



TestCases for the SCA Web Service Binding Specification Version 1.1

Committee Specification Draft 01 revision 1 + Issue 152

1 April 2011

Specification URIs:

This Version:

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csd01-rev1.html>
<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csd01-rev1.odt>
[\(Authoritative\)](http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-csd01-rev1.pdf)

Previous Version:

<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>
[\(Authoritative\)](http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-cd01.pdf)

Latest Version:

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

Technical Committee:

OASIS Service Component Architecture / Bindings (SCA-Bindings) TC
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sca-bindings

Chair(s):

Simon Holdsworth, IBM

Editor(s):

Mike Edwards, IBM

Related Work:

This document is related to:

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

- 1 The Test Suite artifacts relating to this document can be found here:
2 • <http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-testcases-cd01.zip>

Declared XML Namespace(s):

<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 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" 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" 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
<http://www.oasis-open.org/committees/sca-bindings/>

Notices

Copyright © OASIS® 2010. 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 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.

3 Table of Contents

4	1 Introduction	6
5	1.1 TestCase Structure	6
6	1.1.1 Test Client	6
7	1.1.2 Test Application	6
8	1.1.3 Test Artifacts Organization	7
9	1.2 Namespaces and Java Package Names	8
10	1.2.1 SCA Artifact Namespaces	8
11	1.2.2 WSDL Namespace	8
12	1.2.3 Java Package name	8
13	1.3 Terminology	8
14	1.4 Normative References	8
15	2 TestCases	10
16	2.1 TestCases for Web Service Binding Specification Section 2	10
17	2.1.1 BWS_2001_TestCase	10
18	2.1.2 BWS_2002_TestCase	10
19	2.1.3 BWS_2003_TestCase	11
20	2.1.4 BWS_2004_TestCase	11
21	2.1.5 BWS_2005_TestCase	12
22	2.1.6 BWS_2006_TestCase	12
23	2.1.7 BWS_2007_TestCase	13
24	2.1.8 BWS_2008_TestCase	13
25	2.1.9 BWS_2009_TestCase	14
26	2.1.10 BWS_2010_TestCase	14
27	2.1.11 BWS_2011_TestCase	15
28	2.1.12 BWS_2012_TestCase	15
29	2.1.13 BWS_2013_TestCase	16
30	2.1.14 BWS_2014_TestCase	17
31	2.1.15 BWS_2015_TestCase	17
32	2.1.16 BWS_2016_TestCase	18
33	2.1.17 BWS_2017_TestCase	18
34	2.1.18 BWS_2018_TestCase	19
35	2.1.19 BWS_2019_TestCase	19
36	2.1.20 BWS_2020_TestCase	20
37	2.1.21 BWS_2021_TestCase	20
38	2.1.22 BWS_2022_TestCase	21
39	2.1.23 BWS_2023_TestCase	21
40	2.2 TestCases for Web Service Binding Specification Section 4	22
41	2.2.1 BWS_4001_TestCase	22
42	2.2.2 BWS_4002_TestCase	22
43	2.2.3 BWS_4003_TestCase	23
44	2.2.4 BWS_4004_TestCase	23
45	2.2.5 BWS_4005_TestCase	24
46	2.2.6 BWS_4006_TestCase	24
47	2.2.7 BWS_4007_TestCase	25
48	2.2.8 BWS_4008_TestCase	25
49	2.3 TestCases for Web Service Binding Specification Section 5	26
50	2.3.1 BWS_5001_TestCase	26
51	2.3.2 BWS_5002_TestCase	26
52	2.3.3 BWS_5003_TestCase	27
53	2.3.4 BWS_5004_TestCase	28

54	2.3.5 BWS_5005_TestCase.....	28
55	2.3.6 BWS_5006_TestCase.....	29
56	3 Cross Mapping of Test Assertions to TestCases.....	30
57	4 Catalog of Test Artifacts.....	33
58	4.1 Composite Files - lower level.....	33
59	4.2 Java Interfaces.....	33
60	4.3 Java Implementation Classes.....	34
61	4.4 WSDL Interface Files.....	34
62	5 Conformance.....	38
63	Appendix A.Test Assertions for the Web Service Binding.....	39
64	A.1.Example Test Assertion.....	39
65	A.2.Test Assertions for Web Service Binding Specification Section 2.....	39
66	A.3.Test Assertions for Web Service Binding Specification Section 4.....	49
67	A.4.Test Assertions for Web Service Binding Specification Section 5.....	50
68	Appendix B.Cross Mapping of Conformance Statements to Assertions.....	54
69	Appendix C.Cross Mapping of Test Assertions to TestCases.....	56
70	Appendix D.Acknowledgments.....	58
71	Appendix E.Revision History.....	59
72		

73 **1 Introduction**

74 This document defines the testcases for the SCA Web Service Binding specification [SCA-WSBinding].

75 The tests described in this document are derived from the normative statements in SCA Web Service
76 **Binding specification via Test Assertions** which are described in Appendix A, "Web Service Binding Test
77 Assertions".

78 **1.1 TestCase Structure**

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

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

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

86 **1.1.1 Test Client**

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

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

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

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

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

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

107 **1.1.2 Test Application**

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

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

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

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

123 Which implementation language to use for test artifacts is controlled by a system variable or environment
124 variable which is read by the test client application, with the name "OASIS_TESTENV_IMPL_LANG". This
125 variable can have one of the following values:

- 126 • "Java" - for Java implementations
- 127 • "BPEL" - for WS_BPEL process implementations
- 128 • "CPP" - for C++ implementations

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

130 **1.1.3 Test Artifacts Organization**

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

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

139 Broadly, Contribution names are as follows:

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

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

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

166 **1.2 Namespaces and Java Package Names**

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

169 **1.2.1 SCA Artifact Namespaces**

170 These apply to artifacts such as Composites

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

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

173 **1.2.2 WSDL Namespace**

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

175 **1.2.3 Java Package name**

176 For Java interface classes and for Java implementation classes

177 org.oasisopen.sca.test

178 **1.3 Terminology**

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

182 **1.4 Normative References**

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

186
187 [TA-GUIDE] OASIS Committee Draft 05, *Test Assertions Guidelines Version 1.0*, August
188 2010 <http://docs.oasis-open.org/tag/guidelines/v1.0/cd05/testassertionsguidelines-cd-05.pdf>

190 [SCA-WSBinding] OASIS Committee Draft 04, "Service Component Architecture Web Service
191 Binding Specification Version 1.1," May 2010.

192
193

<http://docs.oasis-open.org/opencsa/sca-bindings/sca-wsbinding-1.1-spec-cd04.pdf>

194

2 TestCases

195

2.1 TestCases for Web Service Binding Specification Section 2

196

2.1.1 BWS_2001_TestCase

197

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_0002composite TestComposite1composite TestComposite4composite Service1.wsdl
Expected output	Negative test: “exception”

198

199

2.1.2 BWS_2002_TestCase

200

Testcase ID	BWS_2002_TestCase
Test Assertion	BWS-TA-20002
Description	Tests that a <code>binding.ws/@wsdlElement</code> attribute value points to an existing WSDL 1.1 element
Artifacts	BWS_2002_TestCase.java Test_BWS_2002.composite TestInvocation.wsdl TestClient_0002composite TestComposite1composite TestComposite4composite Service1.wsdl
Expected output	Negative test:

	"exception"
--	-------------

201

2.1.3 BWS_2003_TestCase

203

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_0002composite TestComposite1composite TestComposite4composite Service1.wsdl
Expected output	Negative test: "exception"

204

2.1.4 BWS_2004_TestCase

206

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_0002composite TestComposite1composite TestComposite4composite Service1.wsdl
Expected output	Positive test: "BWS_2004 request service1 operation1 invoked service2 operation1 invoked"

207

208 **2.1.5 BWS_2005_TestCase**

209

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"

210

211 **2.1.6 BWS_2006_TestCase**

212

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

	TestComposite4composite 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"

213

2.1.7 BWS_2007_TestCase

215

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_2007composite TestInvocation.wsdl TestClient_0002composite TestComposite1composite TestComposite4composite Service1.wsdl Service1InvalidService.wsdl
Expected output	Negative test: "exception"

216

2.1.8 BWS_2008_TestCase

218

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_2008composite

	TestInvocation.wsdl TestClient_0002composite TestComposite1composite TestComposite4composite Service1.wsdl Service1WithPort.wsdl
Expected output	Positive test: "BWS_2008 request service1 operation1 invoked service2 operation1 invoked"

219

2.1.9 BWS_2009_TestCase

221

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_2009composite TestInvocation.wsdl TestClient_0002composite TestComposite1composite TestComposite9composite Service1.wsdl Service1Superset.wsdl sca_variables.dtd in BWS_2009 contribution
Expected output	Positive test: "BWS_2009 request service1 operation1 invoked service2 operation1 invoked"

222

2.1.10 BWS_2010_TestCase

224

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 TestComposite1composite Service1.wsdl Service1WithBinding.wsdl
Expected output	Positive test: "BWS_2010 request service1 operation1 invoked service2 operation1 invoked"

225

2.1.11 BWS_2011_TestCase

226

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 TestComposite1composite Service1.wsdl Service1WithBinding.wsdl
Expected output	Negative test: "exception"

227

2.1.12 BWS_2012_TestCase

228

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_2012composite TestInvocation.wsdl TestClient_0002composite TestComposite4composite TestComposite9composite 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"

231

2.1.13 BWS_2013_TestCase

232

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_2013composite TestInvocation.wsdl TestClient_0002composite TestComposite1composite TestComposite4composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2013 sca-variables.dtd in contribution BWS_2013

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

234

2.1.14 BWS_2014_TestCase

235

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"

236

2.1.15 BWS_2015_TestCase

237

Testcase ID	BWS_2015_TestCase
Test Assertion	BWS-TA-20016
Description	Tests that where a <binding.ws/> element has a @wsdl: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

	TestComposite1composite TestComposite4composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2015 sca-variables.dtd in contribution BWS_2015
Expected output	Negative test: "exception"

240

2.1.16 BWS_2016_TestCase

242

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_2016composite TestInvocation.wsdl TestClient_0002composite TestComposite1composite TestComposite4composite Service1.wsdl Service1InvalidService.wsdl in contribution BWS_2016 sca-variables.dtd in contribution BWS_2016
Expected output	Negative test: "exception"

243

2.1.17 BWS_2017_TestCase

245

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_2017composite TestInvocation.wsdl TestClient_0002composite TestComposite9composite TestComposite4composite Service1.wsdl Service1Superset.wsdl in contribution BWS_2017 sca-variables.dtd in contribution BWS_2017
Expected output	Negative test: "exception"

246

2.1.18 BWS_2018_TestCase

248

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_2018composite TestInvocation.wsdl TestClient_0002composite TestComposite54composite TestComposite55composite Service1.wsdl Service5.wsdl Service5Callback.wsdl
Expected output	Negative test: "exception"

249

2.1.19 BWS_2019_TestCase

251

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_0002composite TestComposite1composite TestComposite4composite Service1.wsdl
Expected output	Negative test: "exception"

252

2.1.20 BWS_2020_TestCase

253

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_0002composite TestComposite1composite TestComposite4composite Service1.wsdl
Expected output	Negative test: "exception"

254

2.1.21 BWS_2021_TestCase

255

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 TestComposite1composite TestComposite4.composite Service1.wsdl
Expected output	Positive test: "BWS_2021 request service1 operation1 invoked service2 operation1 invoked"

258

2.1.22 BWS_2022_TestCase

260

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 TestComposite1composite TestComposite4.composite Service1.wsdl Service1WithCallback.wsdl
Expected output	Negative test: "exception"

261

2.1.23 BWS_2023_TestCase

263

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"

264

265 2.2 TestCases for Web Service Binding Specification Section 4

266 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"

267

268 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_0002composite Service1.wsdl TestComposite1composite
Expected output	Positive test: "BWS_4002 request service1 operation1"

269

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 endpoint both sends and receives messages in the SOAP v1.2 protocol
Artifacts	BWS_4003_TestCase.java Test_BWS_4003.composite TestInvocation.wsdl TestClient_0002composite Service1.wsdl TestComposite1composite
Expected output	Positive test: "BWS_4003 request service1 operation1"

271

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_0002composite

	Service1.wsdl TestComposite4.composite
Expected output	Positive test: "BWS_4004 request service1 operation1 invoked request Client service1 operation1 invoked"

273

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"

275

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"
--	---

277

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

279

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 portType 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"

281

283 2.3 TestCases for Web Service Binding Specification Section 5

284 2.3.1 BWS_5001_TestCase

Testcase ID	BWS_5001_TestCase
Test Assertion	BWS-TA-50001
Description	<p>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</p> <p>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</p>
Artifacts	<p>BWS_5001_TestCase.java</p> <p>Test_BWS_5001.composite</p> <p>TestInvocation.wsdl</p> <p>TestClient_0002.composite</p> <p>TestComposite54.composite</p> <p>Service5.wsdl</p> <p>Service5Callback.wsdl</p> <p>jaxwsClient.Service5Impl.java</p>
Expected output	<p>Positive test:</p> <p>"BWS_5001 request service1 operation1 invoked request Client service5 operation1 invoked service1 callback1 invoked"</p>

287 2.3.2 BWS_5002_TestCase

Testcase ID	BWS_5002_TestCase
Test Assertion	<p>BWS-TA-50001</p> <p>BWS-TA-50003</p>
Description	<p>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</p> <p>In this case, an SCA component provides a service over binding.ws where</p>

	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"

289

290 **2.3.3 BWS_5003_TestCase**

291

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"

292

293

2.3.4 BWS_5004_TestCase

294

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_0002composite TestComposite54.composite Service5.wsdl Service5Callback.wsdl jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5004 service1 operation1 invoked callback service5Callback callback1 invoked headers OK"

295

296

2.3.5 BWS_5005_TestCase

297

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_0002composite TestComposite54.composite Service5.wsdl

	Service5Callback.wsdl jaxwsClient.Service5Impl.java
Expected output	Positive test: "BWS_5005 service1 operation1 invoked callback service5Callback callback1 invoked headers OK"

298

299 **2.3.6 BWS_5006_TestCase**

300

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"

301

303

4 Catalog of Test Artifacts

304

305

4.1 Composite Files - lower level

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

306

307

4.2 Java Interfaces

308

Name	Description
Service1.java	
Service1Superset.java	

Service5.java	
Service5Callback.java	
TestInvocation.java	

309 **4.3 Java Implementation Classes**

310

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

311 **4.4 WSDL Interface Files**

312

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:

	<ul style="list-style-type: none"> - "operation1" string input, string output
Service5Callback.wsdl	<p>Service5Callback interface, which is the callback interface for Service5</p> <p>1 operation:</p> <ul style="list-style-type: none"> - "callback1" string input, string output
Service1InvalidService.wsdl (BWS_2005 contribution)	<p>Service1 interface</p> <p>- this is Service1 with an invalid WSDL Binding (invalid transport)</p>
Service1InvalidPort.wsdl (BWS_2007 contribution)	<p>Service1 interface</p> <p>- this is Service1 with a PortType incompatible with the regular Service1 PortType</p>
Service1WithPort.wsdl (BWS_2008 contribution)	<p>Service1 interface</p> <p>- this is Service1 with a Port which includes a target address for the service</p>
Service1Superset.wsdl (BWS_2009 contribution)	<p>Service1Superset interface</p> <p>- with additional WSDL Service, Port & Binding elements</p>
Service1WithBinding.wsdl (BWS_2010 contribution)	<p>Service1 interface</p> <p>- with a valid WSDL Binding element</p>
Service1InvalidService.wsdl (BWS_2011 contribution)	<p>Service1 interface</p> <p>- this is Service1 with a WSDL Service made invalid through the use of an invalid WSDL Binding element</p>
Service1Superset.wsdl (BWS_2012 contribution)	<p>Service1Superset interface</p> <p>- with additional WSDL Service, Port & Binding elements</p>
Service1InvalidService.wsdl (BWS_2013 contribution)	<p>Service1 interface</p> <p>- this is Service1 with a WSDL Service made invalid through the use of an invalid WSDL Binding element</p>
Service1Superset.wsdl (BWS_2014 contribution)	<p>Service1Superset interface</p> <p>- with additional WSDL Service, Port & Binding elements</p>
Service1WithBinding.wsdl (BWS_2015 contribution)	<p>Service1 interface</p> <p>- with a valid WSDL Binding element</p>

Service1WithBinding.wsdl (BWS_2016 contribution)	Service1 interface - with a valid WSDL Binding element
Service1Superset.wsdl (BWS_2017 contribution)	Service1Superset interface - 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/Service1Specia1</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

313

314 5 Conformance

315

316 The artifacts contained in the sca-wsbinding-1.1-testcases.zip are considered to be authoritative and take
317 precedence over the artifacts described in this document.

318 This specification defines two targets for conformance:

319 a) WS Binding Testcases

320 b) Web Service Callback (WSCB) Testcases

321 An implementation that claims to conform to the requirements of the WS Binding Testcases MUST be
322 able to run all test cases in Section 2.1 and Section 2.2, producing the 'Expected Output'.

323 An implementation that claims to conform to the WSCB Testcases MUST be able to run all tests in
324 Section 2.3, producing the 'Expected Output'.

325 **Appendix A. Test Assertions for the Web Service**
326 **Binding**

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

329 **A.1. Example Test Assertion**

330 Test assertions are presented in a tabular format with rows corresponding to the entry types defined in the
331 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

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

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

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

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

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

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

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

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

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

Tags	“uri attribute” “reference target”
------	------------------------------------

347

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

348

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

349

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

350

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”

351

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”

352

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”

353

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”

354

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	Mandatory

<u>Level</u>	
Tags	“wsdlElement”

355

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”

356

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”

357

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”

358

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”

359

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”

360

Assertion ID	BWS-TA-20015
Source	[BWS20015]
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 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
Prescription Level	Mandatory
Tags	“wsdlElement”

361

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

362

Assertion ID	BWS-TA-20017
Source	[BWS20018]
Target	The binding.ws/@wsdl:wsdlLocation attribute
Prerequisites	

Predicate	The attribute value points to an existing WSDL 1.1 document
Prescription Level	Mandatory
Tags	“wsdlLocation”

363

Assertion ID	BWS-TA-20018
Source	[BWS20019]
Target	The binding.ws element
Prerequisites	
Predicate	The uri attribute, the endpointReference element and the wsdlElement attribute referring to a WSDL port or service are mutually exclusive
Prescription Level	Mandatory
Tags	“wsdlElement” “uri” “endpoint reference”

364

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”

365

Assertion ID	BWS-TA-20020
Source	[BWS20021]
Target	SCA runtime
Prerequisites	
Predicate	All attributes of binding.ws, i.e., name, uri, requires, policySets, wsdlElement, wsdl:wsdlLocation are supported
Prescription Level	Mandatory
Tags	“wsdlElement” “uri” “requires” “policySets” “name” “wsdlLocation”

366

Assertion ID	BWS-TA-20021
Source	[BWS20022]
Target	SCA runtime
Prerequisites	
Predicate	binding.ws/endpointReference element is supported
Prescription Level	Preferred

Tags	"endpointReference" "SCA runtime"
------	-----------------------------------

367

Assertion ID	BWS-TA-20022
Source	[BWS20023]
Target	SCA runtime
Prerequisites	SCA runtime does not support binding.ws/endpointReference element
Predicate	Rejects any conformant SCA WS Binding XML document that contains binding.ws/endpointReference element
Prescription Level	Mandatory
Tags	"endpointReference" "SCA runtime"

368

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"

369

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"

370

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"

371

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”

372

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 portType
Prescription Level	Mandatory
Tags	“sca runtime” “interface mapping”

373

Assertion ID	BWS-TA-20028
Source	[BWS20029]
Target	A service that uses binding.ws with an HTTP endpoint is hosted by the SCA runtime
Prerequisites	
Predicate	The service returns a WSDL description of itself in response to an HTTP GET request with the “?wsdl” suffix added to the service endpoint URL
Prescription Level	Preferred
Tags	“sca runtime” “HTTP GET” “WSDL description”

374

Assertion ID	BWS-TA-20029
Source	[BWS20030]
Target	A service that uses one of more instances of binding.ws but none of them have HTTP endpoints
Prerequisites	
Predicate	The SCA runtime provides some means for obtaining the WSDL description of the service
Prescription Level	Preferred

Tags	“sca runtime” “WSDL description”
------	----------------------------------

375

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”

376

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 wsoap11: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”

377

Assertion ID	BWS-TA-20032
Source	[BWS20034]
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 1.2 over HTTP binding, using a wsoap12: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	Preferred
Tags	“sca runtime” “WSDL extension” “SOAP 1.2”

378

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

379

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

380

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

381 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"

382

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"

383

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"

384

Assertion ID	BWS-TA-40004
Source	[BWS40004]
Target	portType of an SCA component <service/> or <reference/> element
Prerequisites	portType is either explicitly specified or is derived from the interface specified for the service or reference
Predicate	The portType follows either the rpc-literal pattern or the document-literal pattern
Prescription Level	Mandatory
Tags	"SCA runtime" "intent" "SOAP"

385

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

386

Assertion ID	BWS-TA-40007
Source	[BWS40007]

Target	SCA runtime
Prerequisites	Default transport binding rules are used along with rpc-literal pattern
Predicate	SCA runtime uses the structural URI associated with the binding as the namespace of the child elements of the SOAP body element
Prescription Level	Preferred
Tags	"SCA runtime" "default transport binding rules" "rpc-literal"

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

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

388

Assertion ID	BWS-TA-50002
Source	[BWS50004]
Target	SCA bidirectional service using the binding.ws binding with the SCA Web Services Callback Protocol
Prerequisites	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"
Predicate	An Invalid Addressing Header fault is generated as specified in Section 6.4.1 of WS-Addressing 1.0 SOAP Binding
Prescription Level	Mandatory
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"

389

Assertion ID	BWS-TA-50003
Source	[BWS50005]
Target	SCA runtime
Prerequisites	The callback interface is invoked using the SCA Web Services Callback Protocol
Predicate	The Callback EPR from a request message that invoked the forward interface is used to invoke the callback interface
Prescription	Mandatory

Level	
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"

390

Assertion ID	BWS-TA-50004
Source	[BWS50006]
Target	SCA runtime
Prerequisites	The Callback EPR is selected to invoke the callback interface
Predicate	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
Prescription Level	Mandatory
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "callback EPR"

391

Assertion ID	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"

392

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"

393

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"

394

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
Predicate	The SCA Web Services Callback Protocol is used to invoke callbacks for that binding
Prescription Level	Mandatory
Tags	"SCA runtime" "SCA Web Services Callback Protocol" "WSCallback policy assertion"

395

Assertion ID	BWS-TA-50009
Source	[BWS50013]
Target	WSDL document (?)
Prerequisites	
Predicate	The wsdl:portType that is used does not have WSCallback policy assertion attached
Prescription Level	Mandatory
Tags	"WSCallback policy assertion"

396

Assertion ID	BWS-TA-50010
Source	[BWS50014]
Target	?
Prerequisites	Policy contains WSCallback policy assertion
Predicate	The policy or the assertion is signed
Prescription Level	Preferred
Tags	"WSCallback policy assertion"

397

Assertion ID	BWS-TA-50011
Source	[BWS50015]
Target	SCA runtime
Prerequisites	Policy contains WSCallback policy assertion which is either not signed or does not have an associated security token to specify that the signer has

	the proper claims for the policy
Predicate	The policy is rejected
Prescription Level	Preferred
Tags	"WSCallback policy assertion"

398

399 **Appendix B. Cross Mapping of Normative Statements**
400 **to Assertions**

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

Statement Identifier	Test Assertion
BWS20001	BWS-TA-20001
BWS20002	BWS-TA-20002
BWS20003	BWS-TA-20003
BWS20004	BWS-TA-20004
BWS20005	BWS-TA-20005
BWS20006	BWS-TA-20006
BWS20007	BWS-TA-20007
BWS20008	BWS-TA-20008
BWS20009	BWS-TA-20009
BWS20010	BWS-TA-20010
BWS20011	BWS-TA-20011
BWS20012	BWS-TA-20012
BWS20013	BWS-TA-20013
BWS20014	BWS-TA-20014
BWS20015	BWS-TA-20015
BWS20017	BWS-TA-20016
BWS20018	BWS-TA-20017
BWS20019	BWS-TA-20018
BWS20020	BWS-TA-20019
BWS20021	BWS-TA-20020
BWS20022	BWS-TA-20021
BWS20023	BWS-TA-20022
BWS20024	BWS-TA-20023
BWS20025	BWS-TA-20024
BWS20026	BWS-TA-20025
BWS20027	BWS-TA-20026
BWS20028	BWS-TA-20027
BWS20029	BWS-TA-20028
BWS20030	BWS-TA-20029
BWS20032	BWS-TA-20030
BWS20033	BWS-TA-20031
BWS20034	BWS-TA-20032
BWS20035	BWS-TA-20033
BWS20036	BWS-TA-20034
BWS20037	BWS-TA-20035

403

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

Statement Identifier	Test Assertion
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

405 **Appendix C. Cross Mapping of Test Assertions to**
406 **TestCases**

407 This section contains a list of Test Assertions and the corresponding 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_2019_TestCase
BWS-TA-20024	BWS_2020_TestCase
BWS-TA-20025	BWS_2021_TestCase
BWS-TA-20026	<i>Language Dependent</i> 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

408

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	Optional function - no test

409

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 understand WS Policy - untestable
BWS-TA-50010	Optional function - no test
BWS-TA-50011	Optional function - no test

410

411 **Appendix D. Acknowledgments**

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

414 **Participants:**

415

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

416

417

Appendix E. Revision History

418

Revision	Date	Editor	Changes Made
wd01	13/05/10	Mike Edwards	Initial version
wd02	17/05/10	Mike Edwards	Added testcases BWS_2002 to BWS_2006
wd03	01/06/10	Mike Edwards	Added testcases BWS_2007 to BWS_2023
wd04	07/06/10	Mike Edwards	Editorial changes Added testcases BWS_4001 to BWS_4007
wd05	14/06/10	Mike Edwards	Modified BWS_5001 Added BWS_5002 to BWS_5006
cd01	01/07/10	Mike Edwards	All changes accepted Frontmatter edited to conform to OASIS requirements Completed artifact catalog in Section 4
csd01-rev1	01/04/10	Simon Holdsworth	Applied resolutions to issues BINDINGS-142, BINDINGS-146, BINDINGS-148

419