative test: "exception"

2.1.23 BWS_2023_TestCase

Testcase ID	BWS_2023_TestCase
Test Assertion	BWS-TA-20033
Description	Tests that the <bindingType/> for binding.ws binding includes the SOAP.v1_1 intent in its @mayProvides or @alwaysProvides attributes
Artifacts	BWS_2023_TestCase.java Test_BWS_2023.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl

Expected output	Positive test: "BWS_2023 request service1 operation1 invoked service2 operation1 invoked"
-----------------	--

2.2 [TestCases for Web Service Binding Specification Section 4](#)

2.2.1 BWS_4001_TestCase

Testcase ID	BWS_4001_TestCase
Test Assertion	BWS-TA-40001
Description	Tests that where a <component/> <service/> element has a <binding.ws/> subelement to which the SOAP intent is applied, that the service endpoint both sends and receives messages in a version of the SOAP protocol
Artifacts	BWS_4001_TestCase.java Test_BWS_4001.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite1.composite
Expected output	Positive test: "BWS_4001 request service1 operation1"

2.2.2 BWS_4002_TestCase

Testcase ID	BWS_4002_TestCase
Test Assertion	BWS-TA-40002
Description	Tests that where a <component/> <service/> element has a <binding.ws/> subelement to which the SOAP.v1_1 intent is applied, that the service endpoint both sends and receives messages in the SOAP v1.1 protocol
Artifacts	BWS_4002_TestCase.java Test_BWS_4002.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite1.composite
Expected output	Positive test: "BWS_4002 request service1 operation1"

2.2.3 BWS_4003_TestCase

Testcase ID	BWS_4003_TestCase
Test Assertion	BWS-TA-40003

Description	Tests that where a <component/> <service/> element has a <binding.ws/> subelement to which the SOAP.V1.2 intent is applied, that the service end-point both sends and receives messages in the SOAP v1.2 protocol
Artifacts	BWS_4003_TestCase.java Test_BWS_4003.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite1.composite
Expected output	Positive test: "BWS_4003 request service1 operation1"

2.2.4 BWS_4004_TestCase

Testcase ID	BWS_4004_TestCase
Test Assertion	BWS-TA-40001
Description	Tests that where a <component/> <reference/> element has a <binding.ws/> subelement to which the SOAP intent is applied, that the reference both sends and receives messages in a version of the SOAP protocol
Artifacts	BWS_4004_TestCase.java Test_BWS_4004.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite4.composite
Expected output	Positive test: "BWS_4004 request service1 operation1 invoked request Client service1 operation1 invoked"

2.2.5 BWS_4005_TestCase

Testcase ID	BWS_4005_TestCase
Test Assertion	BWS-TA-40002
Description	Tests that where a <component/> <reference/> element has a <binding.ws/> subelement to which the SOAP.v1_1 intent is applied, that the reference both sends and receives messages in the SOAP v1.1 protocol
Artifacts	BWS_4005_TestCase.java Test_BWS_4005.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite4.composite

Expected output	Positive test: "BWS_4005 request service1 operation1 invoked request Client service1 operation1 invoked"
-----------------	---

2.2.6 BWS_4006_TestCase

Testcase ID	BWS_4006_TestCase
Test Assertion	BWS-TA-40003
Description	Tests that where a <component/> <reference/> element has a <binding.ws/> subelement to which the SOAP.V1.2 intent is applied, that the reference both sends and receives messages in the SOAP v1.2 protocol
Artifacts	BWS_4006_TestCase.java Test_BWS_4006.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite4.composite
Expected output	Positive test: "BWS_4006 request service1 operation1 invoked request Client service1 operation1 invoked"

2.2.7 BWS_4007_TestCase

Testcase ID	BWS_4007_TestCase
Test Assertion	BWS-TA-40004
Description	Tests that where a <component/> <reference/> element has an explicit portType rpc-literal in form, it is accepted
Artifacts	BWS_4007_TestCase.java Test_BWS_4007.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite4.composite Service1RPCLiteral.wsdl
Expected output	Positive test: "BWS_4007 request service1 operation1 invoked request Client service1 operation1 invoked"
Comment	All other (positive) testcases use a simple document literal wrapped form and so the acceptance of rpc-literal is otherwise not tested

2.2.8 BWS_4008_TestCase

Testcase ID	BWS_4008_TestCase
Test Assertion	BWS-TA-40004
Description	Tests that where <component/> <reference/> element has an explicit port-Type rpc-encoded in form, it is rejected
Artifacts	BWS_4008_TestCase.java Test_BWS_4008.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl TestComposite4.composite Service1RPCEncoded.wsdl
Expected output	Negative test: "exception"

2.3 [TestCases for Web Service Binding Specification Section 5](#)

2.3.1 BWS_5001_TestCase

Testcase ID	BWS_5001_TestCase
Test Assertion	BWS-TA-50001
Description	Tests that where an SCA reference with a bidirectional interface invokes a Web service, that the Web service message sent to the target service contains a Callback EPR, either in the wsa:From SOAP header or in the wsa:ReplyTo header In this case, an SCA component is a client to an external Web service that has a callback, so this checks the ability of the binding.ws implementation of the runtime to transmit a callback address in the right way and host the callback service for a reference
Artifacts	BWS_5001_TestCase.java Test_BWS_5001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5001 request service1 operation1 invoked request Client service5 operation1 invoked service1 callback1 invoked"

2.3.2 BWS_5002_TestCase

Testcase ID	BWS_5002_TestCase
Test Assertion	BWS-TA-50001 BWS-TA-50003
Description	Tests that where an SCA service with a bidirectional interface is invoked by a Web services client, that the Web service message sent to the target service contains a Callback EPR, either in the wsa:From SOAP header or in the wsa:ReplyTo header, which the service is able to use to invoke an operation on the callback interface In this case, an SCA component provides a service over binding.ws where the service has a callback and an external client invokes the service and hosts the callback service.
Artifacts	BWS_5002_TestCase.java Test_BWS_5002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5002 service1 operation1 invoked callback service5Callback callback1 invoked"

2.3.3 BWS_5003_TestCase

Testcase ID	BWS_5003_TestCase
Test Assertion	BWS-TA-50002
Description	Tests that where an SCA service with a bidirectional interface is invoked by a Web services client, that if the Web service invocation message sent to the target service contains a Callback EPR, of the form "http://www.w3.org/2005/08/addressing/anonymous" that the service raises an Invalid Addressing Header fault.
Artifacts	BWS_5003_TestCase.java Test_BWS_5003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl jaxwsClient.Service5Impl.java

Expected output	Negative test: "exception"
-----------------	-------------------------------

2.3.4 BWS_5004_TestCase

Testcase ID	BWS_5004_TestCase
Test Assertion	BWS-TA-50004 BWS-TA-50007
Description	Tests that where an SCA service with a bidirectional interface is invoked by a Web services client, that a callback operation invocation follows the rules defined in Section 3.3 of the WS-Addressing 1.0 - Core specification - deals with case where there are no reference parameters on forward call
Artifacts	BWS_5004_TestCase.java Test_BWS_5004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5004 service1 operation1 invoked callback service5Callback callback1 invoked headers OK"

2.3.5 BWS_5005_TestCase

Testcase ID	BWS_5005_TestCase
Test Assertion	BWS-TA-50005 BWS-TA-50006
Description	Tests that where an SCA service with a bidirectional interface is invoked by a Web services client, and the client invocation message contains a wsa:MessageID SOAP header block, that an invocation of a callback operation contains a wsa:RelatesTo SOAP header block
Artifacts	BWS_5005_TestCase.java Test_BWS_5005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl

	jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5005 service1 operation1 invoked callback service5Callback callback1 invoked headers OK"

2.3.6 BWS_5006_TestCase

Testcase ID	BWS_5006_TestCase
Test Assertion	BWS-TA-50004
Description	Tests that where an SCA service with a bidirectional interface is invoked by a Web services client, that a callback operation invocation follows the rules defined in Section 3.3 of the WS-Addressing 1.0 - Core specification - deals with case where there are reference parameters on forward call
Artifacts	BWS_5006_TestCase.java Test_BWS_5006.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5006 service1 operation1 invoked callback service5Callback callback1 invoked headers OK"

3 Cross Mapping of Test Assertions to TestCases

Test Assertion	Test Cases
BWS-TA-20001	BWS_2001_TestCase
BWS-TA-20002	BWS_2002_TestCase
BWS-TA-20003	BWS_2003_TestCase
BWS-TA-20004	BWS_2004_TestCase
BWS-TA-20005	BWS_2005_TestCase
BWS-TA-20006	BWS_2006_TestCase
BWS-TA-20007	BWS_2007_TestCase
BWS-TA-20008	BWS_2008_TestCase
BWS-TA-20009	BWS_2009_TestCase

BWS-TA-20010	BWS_2005_TestCase
BWS-TA-20011	BWS_2010_TestCase
BWS-TA-20012	BWS_2011_TestCase
BWS-TA-20013	BWS_2012_TestCase
BWS-TA-20014	BWS_2013_TestCase
BWS-TA-20015	BWS_2012_TestCase BWS_2014_TestCase
BWS-TA-20016	BWS_2015_TestCase
BWS-TA-20017	BWS_2016_TestCase
BWS-TA-20018	BWS_2017_TestCase
BWS-TA-20019	BWS_2018_TestCase
BWS-TA-20020	Covered by a series of other testcases that use one or more of each of the attributes of the <binding.ws> element
BWS-TA-20021	Optional function — no test
BWS-TA-20022	Optional function — no test
BWS-TA-20023	BWS_2010_TestCase
BWS-TA-20024	BWS_2020_TestCase
BWS-TA-20025	BWS_2024_TestCase
BWS-TA-20026	Untestable as this requires a specific non-WSDL interface type, and there is no mandatory support for such an interface type
BWS-TA-20027	Untestable as this requires a specific non-WSDL interface type, and there is no mandatory support for such an interface type
BWS-TA-20028	Optional function — no test
BWS-TA-20029	Optional function — no test
BWS-TA-20030	BWS_2022_TestCase
BWS-TA-20031	BWS_2012_TestCase (depends on the SOAP 1.1 binding being supported)
BWS-TA-20032	Optional function — no test
BWS-TA-20033	BWS_2023_TestCase
BWS-TA-20034	Optional function — no test
BWS-TA-20035	Untestable — there is no standard way of producing a conflict between attached intent(s) and the configuration of a <binding.ws> element

Test Assertion	Test Cases
BWS-TA-40004	BWS_4001_TestCase BWS_4004_TestCase

BWS TA 40002	BWS_4002_TestCase BWS_4005_TestCase
BWS TA 40003	BWS_4003_TestCase BWS_4006_TestCase
BWS TA 40004	BWS_4007_TestCase BWS_4008_TestCase
BWS TA 40005	Untestable — or test not required since the assumption behind the TA pervades many other tests in the test suite
BWS TA 40007	Optional function — no test

Test Assertion	Test Cases
BWS-TA-50001	BWS_5001_TestCase BWS_5002_TestCase
BWS-TA-50002	BWS_5003_TestCase
BWS-TA-50003	BWS_5002_TestCase
BWS-TA-50004	BWS_5004_TestCase BWS_5006_TestCase
BWS-TA-50005	BWS_5005_TestCase
BWS-TA-50006	BWS_5005_TestCase
BWS-TA-50007	BWS_5004_TestCase
BWS-TA-50008	Untestable as it requires the support of some other callback protocol
BWS-TA-50009	SCA runtimes are not required to support or under- stand WS Policy — untestable
BWS-TA-50010	Optional function — no test
BWS-TA-50011	Optional function — no test

43 Catalog of Test Artifacts

4.13.1 Composite Files - lower level

Name	Valid	Description
TestComposite1 <i>"Impl Specific"</i>	Y	1 service, interface Service1 0 references service1Impl
TestComposite4 <i>"Impl Specific"</i>	Y	1 service, interface Service1 1 reference (1..1), interface Service1 service1Impl2
TestComposite9 <i>"Impl Specific"</i>	Y	1 service, interface = Service1Superset 0 references service1SupersetImpl
TestComposite54 <i>"Impl Specific"</i>	Y	1 service, interface = Service1 1 reference with a callback - Service5 & Service5Callback service1Callback5Impl
TestComposite55 <i>"Impl Specific"</i>	Y	1 service with callback - Service5 & Service5Callback 0 references service5Impl
TestClient_0002 <i>"Impl Specific"</i>	Y	Test client invocation composite 1 service, interface = TestInvocation 1 reference (1..1), interface = Service1

4.23.2 Java Interfaces

Name	Description
Service1.java	
Service1Superset.java	
Service5.java	
Service5Callback.java	

TestInvocation.java

4.33.3 Java Implementation Classes

Name	Description
Service1Impl.java	
Service1Impl2.java	
Service1SupersetImpl.java	
Service1Callback5Impl.java	
Service5Impl.java	
TestException.java	
ASM_0002_Client.java	

4.43.4 WSDL Interface Files

Name	Description
Service1.wsdl	Service1 interface 1 operation: - "operation1" string input, string output
Service1RPCLiteral.wsdl	Service1 interface in RPC Literal form
Service1Superset.wsdl	Service1Superset interface - superset of Service1 2 operations: - "operation1" string input, string output - "operation2" string input, string output
Service5.wsdl	Service5 interface which has a callback interface Service5Callback 1 operation: - "operation1" string input, string output
Service5Callback.wsdl	Service5Callback interface, which is the callback interface for Service5

	<p>1 operation:</p> <ul style="list-style-type: none"> - "callback1" <p>string input, string output</p>
Service1InvalidService.wsdl (BWS_2005 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - this is Service1 with an invalid WSDL Binding (invalid transport)
Service1InvalidPort.wsdl (BWS_2007 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - this is Service1 with a PortType incompatible with the regular Service1 PortType
Service1WithPort.wsdl (BWS_2008 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - this is Service1 with a Port which includes a target address for the service
Service1Superset.wsdl (BWS_2009 contribution)	<p>Service1Superset interface</p> <ul style="list-style-type: none"> - with additional WSDL Service, Port & Binding elements
Service1WithBinding.wsdl (BWS_2010 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - with a valid WSDL Binding element
Service1InvalidService.wsdl (BWS_2011 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - this is Service1 with a WSDL Service made invalid through the use of an invalid WSDL Binding element
Service1Superset.wsdl (BWS_2012 contribution)	<p>Service1Superset interface</p> <ul style="list-style-type: none"> - with additional WSDL Service, Port & Binding elements
Service1InvalidService.wsdl (BWS_2013 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - this is Service1 with a WSDL Service made invalid through the use of an invalid WSDL Binding element
Service1Superset.wsdl (BWS_2014 contribution)	<p>Service1Superset interface</p> <ul style="list-style-type: none"> - with additional WSDL Service, Port & Binding elements
Service1WithBinding.wsdl (BWS_2015 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - with a valid WSDL Binding element
Service1WithBinding.wsdl (BWS_2016 contribution)	<p>Service1 interface</p> <ul style="list-style-type: none"> - with a valid WSDL Binding element
Service1Superset.wsdl (BWS_2017 contribution)	<p>Service1Superset interface</p> <ul style="list-style-type: none"> - with additional WSDL Service, Port & Binding elements

Service1WithCallback.wsdl (BWS_2022 contribution)	Service1 interface - PortType has a sca:callback attribute added which points to a callback interface Service1Callback, which is contained in the same WSDL document
Service1RPCEncoded.wsdl (BWS_4008 contribution)	Service1 interface in RPC Encoded form
sca_variables.dtd (BWS_General contribution)	DTD holding service addresses which can be modified to suit an SCA runtime
sca_variables.dtd (BWS_2005 contribution)	holds service address for <i>TEST_BWS_2005Component2/Service1</i>
sca_variables.dtd (BWS_2008 contribution)	holds service address for <i>TEST_BWS_2008Component2/Service1Special</i>
sca_variables.dtd (BWS_2009 contribution)	holds service address for <i>TEST_BWS_2009Component2/Service1Superset</i>
sca_variables.dtd (BWS_2011 contribution)	holds service address for <i>TEST_BWS_2011Component2/Service1</i>
sca_variables.dtd (BWS_2012 contribution)	holds service address for <i>TEST_BWS_2012Component2/Service1Superset</i>
sca_variables.dtd (BWS_2013 contribution)	holds service address for <i>TEST_BWS_2013Component2/Service1</i>
sca_variables.dtd (BWS_2014 contribution)	holds service address for <i>TEST_BWS_2014Component2/Service1Superset</i>
sca_variables.dtd (BWS_2015 contribution)	holds service address for <i>TEST_BWS_2015Component2/Service1</i>
sca_variables.dtd (BWS_2016 contribution)	holds service address for <i>TEST_BWS_2016Component2/Service1</i>
sca_variables.dtd (BWS_2017 contribution)	holds service address for <i>TEST_BWS_2017Component2/Service1Superset</i>
sca_variables.dtd (BWS_2021 contribution)	holds service address for

	<i>TEST_BWS_2021Component2/Service1</i>
sca_variables.dtd (BWS_2022 contribution)	holds service address for <i>TEST_BWS_2022/Component2/Service1</i>
sca_variables.dtd (BWS_2023 contribution)	holds service address for <i>TEST_BWS_2023Component2/Service1</i>
sca_variables.dtd (BWS_4008 contribution)	holds service address for service exposed by JAXWS test runner client application
TestClient.wsdl	TestClient interface used to invoke SCA services
TestInvocation.wsdl	TestInvocation interface supplied by SCA component to allow client application to invoke test artifacts 1 operation: - "invokeTest" string input, string output

54 Conformance

There

The artifacts contained in the [sca-wsbinding-1.1-testcases.zip](#) are ~~re~~considered to be authoritative and take precedence over the artifacts described in this document.

This specification defines two targets for conformance:

a) [WS Binding Testcases](#)

b) [Web Service Callback \(WSCB\) Testcases](#)

An implementation that claims to conform to the requirements of the [WS Binding Testcases](#) **MUST** be able to run all test cases in Section 2.1 and Section 2.2, producing the 'Expected Output'.

An implementation that claims to conform to the [WSCB Testcases](#) **MUST** be able to run all tests in Section 2.3, producing the 'Expected Output'.

Appendix A. Test Assertions for the Web Service Binding

This section defines the Test Assertions for the SCA Web Service Binding specification, following the format defined in the [OASIS Test Assertion Guidelines](#) specification [**TA-GUIDE**].

A.1. Example Test Assertion

Test assertions are presented in a tabular format with rows corresponding to the entry types defined in the [OASIS Test Assertions Guidelines](#)

Assertion ID	BJM-TA-nnnnn
	[BJMnnnnn]
Target	<xyuvbghs/> element of composite file
Prerequisites	The [<xyuvbghs/> element] has a @foobar attribute
Predicate	The @foobar attribute of [the <xyuvbghs/> element] is a URI that references a foobar element in the SCA Domain-
Prescription Level	Mandatory
Tags	foobar references

Assertion ID: Is a unique ID for the test assertion. Its format starts with a 3 letter string that identifies the specification to which it relates - "BJM" for the SCA JMS Binding specification. This is followed by "-TA-" to indicate that this identifier is for a test assertion. This is then followed by a unique 5 digit number

Source: Is the identifier(s) of the normative statement(s) in the specification to which this assertion relates.

Target: Identifies the target which is addressed by this assertion. This is typically some SCA document element, but possibly could identify an SCA runtime and its behaviour.

Prerequisites: Defines any prerequisites for this test assertion. The prerequisites can be defined in terms of one or more other test assertions that have to be true.

Predicate: The meat of the assertion - something that evaluates to true or false for the given target.

Prescription Level: Mandatory (for "MUST" requirements) or Preferred (for "SHOULD" requirements) or Permitted (for "MAY" requirements).

Tags: Zero or more labels to be attached to this test assertion - these tags can be used to group sets of assertions.

A.2. Test Assertions for Web Service Binding Specification Section 2

<u>Assertion ID</u>	<u>BWS-TA-20001</u>
<u>Source</u>	<u>[BWS20001]</u>
<u>Target</u>	<u>The reference/binding.ws/@uri attribute</u>
<u>Prerequisites</u>	
<u>Predicate</u>	<u>The value of the attribute is an absolute value</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“uri attribute” “reference target”</u>

<u>Assertion ID</u>	<u>BWS-TA-20002</u>
<u>Source</u>	<u>[BWS20002]</u>
<u>Target</u>	<u>The binding.ws/@wsdlElement attribute</u>
<u>Prerequisites</u>	
<u>Predicate</u>	<u>The value of the attribute points to an existing WSDL 1.1 element</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“wsdlElement”</u>

<u>Assertion ID</u>	<u>BWS-TA-20003</u>
<u>Source</u>	<u>[BWS20003]</u>
<u>Target</u>	<u>The service/binding.ws/@wsdlElement attribute</u>
<u>Prerequisites</u>	
<u>Predicate</u>	<u>The value of the attribute is not of the form <WSDL-namespace-URI>#wsdl.service(<service-name>)</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“wsdlElement”</u>

<u>Assertion ID</u>	<u>BWS-TA-20004</u>
<u>Source</u>	<u>[BWS20004]</u>
<u>Target</u>	<u>The reference/binding.ws/@wsdlElement attribute</u>
<u>Prerequisites</u>	<u>The value of the attribute is of the form <WSDL-namespace-URI>#wsdl.service(<service-name>)</u>
<u>Predicate</u>	<u>The set of available ports for that reference-binding is non-empty</u>
<u>Prescription</u>	<u>Mandatory</u>

Level	
Tags	"wsdlElement" "reference target" "port"
Assertion ID	BWS-TA-20005
Source	[BWS20005]
Target	SCA runtime
Prerequisites	The value of the attribute reference/binding.ws/@wsdlElement is of the form <WSDL-namespace-URI>#wsdl.service(<service-name>) and SCA runtime does not support any of the available ports
Predicate	The SCA runtime raises an error
Prescription Level	Mandatory
Tags	"wsdlElement" "reference target" "error"
Assertion ID	BWS-TA-20006
Source	[BWS20006]
Target	SCA runtime
Prerequisites	The value of the attribute reference/binding.ws/@wsdlElement is of the form <WSDL-namespace-URI>#wsdl.service(<service-name>) and the number of available ports is more than one
Predicate	The SCA runtime uses exactly one port for each invocation
Prescription Level	Mandatory
Tags	"wsdlElement" "reference target"
Assertion ID	BWS-TA-20007
Source	[BWS20007]
Target	The service/binding.ws/@wsdlElement attribute
Prerequisites	The value of the attribute is of the form <WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)
Predicate	The portType associated with the port is compatible with the service interface and satisfies all the policy constraints of the binding
Prescription Level	Mandatory
Tags	"wsdlElement"
Assertion ID	BWS-TA-20008
Source	[BWS20008]

Target	SCA runtime
Prerequisites	The value of the attribute service/binding.ws/@wsdlElement is of the form <WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)
Predicate	The SCA runtime exposes the endpoint specified by the WSDL port or raises an error if the port is not supported
Prescription Level	Mandatory
Tags	"wsdlElement"

Assertion ID	BWS-TA-20009
Source	[BWS20009]
Target	The reference/binding.ws/@wsdlElement attribute
Prerequisites	The value of the attribute is of the form <WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)
Predicate	The portType associated with the port is compatible superset of the reference interface and satisfies all the policy constraints of the binding
Prescription Level	Mandatory
Tags	"wsdlElement"

Assertion ID	BWS-TA-20010
Source	[BWS20010]
Target	SCA runtime
Prerequisites	The value of the attribute reference/binding.ws/@wsdlElement is of the form <WSDL-namespace-URI>#wsdl.port(<service-name>/<port-name>)
Predicate	The SCA runtime uses the endpoint specified by the WSDL port for invocations or raises an error if the port is not supported
Prescription Level	Mandatory
Tags	"wsdlElement"

Assertion ID	BWS-TA-20011
Source	[BWS20011]
Target	The service/binding.ws/@wsdlElement attribute
Prerequisites	The value of the attribute is of the form <WSDL-namespace-URI>#wsdl.binding(<binding-name>)
Predicate	The portType associated with the WSDL binding is compatible with the service interface and satisfies all the policy constraints of the binding

Prescription Level	Mandatory
Tags	"wsdlElement"
Assertion ID	BWS-TA-20012
Source	[BWS20012]
Target	SCA runtime
Prerequisites	The value of the attribute service/binding.ws/@wsdlElement is of the form <WSDL-namespace-URI>#wsdl.binding(<binding-name>)
Predicate	The SCA runtime exposes an endpoint using the specified WSDL binding or raises an error if the WSDL binding is not supported
Prescription Level	Mandatory
Tags	"wsdlElement"
Assertion ID	BWS-TA-20013
Source	[BWS20013]
Target	The reference/binding.ws/@wsdlElement attribute
Prerequisites	The value of the attribute is of the form <WSDL-namespace-URI>#wsdl.binding(<binding-name>)
Predicate	The portType associated with the WSDL binding is compatible superset of the reference interface and satisfies all the policy constraints of the binding
Prescription Level	Mandatory
Tags	"wsdlElement"
Assertion ID	BWS-TA-20014
Source	[BWS20014]
Target	SCA runtime
Prerequisites	The value of the attribute reference/binding.ws/@wsdlElement is of the form <WSDL-namespace-URI>#wsdl.binding(<binding-name>)
Predicate	The SCA runtime uses the WSDL binding specified for reference invocations or raises an error if the WSDL binding is not supported
Prescription Level	Mandatory
Tags	"wsdlElement"
Assertion ID	BWS-TA-20015

<u>Source</u>	[BWS20015]
<u>Target</u>	The reference/binding.ws/@wsdlElement attribute
<u>Prerequisites</u>	The value of the attribute is of the form <WSDL-namespace-URI>#wsdl.binding(<binding-name>)
<u>Predicate</u>	The endpoint address is either specified by the reference/binding.ws/@uri attribute, or the reference/binding.ws/EndpointReference element, except for the cases where the SCA Assembly specification allows the @uri attribute to be omitted
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	“wsdlElement”

<u>Assertion ID</u>	BWS-TA-20016
<u>Source</u>	[BWS20017]
<u>Target</u>	The binding.ws element
<u>Prerequisites</u>	The attribute wsdl:wsdlLocation attribute is present
<u>Predicate</u>	The attribute wsdlElement is also present on the same binding.ws element
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	“wsdlElement” “wsdlLocation”

<u>Assertion ID</u>	BWS-TA-20017
<u>Source</u>	[BWS20018]
<u>Target</u>	The binding.ws/@wsdl:wsdlLocation attribute
<u>Prerequisites</u>	
<u>Predicate</u>	The attribute value points to an existing WSDL 1.1 document
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	“wsdlLocation”

<u>Assertion ID</u>	BWS-TA-20018
<u>Source</u>	[BWS20019]
<u>Target</u>	The binding.ws element
<u>Prerequisites</u>	
<u>Predicate</u>	The uri attribute, the endpointReference element and the wsdlElement attribute referring to a WSDL port or service are mutually exclusive
<u>Prescription</u>	Mandatory

Level	
Tags	"wsdlElement" "uri" "endpoint reference"
Assertion ID	BWS-TA-20019
Source	[BWS20020]
Target	The service/callback/binding.ws element
Prerequisites	
Predicate	The uri attribute or endpointReference element are absent
Prescription Level	Mandatory
Tags	"callback" "uri" "endpoint reference"
Assertion ID	BWS-TA-20020
Source	[BWS20021]
Target	SCA runtime
Prerequisites	
Predicate	All attributes of binding.ws, i.e., name, uri, requires, policSets, wsdlElement, wsdl:wsdlLocation are supported
Prescription Level	Mandatory
Tags	"wsdlElement" "uri" "requires" "policySets" "name" "wsdlLocation"
Assertion ID	BWS-TA-20021
Source	[BWS20022]
Target	SCA runtime
Prerequisites	
Predicate	binding.ws/endpointReference element is supported
Prescription Level	Mandatory
Tags	"endpointReference" "SCA runtime"
Assertion ID	BWS-TA-20023
Source	[BWS20024]
Target	binding.ws element
Prerequisites	

Predicate	Conforms to the schema defined in sca-binding-webservice-1.1.xsd
Prescription Level	Mandatory
Tags	"schema"
Assertion ID	BWS-TA-20024
Source	[BWS20025]
Target	<binding.ws/> subelement of a component <reference> element
Prerequisites	binding.ws does not contain a target address
Predicate	SCA runtime raises an error
Prescription Level	Mandatory
Tags	"target address" "SCA runtime"
Assertion ID	BWS-TA-20025
Source	[BWS20026]
Target	<binding.ws/> subelement of a component <reference/>
Prerequisites	The binding.ws element has a valid target specified by one of the mechanisms defined in the specification section "Endpoint URI resolution"
Predicate	The target address is used by the runtime when the component implementation invokes an operation of the reference
Prescription Level	Mandatory
Tags	"target address"
Assertion ID	BWS-TA-20026
Source	[BWS20027]
Target	SCA runtime
Prerequisites	binding.ws is used on a service or a reference that has an interface which is not defined using interface.wsdl
Predicate	A WSDL portType is derived for the service or the reference using the WSDL mapping rules defined for that interface type
Prescription Level	Mandatory
Tags	"sca runtime" "interface mapping" "Language dependent"
Assertion ID	BWS-TA-20027

Source	[BWS20028]
Target	SCA runtime
Prerequisites	binding.ws is used on a service or a reference that has an interface which is not defined using interface.wsdl
Predicate	An error is raised if the interface used cannot be mapped to a WSDL port-Type
Prescription Level	Mandatory
Tags	"sca runtime" "interface mapping"

Assertion ID	BWS-TA-20030
Source	[BWS20032]
Target	<binding.ws/> element which references a WSDL document by means of its @wsdlElement attribute
Prerequisites	The referenced WSDL document contains one or more of the extensions defined in the SCA namespace "http://docs.oasis-open.org/ns/opencsa/sca/200912" as defined in the SCA Assembly specification
Predicate	The SCA runtime supports the SCA WSDL extensions contained in the WSDL document
Prescription Level	Mandatory
Tags	"sca runtime" "WSDL extension"

Assertion ID	BWS-TA-20031
Source	[BWS20033]
Target	<binding.ws/> element which references a Binding in a WSDL document by means of its @wsdlElement attribute
Prerequisites	The WSDL Binding element uses the SOAP 1.1 over HTTP binding, using a wssoap11:binding that has the @transport attribute with a value of "http://schemas.xmlsoap.org/soap/http"
Predicate	The binding.ws element is supported by the SCA runtime and it is possible to invoke a service operation successfully through the reference or service which uses the binding.ws element.
Prescription Level	Mandatory
Tags	"sca runtime" "WSDL extension" "SOAP 1.1"

Assertion ID	BWS-TA-20032
Source	[BWS20034]

<u>Target</u>	<binding.ws/> element which references a Binding in a WSDL document by means of its @wsdlElement attribute
<u>Prerequisites</u>	The WSDL Binding element uses the SOAP 1.2 over HTTP binding, using a wsoap12:binding that has the @transport attribute with a value of "http://schemas.xmlsoap.org/soap/http"
<u>Predicate</u>	The binding.ws element is supported by the SCA runtime and it is possible to invoke a service operation successfully through the reference or service which uses the binding.ws element.
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	"sca runtime" "WSDL extension" "SOAP 1.2"

<u>Assertion ID</u>	BWS-TA-20033
<u>Source</u>	[BWS20035]
<u>Target</u>	<bindingType> element associated with binding.ws
<u>Prerequisites</u>	
<u>Predicate</u>	The <bindingType> element associated with binding.ws includes the SOAP.v1_1 intent in its @mayProvides or @alwaysProvides attributes
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	"SOAP 1.1" "bindingType" "intent" "mayProvides" "alwaysProvides"

<u>Assertion ID</u>	BWS-TA-20034
<u>Source</u>	[BWS20036]
<u>Target</u>	<bindingType> element associated with binding.ws
<u>Prerequisites</u>	
<u>Predicate</u>	The <bindingType> element associated with binding.ws includes the SOAP.v1_2 intent in its @mayProvides attribute
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	"SOAP 1.2" "bindingType" "intent" "mayProvides"

<u>Assertion ID</u>	BWS-TA-20035
<u>Source</u>	[BWS20037]
<u>Target</u>	<binding.ws/> element configured with a policy intent
<u>Prerequisites</u>	The policy intent conflicts with the configuration of the <binding.ws/> element
<u>Predicate</u>	SCA runtime raises an error

Prescription Level	Mandatory
Tags	"SCA runtime" "intent"

A.3. Test Assertions for Web Service Binding Specification Section 4

Assertion ID	BWS-TA-40001
Source	[BWS40001]
Target	Component <service/> or <reference/> with a <binding.ws/> subelement
Prerequisites	SOAP intent is applied to the <binding.ws/> element
Predicate	The <service/> or <reference/> transmits and receives messages using one or more versions of SOAP
Prescription Level	Mandatory
Tags	"SCA runtime" "intent" "SOAP"

Assertion ID	BWS-TA-40002
Source	[BWS40002]
Target	SCA runtime
Prerequisites	SOAP.v1_1 intent is required
Predicate	The SCA runtime transmits and receives messages using only SOAP 1.1
Prescription Level	Mandatory
Tags	"SCA runtime" "intent" "SOAP"

Assertion ID	BWS-TA-40003
Source	[BWS40003]
Target	SCA runtime
Prerequisites	SOAP.v1_2 intent is required
Predicate	The SCA runtime transmits and receives messages using only SOAP 1.2
Prescription Level	Mandatory
Tags	"SCA runtime" "intent" "SOAP"

Assertion ID	BWS-TA-40004
Source	[BWS40004]
Target	portType of an SCA component <service/> or <reference/> element

<u>Prerequisites</u>	<u>portType is either explicitly specified or is derived from the interface specified for the service or reference</u>
<u>Predicate</u>	<u>The portType follows either the rpc-literal pattern or the document-literal pattern</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“SCA runtime” “intent” “SOAP”</u>

<u>Assertion ID</u>	<u>BWS-TA-40005</u>
<u>Source</u>	<u>[BWS40005]</u>
<u>Target</u>	<u>SCA runtime</u>
<u>Prerequisites</u>	<u>The transport details are not otherwise determined</u>
<u>Predicate</u>	<u>The default transport binding rules are used</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“SCA runtime” “default transport binding rules”</u>

<u>Assertion ID</u>	<u>BWS-TA-40007</u>
<u>Source</u>	<u>[BWS40007]</u>
<u>Target</u>	<u>SCA runtime</u>
<u>Prerequisites</u>	<u>Default transport binding rules are used along with rpc-literal pattern</u>
<u>Predicate</u>	<u>SCA runtime uses the structural URI associated with the binding as the namespace of the child elements of the SOAP body element</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“SCA runtime” “default transport binding rules” “rpc-literal”</u>

A.4. Test Assertions for Web Service Binding Specification Section 5

<u>Assertion ID</u>	<u>BWS-TA-50001</u>
<u>Source</u>	<u>[BWS50002]</u>
<u>Target</u>	<u>Web service request message sent to a Web service endpoint</u>
<u>Prerequisites</u>	<u>a) Web service endpoint has an interface with an associated callback (ie. it is bidirectional)</u> <u>b) SCA Web Services Callback Protocol is used</u>
<u>Predicate</u>	<u>Request message contains a Callback EPR, either in the wsa:From SOAP header block or in the wsa:ReplyTo SOAP header block</u>
<u>Prescription Level</u>	<u>Mandatory</u>

<u>Tags</u>	"SCA runtime" "SCA Web Services Callback Protocol"
<u>Assertion ID</u>	BWS-TA-50002
<u>Source</u>	[BWS50004]
<u>Target</u>	SCA bidirectional service using the binding.ws binding with the SCA Web Services Callback Protocol
<u>Prerequisites</u>	The service receives an operation invocation of the service where the Callback EPR's address value is "http://www.w3.org/2005/08/addressing/anonymous" or "http://www.w3.org/2005/08/addressing/none"
<u>Predicate</u>	An Invalid Addressing Header fault is generated as specified in Section 6.4.1 of WS-Addressing 1.0 SOAP Binding
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"
<u>Assertion ID</u>	BWS-TA-50003
<u>Source</u>	[BWS50005]
<u>Target</u>	SCA runtime
<u>Prerequisites</u>	The callback interface is invoked using the SCA Web Services Callback Protocol
<u>Predicate</u>	The Callback EPR from a request message that invoked the forward interface is used to invoke the callback interface
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"
<u>Assertion ID</u>	BWS-TA-50004
<u>Source</u>	[BWS50006]
<u>Target</u>	SCA runtime
<u>Prerequisites</u>	The Callback EPR is selected to invoke the callback interface
<u>Predicate</u>	The rules defined in Section 3.3 of WS-Addressing 1.0 – Core are followed to invoke operations on the callback interface using the Callback EPR
<u>Prescription Level</u>	Mandatory
<u>Tags</u>	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"
<u>Assertion ID</u>	BWS-TA-50005

Source	[BWS50007]
Target	SCA runtime
Prerequisites	The request message from which the Callback EPR was obtained contained the wsa:MessageID SOAP header block
Predicate	The callback message contains the wsa:RelatesTo SOAP header block
Prescription Level	Mandatory
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"

Assertion ID	BWS-TA-50006
Source	[BWS50008]
Target	SCA runtime
Prerequisites	The request message from which the Callback EPR was obtained contained the wsa:MessageID SOAP header block
Predicate	The wsa:RelatesTo SOAP header block in the callback message has the relationship type value of "http://docs.oasis-open.org/opencsa/sca-bindings/ws/callback" and the related message id is the wsa:MessageID of the message from which the Callback EPR was obtained
Prescription Level	Mandatory
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"

Assertion ID	BWS-TA-50007
Source	[BWS50009]
Target	SCA runtime
Prerequisites	The request message from which the Callback EPR was obtained does not contain the wsa:MessageID SOAP header block
Predicate	The callback message does not contain the wsa:RelatesTo SOAP header block with a relationship type value of "http://docs.oasis-open.org/opencsa/sca-bindings/ws/callback"
Prescription Level	Mandatory
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"

Assertion ID	BWS-TA-50008
Source	[BWS50010]
Target	SCA runtime
Prerequisites	WSCallback policy assertion is present in the effective policy of a service/binding or a reference/binding

<u>Predicate</u>	<u>The SCA Web Services Callback Protocol is used to invoke callbacks for that binding</u>
<u>Prescription Level</u>	<u>Mandatory</u>
<u>Tags</u>	<u>“SCA runtime” “SCA Web Services Callback Protocol” “WSCallback policy assertion”</u>

Appendix B. Cross Mapping of Normative Statements to Assertions

This section contains a list of normative statements ~~relating to the~~ in the SCA Web Service Binding specification and the corresponding Test Assertions.

<u>Statement Identifier</u>	<u>Test Assertion</u>
<u>BWS20001</u>	<u>BWS-TA-20001</u>
<u>BWS20002</u>	<u>BWS-TA-20002</u>
<u>BWS20003</u>	<u>BWS-TA-20003</u>
<u>BWS20004</u>	<u>BWS-TA-20004</u>
<u>BWS20005</u>	<u>BWS-TA-20005</u>
<u>BWS20006</u>	<u>BWS-TA-20006</u>
<u>BWS20007</u>	<u>BWS-TA-20007</u>
<u>BWS20008</u>	<u>BWS-TA-20008</u>
<u>BWS20009</u>	<u>BWS-TA-20009</u>
<u>BWS20010</u>	<u>BWS-TA-20010</u>
<u>BWS20011</u>	<u>BWS-TA-20011</u>
<u>BWS20012</u>	<u>BWS-TA-20012</u>
<u>BWS20013</u>	<u>BWS-TA-20013</u>
<u>BWS20014</u>	<u>BWS-TA-20014</u>
<u>BWS20015</u>	<u>BWS-TA-20015</u>
<u>BWS20017</u>	<u>BWS-TA-20016</u>
<u>BWS20018</u>	<u>BWS-TA-20017</u>
<u>BWS20019</u>	<u>BWS-TA-20018</u>
<u>BWS20020</u>	<u>BWS-TA-20019</u>
<u>BWS20021</u>	<u>BWS-TA-20020</u>
<u>BWS20022</u>	<u>BWS-TA-20021</u>
<u>BWS20024</u>	<u>BWS-TA-20023</u>
<u>BWS20025</u>	<u>BWS-TA-20024</u>
<u>BWS20026</u>	<u>BWS-TA-20025</u>
<u>BWS20027</u>	<u>BWS-TA-20026</u>
<u>BWS20028</u>	<u>BWS-TA-20027</u>
<u>BWS20032</u>	<u>BWS-TA-20030</u>
<u>BWS20033</u>	<u>BWS-TA-20031</u>
<u>BWS20034</u>	<u>BWS-TA-20032</u>
<u>BWS20035</u>	<u>BWS-TA-20033</u>
<u>BWS20036</u>	<u>BWS-TA-20034</u>
<u>BWS20037</u>	<u>BWS-TA-20035</u>

<u>Statement Identifier</u>	<u>Test Assertion</u>
BWS40001	BWS-TA-40001
BWS40002	BWS-TA-40002
BWS40003	BWS-TA-40003
BWS40004	BWS-TA-40004
BWS40005	BWS-TA-40005
BWS40007	BWS-TA-40007

<u>Statement Identifier</u>	<u>Test Assertion</u>
BWS50002	BWS-TA-50001
BWS50004	BWS-TA-50002
BWS50005	BWS-TA-50003
BWS50006	BWS-TA-50004
BWS50007	BWS-TA-50005
BWS50008	BWS-TA-50006
BWS50009	BWS-TA-50007
BWS50010	BWS-TA-50008
BWS50013	BWS-TA-50009
BWS50014	BWS-TA-50010
BWS50015	BWS-TA-50011

Appendix A. Appendix C. Cross Mapping of Test Assertions to TestCases.

This section contains a list of Test Assertions and the corresponding TestCases.

<u>Test Assertion</u>	<u>Test Cases</u>
BWS-TA-20001	BWS_2001_TestCase
BWS-TA-20002	BWS_2002_TestCase
BWS-TA-20003	BWS_2003_TestCase
BWS-TA-20004	BWS_2004_TestCase
BWS-TA-20005	BWS_2005_TestCase
BWS-TA-20006	BWS_2006_TestCase
BWS-TA-20007	BWS_2007_TestCase
BWS-TA-20008	BWS_2008_TestCase
BWS-TA-20009	BWS_2009_TestCase
BWS-TA-20010	BWS_2005_TestCase
BWS-TA-20011	BWS_2010_TestCase
BWS-TA-20012	BWS_2011_TestCase
BWS-TA-20013	BWS_2012_TestCase
BWS-TA-20014	BWS_2013_TestCase
BWS-TA-20015	BWS_2012_TestCase BWS_2014_TestCase

BWS-TA-20016	BWS_2015_TestCase
BWS-TA-20017	BWS_2016_TestCase
BWS-TA-20018	BWS_2017_TestCase
BWS-TA-20019	BWS_2018_TestCase
BWS-TA-20020	Covered by a series of other testcases that use one or more of each of the attributes of the <binding.ws/> element
BWS-TA-20021	Untested
BWS-TA-20023	BWS_2019_TestCase
BWS-TA-20024	BWS_2020_TestCase
BWS-TA-20025	BWS_2021_TestCase
BWS-TA-20026	Language Dependent
BWS-TA-20027	Untestable as this requires a specific non-WSDL interface type, and there is no mandatory support for such an interface type
BWS-TA-20029	Untestable
BWS-TA-20030	BWS_2022_TestCase
BWS-TA-20031	BWS_2012_TestCase (depends on the SOAP 1.1 binding being supported)
BWS-TA-20032	Untested
BWS-TA-20033	BWS_2023_TestCase
BWS-TA-20034	Untested
BWS-TA-20035	Untestable - there is no standard way of producing a conflict between attached intent(s) and the configuration of a <binding.ws/> element

Test Assertion	Test Cases
BWS-TA-40001	BWS_4001_TestCase BWS_4004_TestCase
BWS-TA-40002	BWS_4002_TestCase BWS_4005_TestCase
BWS-TA-40003	BWS_4003_TestCase BWS_4006_TestCase
BWS-TA-40004	BWS_4007_TestCase BWS_4008_TestCase
BWS-TA-40005	Untestable - or test not required since the assumption behind the TA pervades many other tests in the test suite
BWS-TA-40007	Untested

<u>Test Assertion</u>	<u>Test Cases</u>
BWS-TA-50001	BWS_5001_TestCase BWS_5002_TestCase
BWS-TA-50002	BWS_5003_TestCase
BWS-TA-50003	BWS_5002_TestCase
BWS-TA-50004	BWS_5004_TestCase BWS_5006_TestCase
BWS-TA-50005	BWS_5005_TestCase
BWS-TA-50006	BWS_5005_TestCase
BWS-TA-50007	BWS_5004_TestCase
BWS-TA-50008	Untestable as it requires the support of some other callback protocol
BWS-TA-50009	SCA runtimes are not required to support or understand WS Policy - untestable
BWS-TA-50010	Untested
BWS-TA-50011	Untested

Appendix B. Appendix D. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged

Participants:

Participant Name	Affiliation
Mike Edwards	IBM
Eric Johnson	TIBCO
Anish Karmarkar	Oracle
Simon Holdsworth	IBM

Appendix C. Appendix E. Revision History

Revision	Date	Editor	Changes Made
wd01	05/13/05/10	Mike Edwards	Initial version

wd02	05/17/05/10	Mike Edwards	Added testcases BWS_2002 to BWS_2006
wd03	06/01/06/10	Mike Edwards	Added testcases BWS_2007 to BWS_2023
wd04	06/07/06/10	Mike Edwards	Editorial changes Added testcases BWS_4001 to BWS_4007
wd05	06/14/06/10	Mike Edwards	Modified BWS_5001 Added BWS_5002 to BWS_5006
cd01	07/01/07/10	Mike Edwards	All changes accepted Frontmatter edited to conform to OASIS requirements Completed artifact catalog in Section 4
csd01-rev1	04/01/10	Simon Holdsworth	Applied resolutions to issues BINDINGS-142, BINDINGS-146, BINDINGS-148
wd06	06/24/11	Simon Holdsworth	Applied resolutions to issues BINDINGS-151, BINDINGS-152, BINDINGS-153, BINDINGS-160, BINDINGS-161, BINDINGS-162, BINDINGS-163, BINDINGS-168
wd07	07/04/11	Simon Holdsworth	Editorial formatting updates