



TestCases for the SCA Assembly Model Version 1.1 Specification

Committee Draft 02 / Public Review Draft 021 / ~~Public Review Draft 01~~

8 June 2010

~~16 June 2009~~

Specification URIs:

This Version:

<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd02.html>
<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd02.odt>
<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd02.pdf>
(Authoritative)

Previous Version:

<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd01.html>
<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd01.odt>
<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd01.pdf>
(Authoritative)

~~Previous Version:~~

Latest Version:

<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases.html>
<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases.odt>
<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases.pdf> (Authoritative)

Technical Committee:

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

Chair(s):

Martin Chapman, Oracle
Mike Edwards, IBM

Editor(s):

Mike Edwards, IBM
Eric Wells, Hitachi
Plamen Pavlov, SAP

Related Work:

The Test Suite artifacts relating to this document can be found here:

<http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-spec.pdf>
[testcases-ed01.zip](#)

This document is related to:

- Service Component Architecture Assembly Specification Version 1.1

Declared XML Namespace(s):

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

Abstract:

This document defines the TestCases for the SCA Assembly 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 Assembly specification.

Status:

This document was last revised or approved by the OASIS Service Component Architecture / Assembly (SCA-Assembly) 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-assembly/>.

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-assembly/ipr.php>).

The non-normative errata page for this specification is located at

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

Notices

Copyright © OASIS® 2009 - 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, [insert specific trademarked names, abbreviations, etc. here] 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.....	5
1.1 TestCase Structure.....	5
1.2 Namespaces and Java Package Names.....	7
1.3 Terminology.....	8
1.4 Normative References.....	8
1.5 Non_Normative References.....	8
2 TestCases.....	9
2.1 Section 4.....	9
2.2 Section 5.....	12
2.3 Section 6.....	33
2.4 Section 7.....	52
2.5 Section 8.....	52
2.6 Section 9.....	62
2.7 Section 10.....	64
2.8 Section 12.....	65
2.9 Section 13.....	72
2.10 Section 14.....	76
3 Cross Mapping of Test Assertions to TestCases.....	77
4 Catalog of Test Artifacts.....	84
4.1 Composite Files - lower level.....	84
4.2 Java Interfaces.....	91
4.3 Java Implementation Classes.....	92
4.4 WSDL Interface Files.....	94
5 Conformance.....	96
Appendix A.Acknowledgments.....	97
Appendix B.Revision History.....	98

1 Introduction

This document defines the TestCases for the SCA Assembly specification.

The tests described in this document are related to the Test Assertions described in the [SCA Assembly Test Assertions document \[ASM-TA\]](#).

1.1 TestCase Structure

The SCA Assembly 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.

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. The client is structured to permit Web services to be replaced by some other binding (eg JMS) should the SCA runtime under test not support Web services as a binding technology.

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

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:

- ASM_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.
- ASM_nnnn_Java - a contribution for a specific testcase where there is a need for language specific artifacts that relate to that testcase alone
- General - a shared contribution containing implementation type independent artifacts that can be used by many testcases.
- 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.
- General_BPEL - a shared contribution containing implementation type dependent artifacts for BPEL processes. These artifacts can include both BPEL processes and also SCA composites that directly use BPEL processes.

- 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.

Replaceable Values for SCA Artifacts

In general, the SCA artifacts contained in the various contributions cannot be changed - such changes would invalidate the test suite. However, there are a small number of instances where there is a need to adjust some values used in the SCA artifacts to meet the configuration of the SCA Runtime under test. This in particular applies to some service endpoint addresses of some Web services that occur in some of the testcases.

To deal with this situation, the adjustable values are expressed in terms of XML Entities, with Entity References used within SCA artifacts such as composites. The XML Entities are declared within a DTD file, which is referenced from the SCA artifact files. The Entity values declared in the DTD file can be changed to match the configuration of the SCA Runtime under test - this is the only change permitted to the SCA artifacts when run against a particular SCA runtime.

The DTD file has the name **sca_variables.dtd** and it is contained in the **General** contribution.

Each Entity value declared in the **sca_variables.dtd** file has an explanation of what the Entity is intended to hold and it is expected that the values will be adjusted to suit the runtime under test before the test suite is executed.

1.2 Namespaces and Java Package Names

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

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>

WSDL Namespace

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

122 **Java Package name**

123 For Java interface classes and for Java implementation classes

124 `org.oasisopen.sca.test`

125 **1.3 Terminology**

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

129 **1.4 Normative References**

130	[RFC 2119]	S. Bradner. <i>Key words for use in RFCs to Indicate Requirement Levels</i> . IETF
131		RFC 2119, March 1997.
132		http://www.ietf.org/rfc/rfc2119.txt .
133	[ASM-TA]	Oasis Committee Draft 02, <i>Test Assertions for the SCA Assembly Model Version</i>
134		1.1 Specification , June 2010. http://docs.oasis-open.org/opencsa/sca-
135		assembly/sca-assembly-1.1-test-assertions-cd02.pdf SCA Assembly Test
136		Assertions
137		http://docs.oasis-open.org/sca-assembly/sca-assembly-1.1-test-assertions-WD16.pdf

139 Non-normative References

140	None
141	[TBD] ————— [TBD]

143 **1.5 Non_Normative References**

144 [N/A](#)

2 TestCases

2.1 Section 4

ASM_4001_TestCase

Testcase ID	ASM_4001_TestCase
Test Assertion	ASM-TA-4001
Description	Tests that the @constrainingType attribute value of a <componentType/> element is the QName of a <constrainingType/> element
Artifacts	ASM_4001_TestCase.java Test_ASM_4001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite42.composite TestComposite42.constraint Service1.wsdl
Expected output	Positive test: "ASM_4001 request service1 operation1 invoked"

ASM_4002_TestCase

Testcase ID	ASM_4002_TestCase
Test Assertion	ASM-TA-4002
Description	Tests that the @name attribute value of a <service/> element of a <componentType/> is not the same as the @name attribute value of any other <service/> of the <componentType/>
Artifacts	ASM_4002_TestCase.java Test_ASM_4002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite16.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_4003_TestCase

Testcase ID	ASM_4003_TestCase
Test Assertion	ASM-TA-4003
Description	Tests that the @name attribute value of a <reference/> element of a <componentType/> is not the same as the @name attribute value of any other <reference/> of the <componentType/>
Artifacts	ASM_4003_TestCase.java Test_ASM_4003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite19.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_4004_TestCase

Testcase ID	ASM_4004_TestCase
Test Assertion	ASM-TA-4004
Description	Tests that the @name attribute value of a <property/> element of a <componentType/> is not the same as the @name attribute value of any other <property/> of the <componentType/>
Artifacts	ASM_4004_TestCase.java Test_ASM_4004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite26.composite
Expected output	Negative test: "exception"

ASM_4005_TestCase

Testcase ID	ASM_4005_TestCase
Test Assertion	ASM-TA-4006

Description	Tests that the @type attribute value of a <property/> element of a <componentType/> is the QName of an XML schema type
Artifacts	ASM_4005_TestCase.java Test_ASM_4005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite43.composite Service1.wsdl
Expected output	Positive test: “ASM_4005 request service1 operation1 invoked service2 operation1 invoked ”

158

159 **ASM_4006_TestCase**

160

Testcase ID	ASM_4006_TestCase
Test Assertion	ASM-TA-4007
Description	Tests that the @element attribute value of a <property/> element of a <componentType/> is the QName of an XSD global element
Artifacts	ASM_4006_TestCase.java Test_ASM_4006.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite44.composite Service1.wsdl
Expected output	Positive test: “ASM_4006 request service1 operation1 invoked service2 operation1 invoked ”

161

162 **ASM_4007_TestCase**

163

Testcase ID	ASM_4007_TestCase
Test Assertion	ASM-TA-4009 ASM-TA-4010

Description	Tests that a <property/> element of a <componentType/> does not have both @element and @type attributes set at the same time
Artifacts	ASM_4007_TestCase.java Test_ASM_4007.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite45.composite Service1.wsdl
Expected output	Negative test: "exception"

164

ASM_4008_TestCase

165

166

Testcase ID	ASM_4008_TestCase
Test Assertion	ASM-TA-4011
Description	Tests that a <property/> element of a <componentType/> which has @mustSupply="true" must have a value specified for that property in the <component/> that uses the <implementation/> with that <componentType/>
Artifacts	ASM_4008_TestCase.java Test_ASM_4008.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite46.composite Service1.wsdl
Expected output	Negative test: "exception"

167

2.2 Section 5

ASM_5002_TestCase

Testcase ID	ASM_5002_TestCase
Test Assertion	ASM-TA-5025 (+ ASM-TA-5031 as it uses simple property values) (+ ASM-TA-4008 as it involves a component property value being used by the runtime implementation instance)
Description	Tests that a component with a single reference of multiplicity 1..1 with @target attribute set to the service of a second component in the same composite, where the service interface is compatible with the reference interface, can successfully invoke the target service.
Artifacts	ASM_5002_TestCase.java Test_ASM_5002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_5002 request service1 operation1 invoked"

ASM_5003_TestCase

Testcase ID	ASM_5003_TestCase
Test Assertion	ASM-TA-5026
Description	Tests that a component with a single reference of multiplicity 1..n with @target attribute having 3 entries (the service of three other components in the same composite, where the service interface is compatible with the reference interface), can successfully invoke all the target services.
Artifacts	ASM_5003_TestCase.java Test_ASM_5003.composite TestInvocation.wsdl TestClient_5003.composite TestComposite1.composite Service1.wsdl

Expected output	Positive test: "ASM_5003 request service1 operation1 invokedservice2 operation1 invoked service3 operation1 invoked"
-----------------	---

174

175 **ASM_5004_TestCase**

176

Testcase ID	ASM_5004_TestCase
Test Assertion	ASM-TA-5027
Description	Tests that a component with a single reference of multiplicity 1..1 with @target attribute having 3 entries (the service of three other components in the same composite, where the service interface is compatible with the reference interface), has an invalid configuration and cannot run.
Artifacts	ASM_5004_TestCase.java Test_ASM_5004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

177

178 **ASM_5005_TestCase**

179

Testcase ID	ASM_5005_TestCase
Test Assertion	ASM-TA-5001
Description	Tests that a component with multiple <service/> elements must have unique @name attributes for each service
Artifacts	ASM_5005_TestCase.java Test_ASM_5005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite2.composite Service1.wsdl
Expected output	Negative test: "exception"

180

ASM_5006_TestCase

Testcase ID	ASM_5006_TestCase
Test Assertion	ASM-TA-5002
Description	Tests that the @name attribute of a <service/> element of a <component/> must match the @name attribute of a <service/> element in the componentType of the <implementation/> used by the <component/>
Artifacts	ASM_5006_TestCase.java Test_ASM_5006.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5007_TestCase

Testcase ID	ASM_5007_TestCase
Test Assertion	ASM-TA-5003
Description	Test that the @name attribute of a <reference/> element of a component is not equal to the @name attribute of any other <reference/> element of the same component
Artifacts	ASM_5007_TestCase.java Test_ASM_5007.composite TestInvocation.wsdlTestClient_0002.composite TestComposite3.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5008_TestCase

Testcase ID	ASM_5008_TestCase
Test Assertion	ASM-TA-5004
Description	Tests that the @name attribute of a <reference/> element of a component is equal to the @name attribute of a <reference/> element of the <componentType/> of the <implementation/> of the <component/>
Artifacts	ASM_5008_TestCase.java Test_ASM_5008.composite TestInvocation.wsdl TestClient_0002.composite TestComposite4.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5009_TestCase

Testcase ID	ASM_5009_TestCase
Test Assertion	ASM-TA-5005
Description	Tests that the @name attribute of a <component/> element in a <composite/> is not the same as the @name attribute of a second component in the same <composite/>
Artifacts	ASM_5009_TestCase.java Test_ASM_5009.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5010_TestCase

Testcase ID	ASM_5010_TestCase
Test Assertion	ASM-TA-5006
Description	Tests that the interface specified by the <interface/> child element of a <service/> element is a compatible subset of the interface declared for the <service/> element with the same @name attribute in the <componentType/> of the <implementation/> of the <component/> parent of the <service/>
Artifacts	ASM_5010_TestCase.java Test_ASM_5010.composite TestInvocation.wsdl TestClient_0002.composite TestComposite5.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5011_TestCase

Testcase ID	ASM_5011_TestCase
Test Assertion	ASM-TA-5010
Description	Tests that the multiplicity of a <reference/> element of a <component/> can be 1..1 where the multiplicity of the equivalent <reference/> element in the <componentType/> of the <implementation/> of the <component/> is 1..n
Artifacts	ASM_5011_TestCase.java Test_ASM_5011.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite6.composite ASM_0002_Client.java Service1.wsdl
Expected output	Positive test: "ASM_5011 request service1 operation1 invoked service2 operation1 invoked"

ASM_5012_TestCase

Testcase ID	ASM_0012_TestCase
Test Assertion	ASM-TA-5011
Description	Tests that the multiplicity of a <reference/> element of a <component/> can be any one of 0..1, 1..1, 0..n, 1..n where the multiplicity of the equivalent <reference/> element in the <componentType/> of the <implementation/> of the <component/> is 0..n
Artifacts	ASM_5012_TestCase.java Test_ASM_5012.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite7.composite Service1.wsdl
Expected output	Positive test: “ASM_5012 request service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked service4 operation1 invoked service5 operation1 invoked”

ASM_5013_TestCase

Testcase ID	ASM_5013_TestCase
Test Assertion	ASM-TA-5012
Description	Tests that the multiplicity of a <reference/> element of a <component/> is 1..1 where the multiplicity of the equivalent <reference/> element in the <componentType/> of the <implementation/> of the <component/> is 1..1
Artifacts	ASM_5013_TestCase.java Test_ASM_5013.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl

Expected output	Positive test: "ASM_5013 request service1 operation1 invoked service2 operation1 invoked"
-----------------	--

ASM_5014_TestCase

Testcase ID	ASM_5014_TestCase
Test Assertion	ASM-TA-5012
Description	Tests that the multiplicity of a <reference/> element of a <component/> is 0..n where the multiplicity of the equivalent <reference/> element in the <componentType/> of the <implementation/> of the <component/> is 1..1
Artifacts	ASM_5014_TestCase.java Test_ASM_5014.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite ASM_0002_Client.java Service1.wsdl
Expected output	Negative test: "exception"

ASM_5015_TestCase

Testcase ID	ASM_5015_TestCase
Test Assertion	ASM-TA-5013
Description	Tests that the multiplicity of a <reference/> element of a <component/> is 1..1 where the multiplicity of the equivalent <reference/> element in the <componentType/> of the <implementation/> of the <component/> is 0..1
Artifacts	ASM_5015_TestCase.java Test_ASM_5015.composite TestInvocation.wsdl TestClient_0002.composite

	TestComposite1.composite TestComposite8.composite Service1.wsdl
Expected output	Positive test: "ASM_5015 request service1 operation1 invoked service2 operation1 invoked"

217

218

219 **ASM_5016_TestCase**

220

Testcase ID	ASM_5016_TestCase
Test Assertion	ASM-TA-5016
Description	Tests that the interface declared by the <interface/> child element of a <reference/> of a <component/> is a compatible superset of the interface declared on the corresponding <reference/> of the componentType of the <implementation/> of the <component/>
Artifacts	ASM_5016_TestCase.java Test_ASM_5016.composite TestInvocation.wsdl TestClient_0002.composite TestComposite4.composite TestComposite9.composite Service1.wsdl Service1Superset.wsdl
Expected output	Positive test: "ASM_5016 request service1 operation1 invoked service2 operation1 invoked"

221

222

223 **ASM_5017_TestCase**

224

Testcase ID	ASM_5017_TestCase
Test Assertion	ASM-TA-5017
Description	Tests that where a <reference/> of a <component/> has no <binding/> child element, but the corresponding <reference/> element in the componentType does have a binding child element, the <binding/> from

	the componentType is used for the reference
Artifacts	ASM_5017_TestCase.java Test_ASM_5017.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite10.composite ASM_0002_Client.java Service1.wsdl
Expected output	Positive test: “ASM_5017 request service1 operation1 invoked service2 operation1 invoked”

225

226 **ASM_5018_TestCase**

227

Testcase ID	ASM_5018_TestCase
Test Assertion	ASM-TA-5018 ASM-TA-5024
Description	Tests that where a <reference/> of a <component/> has a <binding/> child element and the corresponding <reference/> element in the componentType has a binding child element, the <binding/> from the <reference/> is used for the reference
Artifacts	ASM_5018_TestCase.java Test_ASM_5018.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite10.composite Service1.wsdl
Expected output	Positive test: “ASM_5018 request service1 operation1 invoked service3 operation1 invoked”

228

229 **ASM_5019_TestCase**

230

Testcase ID	ASM_5019_TestCase
Test Assertion	ASM-TA-5021
Description	Tests that where a <reference/> of a <component/> has @autowire="true" and also has @target identifying a single target service that the reference is wired only to the single service identified by the @target attribute
Artifacts	ASM_5019_TestCase.java Test_ASM_5019.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite7.composite Service1.wsdl
Expected output	Positive test: "ASM_5019 request service1 operation1 invoked service3 operation1 invoked"

231

232 ASM_5020_TestCase

233

Testcase ID	ASM_5020_TestCase
Test Assertion	ASM-TA-5022
Description	Tests that where a <reference/> of a <component/> has @autowire=true and where there is also a single <wire/> element which as that <reference/> as its source that the reference is wired only to the single service identified by the <wire/> element
Artifacts	ASM_5020_TestCase.java Test_ASM_5020.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite7.composite Service1.wsdl
Expected output	Positive test: "ASM_5020 request service1 operation1 invoked service3 operation1 invoked"

234

235

236

ASM_5021_TestCase

Testcase ID	ASM_5021_TestCase
Test Assertion	ASM-TA-5023
Description	Tests that where a <reference/> of a <component/> has @autowire=true and where the <reference/> has a <binding/> child element which declares a single target service, the reference is wired only to the single service identified by the <wire/> element
Artifacts	ASM_5021_TestCase.java Test_ASM_5021.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite7.composite Service1.wsdl
Expected output	Positive test: "ASM_5021 request service1 operation1 invoked service3 operation1 invoked"

ASM_5022_TestCase

Testcase ID	ASM_5022_TestCase
Test Assertion	ASM-TA-5028
Description	Tests that where a <component/> <reference/> with multiplicity=0..1 is unwired (by any means) that the component executes correctly
Artifacts	ASM_5022_TestCase.java Test_ASM_5022.composite TestInvocation.wsdl TestClient_0002.composite TestComposite8.composite Service1.wsdl
Expected output	Positive test: "ASM_5022 request service1 operation1 invoked"

ASM_5023_TestCase

Testcase ID	ASM_5023_TestCase
Test Assertion	ASM-TA-5029
Description	Tests that where a <component/> <reference/> with multiplicity=1..1 is wired to a component service in the same composite and is also promoted by a <composite/> <reference/> which is wired to a target service by a component using the <composite/> as its implementation, that this is an error
Artifacts	ASM_5023_TestCase.java Test_ASM_5023.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite11.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5024_TestCase

Testcase ID	ASM_0024_TestCase
Test Assertion	ASM-TA-5030
Description	Tests that where a <component/> <reference/> has @target set to some service, that the reference can have no child <binding/> elements
Artifacts	ASM_5024_TestCase.java Test_ASM_5024.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

252

253 **ASM_5025_TestCase**

254

<u>Testcase ID</u>	<u>ASM_5025_TestCase</u>
<u>Test Assertion</u>	<u>ASM-TA-5032</u> <u>ASM-TA-5046</u>
<u>Description</u>	Tests that where a <component/> <property/> has its value set by means of a child <value/> element, that the type of the <value/> element matches the type declared for the <property/> element
<u>Artifacts</u>	<u>ASM_5025_TestCase.java</u> <u>Test_ASM_5025.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite12.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.wsdl</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"ASM_5025 request service1 operation1 invokedcomplex1complex2"</u>
<u>Testcase ID</u>	<u>ASM_0025_TestCase</u>
<u>Test Assertion</u>	<u>ASM-TA-5032</u>
<u>Description</u>	Tests that where a <component/> <property/> has its value set by means of a child <value/> element, that the type of the <value/> element matches the type declared for the <property/> element
<u>Artifacts</u>	<u>ASM_5025_TestCase.java</u> <u>Test_ASM_5025.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite12.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.wsdl</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"ASM_5025 request service1 operation1 invokedcomplex1complex2"</u>

255

256

ASM_5026_TestCase

Testcase ID	ASM_5026_TestCase
Test Assertion	ASM-TA-5033
Description	Tests that where a <component/> <property/> has its value set by means of a child element that is NOT a <value/> element, that a) the type of the <property/> element is declared as an XML Schema global element by its @element attribute and that the child element is an instance of that global element
Artifacts	ASM_5026_TestCase.java Test_ASM_5026.composite TestInvocation.wsdl TestClient_0002.composite TestComposite13.composite Service1.wsdl
Expected output	Positive test: "ASM_5026 request service1 operation1 invokedcomplex1complex2"

ASM_5027_TestCase

Testcase ID	ASM_5027_TestCase
Test Assertion	ASM-TA-5034
Description	Tests that where a <component/> has 2 or more <property/> child elements that each <property/> element has a unique @name attribute value
Artifacts	ASM_5027_TestCase.java Test_ASM_5027.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5028_TestCase

Testcase ID	ASM_5028_TestCase
Test Assertion	ASM-TA-5035
Description	Tests that a <component/> <property/> @name attribute matches the @name attribute of one of the <property/> elements of the <componentType/> of the <implementation/> of the <component/>
Artifacts	ASM_5028_TestCase.java Test_ASM_5028.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5029_TestCase

Testcase ID	ASM_5029_TestCase
Test Assertion	ASM-TA-5036
Description	Tests that where a <component/> <property/> has @many=false that there is only one <value/> child element of the <property/>
Artifacts	ASM_5029_TestCase.java Test_ASM_5029.composite TestInvocation.wsdl TestClient_0002.composite TestComposite12.composite Service1.wsdl Test_Types.xsd
Expected output	Negative test: "exception"

ASM_5030_TestCase

Testcase ID	ASM_5030_TestCase
-------------	-------------------

Test Assertion	ASM-TA-5037
Description	Tests that a <component/> <property/> with a @value attribute set does not have any <value/> child elements
Artifacts	ASM_5030_TestCase.java Test_ASM_5030.composite TestInvocation.wsdl TestClient_0002.composite TestComposite14.composite Service1.wsdl
Expected output	Negative test: "exception"

273

274 **ASM_5031_TestCase**

275

Testcase ID	ASM_5031_TestCase
Test Assertion	ASM-TA-5038
Description	Tests that where a <component/> <reference/> has its @target attribute set to a valid service and the <composite/> also contains a <wire/> element with @source set to the <component/> <reference/> and @target set to a second valid service and has @replace=true, the <reference/> is wired to the second service only
Artifacts	ASM_5031_TestCase.java Test_ASM_5031.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Positive test: "ASM_0031 request service1 operation1 invoked service3 operation1 invoked"

276

277 **ASM_5032_TestCase**

278

Testcase ID	ASM_5032_TestCase
Test Assertion	ASM-TA-5007

<u>Description</u>	<u>Tests that where a <component/> <service/> has no <binding/> subelements and the corresponding <service/> element of the <componentType/> of the <component/> <implementation/> has one or more <binding/> subelements, the <component/> <service/> is made available with all the bindings declared on the <service/> element of the <componentType/></u>
<u>Artifacts</u>	<u>ASM_5032_TestCase.java</u> <u>Test_ASM_5032.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite6.composite</u> <u>TestComposite15.composite</u> <u>Service1.wsdl</u> <u>sca-variables.dtd</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"ASM_5032 request service1 operation1 invokedservice2 operation1 invokedservice2 operation1 invoked"</u>
<u>Testcase ID</u>	<u>ASM_5032_TestCase</u>
<u>Test Assertion</u>	<u>ASM-TA-5007</u>
<u>Description</u>	<u>Tests that where a <component/> <service/> has no <binding/> subelements and the corresponding <service/> element of the <componentType/> of the <component/> <implementation/> has one or more <binding/> subelements, the <component/> <service/> is made available with all the bindings declared on the <service/> element of the <componentType/></u>
<u>Artifacts</u>	<u>ASM_5032_TestCase.java</u> <u>Test_ASM_5032.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite6.composite</u> <u>TestComposite15.composite</u> <u>Service1.wsdl</u>
<u>Expected-output</u>	<u>Positive test:</u> <u>"ASM_5032 request service1 operation1 invokedservice2 operation1 invokedservice2 operation1 invoked"</u>

279

280

281

ASM_5033_TestCase

Testcase ID	ASM_5033_TestCase
Test Assertion	ASM-TA-5008
Description	Tests that where a <component/> <service/> has no <binding/> subelements and the corresponding <service/> element of the <componentType/> of the <component/> <implementation/> has no <binding/> subelements, the <component/> <service/> is made available with binding.sca
Artifacts	ASM_5033_TestCase.java Test_ASM_5033.composite TestInvocation.wsdl TestClient_0002.composite TestComposite6.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_5033 request service1 operation1 invokedservice2 operation1 invokedservice2 operation1 invoked"

ASM_5034_TestCase

Testcase ID	ASM_5034_TestCase
Test Assertion	ASM-TA-5009
Description	Tests that where a <component/> <service/> has 1 or more <binding/> subelements the <component/> <service/> is made available with all the bindings declared on the <service/> element
Artifacts	ASM_5034_TestCase.java Test_ASM_5034.composite TestInvocation.wsdl TestClient_0002.composite TestComposite6.composite TestComposite1.composite Service1.wsdl sca-variables.dtd

Expected output	Positive test: "ASM_5034 request service1 operation1 invokedservice2 operation1 invokedservice2 operation1 invoked"
Testcase ID	ASM_5034_TestCase
Test Assertion	ASM-TA-5009
Description	Tests that where a <component/> <service/> has 1 or more <binding/> subelements the <component/> <service/> is made available with all the bindings declared on the <service/> element
Artifacts	ASM_5034_TestCase.java Test_ASM_5034.composite TestInvocation.wsdl TestClient_0002.composite TestComposite6.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_5034 request service1 operation1 invokedservice2 operation1 invokedservice2 operation1 invoked"

289

290 ASM_5035_TestCase

291

Testcase ID	ASM_5035_TestCase
Test Assertion	ASM-TA-5039 ASM-TA-5040
Description	Tests that where a <component/> <property/> element has a @type attribute declared it does not also have a @element attribute declared
Artifacts	ASM_5035_TestCase.java Test_ASM_5035.composite TestInvocation.wsdl TestClient_0002.composite TestComposite12.composite Service1.wsdl
Expected output	Negative test: "exception"

292

ASM_5036_TestCase

Testcase ID	ASM_5036_TestCase
Test Assertion	ASM-TA-5041 ASM-TA-5046
Description	Tests that where a <component/> <property/> has a type declared, the declared type is the same type as the declared type for the corresponding <property/> element in the componentType of the <implementation/> subelement of the <component/>
Artifacts	ASM_5036_TestCase.java Test_ASM_5036.composite TestInvocation.wsdl TestClient_0002.composite TestComposite12.composite Service1.wsdl
Expected output	Negative test: "exception"
Testcase ID	ASM_5036_TestCase
Test Assertion	ASM-TA-5041
Description	Tests that where a <component/> <property/> has a type declared, the declared type is the same type as the declared type for the corresponding <property/> element in the componentType of the <implementation/> subelement of the <component/>
Artifacts	ASM_5036_TestCase.java Test_ASM_5036.composite TestInvocation.wsdl TestClient_0002.composite TestComposite12.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_5037_TestCase

Testcase ID	ASM_5037_TestCase
Test Assertion	ASM-TA-5042
Description	Tests that a <component/> <property/> has an @name attribute value that

	matches the @name attribute value of one of the <property/> elements in the componentType of the <implementation/> subelement of the <component/>
Artifacts	ASM_5037_TestCase.java Test_ASM_5037.composite TestInvocation.wsdl TestClient_0002.composite TestComposite12.composite Service1.wsdl
Expected output	Negative test: "exception"

298

299 | ASM_5038_TestCase

300 |

<u>Testcase ID</u>	<u>ASM_5038_TestCase</u>
<u>Test Assertion</u>	<u>ASM-TA-5031</u>
<u>Description</u>	<u>Tests that where a <component/> <property/> element has an XML schema simple type and has a @value attribute declared, that the @value attribute value contains a single value with a type matching the type of the <property/> element</u>
<u>Artifacts</u>	<u>ASM_5038_TestCase.java</u> <u>Test_ASM_5038.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite73.composite</u> <u>Service1.wsdl</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

301 |

302 | ASM_5039_TestCase

303 |

<u>Testcase ID</u>	<u>ASM_5039_TestCase</u>
<u>Test Assertion</u>	<u>ASM-TA-5045</u>
<u>Description</u>	<u>Tests that where a property element of a component declares a type and has a @source attribute that gets the value of the property from a composite <property/> element, that the type of the component property is compatible with the type of the composite property</u>

<u>Artifacts</u>	ASM_5039_TestCase.java Test_ASM_5039.composite TestInvocation.wsdl TestClient_0002.composite TestComposite12.composite TestComposite71.composite Service1.wsdl
<u>Expected output</u>	Negative test: "exception"

304

ASM_5040_TestCase

305

306

<u>Testcase ID</u>	ASM_5040_TestCase
<u>Test Assertion</u>	ASM-TA-5047
<u>Description</u>	Tests that where a component <reference/> of multiplicity 1..1 has @nonOverridable="true", that the reference is not promoted by a composite reference
<u>Artifacts</u>	ASM_5040_TestCase.java Test_ASM_5040.composite TestInvocation.wsdl TestClient_0002.composite TestComposite74.composite Service1.wsdl
<u>Expected output</u>	Negative test: "exception"

307

ASM_5041_TestCase

308

309

<u>Testcase ID</u>	ASM_5041_TestCase
<u>Test Assertion</u>	ASM-TA-5048
<u>Description</u>	Tests that where a component <reference/> has no @autowire attribute, no @target attribute, no <binding/> subelement and is not promoted by a composite reference, and where the component declares @autowire="true" and where the composite containing the component contains one target service that is compatible with the reference, that the reference is wired to that target service and can invoke the service

<u>Artifacts</u>	ASM_5041_TestCase.java Test_ASM_5041.composite TestInvocation.wsdl TestClient_0002.composite TestComposite8.composite TestComposite1.composite Service1.wsdl
<u>Expected output</u>	Positive test: "ASM_5041 request service1 operation1 invoked second operation1 invoked"

310

311 **ASM_5042_TestCase**

312

<u>Testcase ID</u>	ASM_5042_TestCase
<u>Test Assertion</u>	ASM-TA-5049
<u>Description</u>	Tests that where a component <reference/> has no @autowire attribute, no @target attribute, no <binding/> subelement and is not promoted by a composite reference, and where the component does not declare an @autowire attribute but the composite containing the component declares @autowire="true" and where the composite containing the component contains one target service that is compatible with the reference, that the reference is wired to that target service and can invoke the service
<u>Artifacts</u>	ASM_5042_TestCase.java Test_ASM_5042.composite TestInvocation.wsdl TestClient_0002.composite TestComposite8.composite TestComposite1.composite Service1.wsdl
<u>Expected output</u>	Positive test: "ASM_5042 request service1 operation1 invoked second operation1 invoked"

313

2.3 Section 6

ASM_6001_TestCase

Testcase ID	ASM_6001_TestCase
Test Assertion	ASM-TA-6004
Description	Tests that where an SCA contribution contains two <composite/> elements with the same name and same targetNamespace that an SCA application using that composite does not run and that the SCA runtime reports an error
Artifacts	ASM_6001_TestCase.java Test_ASM_6001.composite TestInvocation.wsdl TestClient_0002.composite TestDuplicate1.composite TestDuplicate2.composite ASM_0002_Client.java Service1.wsdl
Expected output	Negative test: "exception"

ASM_6002_TestCase

Testcase ID	ASM_6002_TestCase
Test Assertion	ASM-TA-6006
Description	Tests that where a <composite/> has multiple <service/> subelements that the @name attributes are unique across those subelements
Artifacts	ASM_6002_TestCase.java Test_ASM_6002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite16.composite
Expected output	Negative test: "exception"

322

323 **ASM_6003_TestCase**

324

Testcase ID	ASM_6003_TestCase
Test Assertion	ASM-TA-6007
Description	Tests that where a <composite/> has a <service/> subelement that the @promote attribute identifies a <service/> of a <component/> in the <composite/>
Artifacts	ASM_6003_TestCase.java Test_ASM_6003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite17.composite Service1.wsdl
Expected output	Negative test: "exception"

325

326 **ASM_6004_TestCase**

327

Testcase ID	ASM_6004_TestCase
Test Assertion	ASM-TA-6008
Description	Tests that the <interface/> declared on a <service/> subelement of a <composite/> is a compatible subset of the <interface/> declared on the <component/> <service/> reference by the @promote attribute
Artifacts	ASM_6004_TestCase.java Test_ASM_6004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite18.composite
Expected output	Positive test: "ASM_6004 request service1 operation1 invoked"

328

329 **ASM_6005_TestCase**

330

Testcase ID	ASM_6005_TestCase
Test Assertion	ASM-TA-6009
Description	Tests that the @name attribute of a <composite/> <reference/> element is unique amongst the @name attributes of all the <reference/> elements of that <composite/>
Artifacts	ASM_6005_TestCase.java Test_ASM_6005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite19.composite
Expected output	Negative test: "exception"

331

332 ASM_6006_TestCase

333

Testcase ID	ASM_6006_TestCase
Test Assertion	ASM-TA-6010
Description	Tests that the @promote attribute of a <composite/> <reference/> contains one or more URIs that each point to a <reference/> of a <component/> in the <composite/>
Artifacts	ASM_6006_TestCase.java Test_ASM_6006.composite TestInvocation.wsdl TestClient_0002.composite TestComposite20.composite TestComposite1.composite
Expected output	Negative test: "exception"

334

335

336 ASM_6007_TestCase

337

Testcase ID	ASM_6007_TestCase
Test Assertion	ASM-TA-6011
Description	Tests that a <composite/> <reference/> can declare an <interface/> that is a compatible superset of the <interface/> declared by each <component/>

	<reference/> promoted by the composite reference
Artifacts	ASM_6007_TestCase.java Test_ASM_6007.composite TestInvocation.wsdl TestClient_0002.composite TestComposite21.composite TestComposite9.composite
Expected output	Positive test: "ASM_6007 request service1 operation1 invoked service2 operation1 invoked"

338

339 **ASM_6008_TestCase**

340

Testcase ID	ASM_6008_TestCase
Test Assertion	ASM-TA-6012
Description	Tests that where a <composite/> <reference/> has no <interface/> subelement, and that <reference/> promotes 2 or more <component/> <reference/>s, that the <interface/> declared on each of the <component/> <reference/>s are the same
Artifacts	ASM_6008_TestCase.java Test_ASM_6008.composite TestInvocation.wsdl TestClient_0002.composite TestComposite22.composite TestComposite1.composite
Expected output	Negative test: "exception"

341

342 **ASM_6009_TestCase**

343

Testcase ID	ASM_6009_TestCase
Test Assertion	ASM-TA-6015
Description	Tests that a <component/> <reference/> with @multiplicity=0..1 can be promoted by a <composite/> <reference/> with @multiplicity=1..1
Artifacts	ASM_6009_TestCase.java

	Test_ASM_6009.composite TestInvocation.wsdl TestClient_0002.composite TestComposite23.composite TestComposite1.composite
Expected output	Positive test: "ASM_6009 request service1 operation1 invoked service2 operation1 invoked"

344

345 **ASM_6010_TestCase**

346

Testcase ID	ASM_6010_TestCase
Test Assertion	ASM-TA-6017
Description	Tests that a <component/> <reference/> with @multiplicity=0..n can be promoted by a <composite/> <reference/> with @multiplicity=0..1
Artifacts	ASM_6010_TestCase.java Test_ASM_6010.composite TestInvocation.wsdl TestClient_0002.composite TestComposite24.composite TestComposite1.composite
Expected output	Positive test: "ASM_6010 request service1 operation1 invoked service2 operation1 invoked"

347

348

349 **ASM_6011_TestCase**

350

Testcase ID	ASM_6011_TestCase
Test Assertion	ASM-TA-6018
Description	Tests that a <component/> <reference/> with @multiplicity=1..n can be promoted by a <composite/> <reference/> with @multiplicity=1..1
Artifacts	ASM_6011_TestCase.java Test_ASM_6011.composite TestInvocation.wsdl

	TestClient_0002.composite TestComposite25.composite TestComposite1.composite
Expected output	Positive test: "ASM_6011 request service1 operation1 invoked service2 operation1 invoked"

351

352

353 **ASM_6012_TestCase**

354

Testcase ID	ASM_6012_TestCase
Test Assertion	ASM-TA-6019
Description	Tests that the @name attribute of a <property/> element of a <composite/> is unique amongst the @name attribute values of all the <property/> elements of that <composite/>
Artifacts	ASM_6012_TestCase.java Test_ASM_6012.composite TestInvocation.wsdl TestClient_0002.composite TestComposite26.composite TestComposite1.composite
Expected output	Negative test: "exception"

355

356

357 **ASM_6013_TestCase**

358

Testcase ID	ASM_6013_TestCase
Test Assertion	ASM-TA-6020
Description	Tests that a <component/> <reference/> with @autowire=true is wired to a compatible <component/> <service/> in the same <composite/>
Artifacts	ASM_6013_TestCase.java Test_ASM_6013.composite TestInvocation.wsdl TestClient_0002.composite

	TestComposite27.composite
Expected output	Positive test: "ASM_6013 request service1 operation1 invoked third operation1 invoked"

ASM_6014_TestCase

Testcase ID	ASM_6014_TestCase
Test Assertion	ASM-TA-6021
Description	Tests that a <component/> <reference/> with @autowire=true is wired to a <component/> <service/> with an interface which is a compatible superset of the interface on the <reference/> in the same <composite/>
Artifacts	ASM_6014_TestCase.java Test_ASM_6014.composite TestInvocation.wsdl TestClient_0002.composite TestComposite28.composite
Expected output	Positive test: "ASM_6014 request service1 operation1 invoked third operation1 invoked"

ASM_6015_TestCase

Testcase ID	ASM_6015_TestCase
Test Assertion	ASM-TA-6022
Description	Tests that a <component/> <reference/> with @autowire=true and with some intents attached using its @requires attribute is wired to a compatible <component/> <service/> in the same <composite/>, where that <service/> has the same intents attached using its @requires attribute
Artifacts	ASM_6015_TestCase.java Test_ASM_6015.composite TestInvocation.wsdl TestClient_0002.composite TestComposite29.composite Service1.wsdl
Expected output	Positive test:

"ASM_6015-request-service1-operation1-invoked-third-operation1-invoked"

ASM_6016_TestCase

Testcase ID	ASM_6016_TestCase
Test Assertion	ASM-TA-6023
Description	Tests that a <component/> <reference/> with @autowire=true and with some intents attached using its @requires attribute is not wired to a compatible <component/> <service/> in the same <composite/>, where that <service/> has different intents attached using its @requires attribute
Artifacts	ASM_6016_TestCase.java Test_ASM_6016.composite TestInvocation.wsdl TestClient_0002.composite TestComposite30.composite
Expected output	Negative test: "exception"

ASM_6017_TestCase

Testcase ID	ASM_6017_TestCase
Test Assertion	ASM-TA-6024
Description	Tests that a <component/> <reference/> with @autowire=true and @multiplicity="0..1" is wired to one compatible <component/> <service/> in the same <composite/>
Artifacts	ASM_6017_TestCase.java Test_ASM_6017.composite TestInvocation.wsdl TestClient_0002.composite TestComposite8.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test with 2 alternative outputs:

	"ASM_6017 request service1 operation1 invoked second operation1 invoked"
	"ASM_6017 request service1 operation1 invoked third operation1 invoked"

375

376

377 **ASM_6018_TestCase**

378

Testcase ID	ASM_6018_TestCase
Test Assertion	ASM-TA-6025
Description	Tests that a <component/> <reference/> with @autowire=true and @multiplicity="0..n" is wired to all compatible <component/> <service/>s in the same <composite/>
Artifacts	ASM_6018_TestCase.java Test_ASM_6018.composite TestInvocation.wsdl TestClient_0002.composite TestComposite7.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_6018 request service1 operation1 invoked second operation1 invoked third operation1 invoked fourth operation1 invoked"

379

380 **ASM_6019_TestCase**

381

Testcase ID	ASM_6019_TestCase
Test Assertion	ASM-TA-6026
Description	Tests that a <component/> <reference/> with @autowire=true and @multiplicity="0..n" runs successfully when there are no compatible <component/> <service/>s in the same <composite/>
Artifacts	ASM_6019_TestCase.java Test_ASM_6019.composite TestInvocation.wsdl TestClient_0002.composite TestComposite7.composite Service1.wsdl

Expected output	Positive test: "ASM_6019 request service1 operation1 invoked"
-----------------	--

ASM_6020_TestCase

Testcase ID	ASM_6020_TestCase
Test Assertion	ASM-TA-6027
Description	Tests that a <component/> <reference/> with @autowire=true and @multiplicity="1..n" fails and causes an error to be raised when there are no compatible <component/> <service/>s in the same <composite/>
Artifacts	ASM_6020_TestCase.java Test_ASM_6020.composite TestInvocation.wsdl TestClient_0002.composite TestComposite6.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_6021_TestCase

Testcase ID	ASM_6021_TestCase
Test Assertion	ASM-TA-6028
Description	Tests that a <composite/> <service/> @promote attribute contains the URI of a <component/> <service/> in the <composite/>
Artifacts	ASM_6021_TestCase.java Test_ASM_6021.composite TestInvocation.wsdl TestClient_0002.composite TestComposite31.composite
Expected output	Negative test: "exception"

ASM_6022_TestCase

Testcase ID	ASM_6022_TestCase
Test Assertion	ASM-TA-6029
Description	Tests that a <component/> <reference/> with multiplicity ="1..n" is wired to a <service/> in the same <composite/> or is promoted by a <composite> <reference/>, when the containing <composite/> is used as an <implementation/>
Artifacts	ASM_6022_TestCase.java Test_ASM_6022.composite TestInvocation.wsdl TestClient_0002.composite TestComposite32.composite
Expected output	Negative test: "exception"

ASM_6023_TestCase

Testcase ID	ASM_6023_TestCase
Test Assertion	ASM-TA-6030
Description	Tests that a <component/> <property/> which is marked with @mustSupply="true" in the componentType of the <component/> <implementation/> is given a value
Artifacts	ASM_6023_TestCase.java Test_ASM_6023.composite TestInvocation.wsdl TestClient_0002.composite TestComposite33.composite
Expected output	Negative test: "exception"

ASM_6024_TestCase

Testcase ID	ASM_6024_TestCase
Test Assertion	ASM-TA-6031
Description	Tests that the @name attribute of an <implementation.composite/>

	element is the QName of a <composite/> in the SCA Domain
Artifacts	ASM_6024_TestCase.java Test_ASM_6024.composite TestInvocation.wsdl TestClient_0002.composite
Expected output	Negative test: "exception"

398

399 **ASM_6025_TestCase**

400

Testcase ID	ASM_6025_TestCase
Test Assertion	ASM-TA-6032
Description	Tests that when an <include/> element is processed and results in an invalid <composite/> configuration, then the SCA runtime raises an error and does not run the resulting composite
Artifacts	ASM_6025_TestCase.java Test_ASM_6025.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite34.composite
Expected output	Negative test: "exception"

401

402 **ASM_6026_TestCase**

403

Testcase ID	ASM_6026_TestCase
Test Assertion	ASM-TA-6035 ASM-TA-6036
Description	Tests that when a @promote attribute of a <composite/> <reference/> has a URI for a <component/> <reference/> that is present in a <composite> which is referenced through an <include/> element in the first <composite/> and also when a @promote attribute of a <composite/> <service/> has a URI for a <component/> <service/> present in the same included <composite/> then the promotions succeed and the <component/> <service/> can be invoked and the <component/> is able to invoke the reference

Artifacts	ASM_6026_TestCase.java Test_ASM_6026.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite TestComposite35.composite TestComposite36.composite Service1.wsdl
Expected output	Positive test: "ASM_6026 request service1 operation1 invoked service2 operation1 invoked"

404

405 **ASM_6027_TestCase**

406

Testcase ID	ASM_6027_TestCase
Test Assertion	ASM-TA-6037
Description	Tests that when a first <composite/> contains a <wire/> element which has a @source attribute referencing a <component/> <reference/> in a second <composite/> and a @target attribute referencing a <component/> <service/> element also in that second <composite/>, where the second composite is referenced by an <include/> element in the first <composite/>, then the wire successfully connects the <reference/> to the <service/> and the client component can invoke operations on the service.
Artifacts	ASM_6027_TestCase.java Test_ASM_6027.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite TestComposite37.composite
Expected output	Positive test: "ASM_6027 request service1 operation1 invoked service2 operation1 invoked"

407

ASM_6028_TestCase

Testcase ID	ASM_6028_TestCase
Test Assertion	ASM-TA-6038 ASM-TA-6039
Description	Tests that when a <composite/> <property/> element has a @type attribute declared it does not also have a @element attribute declared
Artifacts	ASM_6028_TestCase.java Test_ASM_6028.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite
Expected output	Negative test: "exception"

ASM_6029_TestCase

Testcase ID	ASM_6029_TestCase
Test Assertion	ASM-TA-6040
Description	Tests that when a <composite/> has an <include/> which references a second <composite/> which has @local="true", then the first composite also has @local="true"
Artifacts	ASM_6029_TestCase.java Test_ASM_6029.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite4.composite TestComposite38.composite
Expected output	Negative test: "exception"

ASM_6030_TestCase

Testcase ID	ASM_6030_TestCase
Test Assertion	ASM-TA-6041
Description	Tests that when a <composite/> has an <include/> element, the @name of the <include/> references the QName of a composite in the Domain
Artifacts	ASM_6030_TestCase.java Test_ASM_6030.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite
Expected output	Negative test: "exception"

416

417 ASM_6031_TestCase

418

Testcase ID	ASM_6031_TestCase
Test Assertion	ASM-TA-6013
Description	Tests that when a <composite/> <reference/> has @requires with intents declared and the <component/> <reference/> promoted by the composite reference has @requires with intents declared, the two sets of intents do not contain a mutually exclusive pair of intents
Artifacts	ASM_6031_TestCase.java Test_ASM_6031.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite39.composite
Expected output	Negative test: "exception"

419

420 ASM_6032_TestCase

421

Testcase ID	ASM_6032_TestCase
Test Assertion	ASM-TA-6014
Description	Tests that when a <composite/> <reference/> has @requires with 2 or more intents in the declared set, that the set does not contain 2 mutually exclusive intents

Artifacts	ASM_6032_TestCase.java Test_ASM_6032.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite40.composite
Expected output	Negative test: "exception"

422

423 **ASM_6033_TestCase**

424

Testcase ID	ASM_6033_TestCase
Test Assertion	ASM-TA-6016
Description	Tests that when a <composite/> <reference/> element promotes a <component/> <reference/> element with @multiplicity="1..1", the @multiplicity of the <composite/> <reference/> is also "1..1"
Artifacts	ASM_6033_TestCase.java Test_ASM_6033.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestComposite41.composite
Expected output	Negative test: "exception"

425

426 **ASM_6034_TestCase**

427

<u>Testcase ID</u>	<u>ASM_6034_TestCase</u>
<u>Test Assertion</u>	<u>ASM-TA-6042</u>
<u>Description</u>	<u>Tests that when a <wire/> element connects a component <reference/> to a component <service/>, that the interface of the service is a compatible superset of the interface of the reference</u>
<u>Artifacts</u>	<u>ASM_6034_TestCase.java</u> <u>Test_ASM_6034.composite</u> <u>TestInvocation.wsdl</u>

	TestClient_0002.composite TestComposite9.composite TestComposite4.composite
Expected output	Positive test: "ASM_6034 request service1 operation1 invoked service2 operation1 invoked"

428

429 [ASM_6035_TestCase](#)

430 **2.4 Section 7**

431 [ASM_7001_TestCase](#)

432

Testcase ID	ASM_6035_TestCase
Test Assertion	ASM-TA-6044
Description	Tests that when a component<reference/> element has a @target attribute containing a single target in the form "component-name/service-name" and the component with the name "component-name" has multiple services, each of which are compatible with the reference, and one of which has the name "service-name", that the reference is wired to the service with the name "service-name" and is able to invoke it.
Artifacts	ASM_6035_TestCase.java Test_ASM_6035.composite TestInvocation.wsdl TestClient_0002.composite TestComposite75.composite TestComposite1.composite
Expected output	Positive test: "ASM_6035 request serviceA operation1 invoked"
Testcase ID	ASM_7001_TestCase
Test Assertion	ASM-TA-7004 ASM-TA-7006
Description	Tests that when a <component/> has a @constrainingType attribute set, that the <component/> implements every <service/> element defined in the <constrainingType/>
Artifacts	ASM_7001_TestCase.java Test_ASM_7001.composite TestInvocation.wsdl

	TestClient_0002.composite Test_ASM_7001.constraint TestComposite1.composite Service1.wsdl
Expected-output	Negative test: "exception"

433

434 [ASM_60367002_TestCase](#)

435

Testcase ID	ASM_6036_TestCase
Test Assertion	ASM-TA-6045
Description	Tests that when a component<reference/> element has a @target attribute containing a single target in the form "component-name/service-name" and the component with the name "component-name" has multiple services, each of which are compatible with the reference, but none of them has the name "service-name", that the reference is not wired to any service and the SCA runtime raises an error
Artifacts	ASM_6036_TestCase.java Test_ASM_6036.composite TestInvocation.wsdl TestClient_0002.composite TestComposite75.composite TestComposite1.composite
Expected output	Negative test: "exception"
Testcase ID	ASM_7002_TestCase
Test Assertion	ASM-TA-7002 ASM-TA-7007
Description	Tests that when a <component/> has a @constrainingType attribute set, that the <component/> implements every <reference/> element defined in the <constrainingType/>
Artifacts	ASM_7002_TestCase.java Test_ASM_7002.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7002.constraint TestComposite1.composite

	TestComposite4.composite-
Expected-output	Negative-test: "exception"

ASM_60377003_TestCase

Testcase ID	ASM_6037_TestCase
Test Assertion	ASM-TA-6046
Description	Tests that when a component<reference/> element has a @target attribute containing a single target in the form "component-name" and the component with the name "component-name" has a single service with the name "service-name", which is compatible with the reference, that the reference is wired to the service and is able to invoke it
Artifacts	ASM_6037_TestCase.java Test_ASM_6037.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite
Expected output	Positive test: "ASM_6037 request service1 operation1 invoked"
Testcase ID	ASM_7003_TestCase
Test Assertion	ASM-TA-7003 ASM-TA-7008
Description	Tests that when a <component/> has a @constrainingType attribute set, that the <component/> implements every <property/> element defined in the <constrainingType/>
Artifacts	ASM_7003_TestCase.java Test_ASM_7003.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7003.constraint TestComposite1.composite-
Expected-output	Negative-test: "exception"

440 | **ASM_60387004_TestCase**

441

Testcase ID	ASM_6038_TestCase
Test Assertion	ASM-TA-6047
Description	Tests that when a component<reference/> element has a @target attribute containing a single target in the form "component-name/service-name/binding-name" and the component with the name "component-name" has a service with the name "service-name" with a binding with the name "binding-name", which is compatible with the reference, that the reference is wired to the service and is able to invoke it
Artifacts	ASM_6038_TestCase.java Test_ASM_6038.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite
Expected output	Positive test: "ASM_6038 request service1 operation1 invoked"
Testcase ID	ASM_7004_TestCase
Test Assertion	ASM-TA-7009
Description	Tests that when a <component/> has a @constrainingType attribute set, and the <component/> has an <implementation/> with a <componentType/> with one or more <service/> elements that are not in the <constrainingType/> that the <component/> runs successfully
Artifacts	ASM_7004_TestCase.java Test_ASM_7004.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7004.constraint TestComposite48.composite
Expected output	Positive test: "ASM_7004 request service1 operation1 invoked"

442 |

443 | **ASM_60397005_TestCase**

444

Testcase ID	ASM_6039_TestCase
Test Assertion	ASM-TA-6048
Description	Tests that when a component<reference/> element has a @target attribute containing a single target in the form "component-name/service-

	<u>name/binding-name" and the component with the name "component-name" has a service with the name "service-name" which is compatible with the reference, but which does not have a binding with the name "binding-name", that the reference is not wired to the service and the SCA runtime raises an exception</u>
<u>Artifacts</u>	<u>ASM_6039_TestCase.java</u> <u>Test_ASM_6039.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite1.composite</u>
<u>Expected output</u>	Negative test: "exception"
<u>Testcase ID</u>	ASM_7005_TestCase
<u>Test Assertion</u>	ASM-TA-7010
<u>Description</u>	Tests that when a <component/> has a @constrainingType attribute set, and the <component/> has an <implementation/> with a <componentType/> with one or more <reference/> elements that are not in the <constrainingType/> that those <reference/> elements have multiplicity of 0..1 or 0..n
<u>Artifacts</u>	<u>ASM_7005_TestCase.java</u> <u>Test_ASM_7005.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Test_ASM_7004.constraint</u> <u>TestComposite49.composite</u>
<u>Expected output</u>	Positive test: "ASM_7005-request-service1-operation1-invoked"

445

446 **ASM_60407006_TestCase**

447

<u>Testcase ID</u>	ASM_6040_TestCase
<u>Test Assertion</u>	ASM-TA-6049
<u>Description</u>	<u>Tests that when a component<reference/> element has a @target attribute containing a single target in the form "component-name/service-name" and the reference has a policy requirement attached, and the component with the name "component-name" has a service with the name "service-name" with multiple bindings one of which has the same policy requirement as the reference, that the reference is wired to the service using that binding and is able to invoke it</u>
<u>Artifacts</u>	<u>ASM_6040_TestCase.java</u>

	Test_ASM_6040.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite
Expected output	Positive test: "ASM_6040 request service1 operation1 invoked"
Testcase ID	ASM_7006_TestCase
Test Assertion	ASM-TA-7011
Description	Tests that when a <component/> has a @constrainingType attribute set, and the <component/> has an <implementation/> with a <componentType/> with one or more <property/> elements that are not in the <constrainingType/> that those <property/> elements have @mustSupply="false"
Artifacts	ASM_7006_TestCase.java Test_ASM_7006.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7004.constraint TestComposite50.composite
Expected output	Positive test: "ASM_7006 request service1 operation1 invokedData1Data2"

448 [ASM_7007_TestCase](#)

449

Testcase ID	ASM_7007_TestCase
Test Assertion	ASM-TA-7012
Description	Tests that when a <component/> has a @constrainingType attribute set, and the <component/> has an <implementation/> with a <componentType/> with one or more <reference/> elements that are not in the <constrainingType/> that those <reference/> elements are not wired by the component
Artifacts	ASM_7007_TestCase.java Test_ASM_7007.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7004.constraint TestComposite1.composite TestComposite49.composite
Expected output	Negative test:

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

450

451 **2.5 Section 7**

452 **ASM_7008_TestCase**

453

Testcase ID	ASM_7008_TestCase
Test Assertion	ASM-TA-7013
Description	Tests that when a <component/> has a @constrainingType attribute set, and the <component/> has an <implementation/> with a <componentType/> with one or more <service/> elements that are not in the <constrainingType/> that those <service/> elements are not promoted by the composite containing the component
Artifacts	ASM_7008_TestCase.java Test_ASM_7008.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7004.constraint TestComposite48.composite
Expected output	Negative test: "exception"

454 **ASM_7009_TestCase**

455

Testcase ID	ASM_7009_TestCase
Test Assertion	ASM-TA-7014
Description	Tests that when a <component/> has a @constrainingType attribute set, and the <component/> has an <implementation/> with a <componentType/> with one or more <service/> elements that are not in the <constrainingType/> that those <reference/> elements are not promoted by the composite containing the component
Artifacts	ASM_7009_TestCase.java Test_ASM_7009.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7004.constraint TestComposite49.composite
Expected output	Negative test:

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

ASM_7010_TestCase

Testcase ID	ASM_7010_TestCase
Test Assertion	ASM-TA-7005
Description	Tests that where the SCA Domain contains 2 or more <constrainingType/> documents, that the @name attribute value of any <constrainingType/> element is not the same as the @name attribute value of any other <constrainingType/> element
Artifacts	ASM_7010_TestCase.java Test_ASM_7010.composite TestInvocation.wsdl TestClient_0002.composite Test_ASM_7010_A.constraint Test_ASM_7010_B.constraint TestComposite49.composite
Expected-output	Negative test: "exception"

2.6 Section 8

ASM_8001_TestCase

Testcase ID	ASM_8001_TestCase
Test Assertion	ASM-TA-8001
Description	Tests that when an <interface/> element references an interface document which is marked remotable, that the interface document does not contain the declaration of 2 operations with the same name
Artifacts	ASM_8001_TestCase.java Test_ASM_8001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite51.composite

	Service3.wsdl
Expected output	Negative test: "exception"

464

465 **ASM_8002_TestCase**

466

Testcase ID	ASM_8002_TestCase
Test Assertion	ASM-TA-8002
Description	Tests that when an <interface/> element references an interface document which is marked remotable, that the value of an input parameter of an operation in the interface in the client component implementation is the same after the operation completes as immediately before invoking the operation
Artifacts	ASM_8002_TestCase.java Test_ASM_8002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite52.composite TestComposite53.composite Service1.wsdl Service4.wsdl
Expected output	Positive test: "ASM_8002 request service1 operation1 invoked"

467

468 **ASM_8003_TestCase**

469

Testcase ID	ASM_8003_TestCase
Test Assertion	ASM-TA-8004 ASM-TA-8007
Description	Tests that when an <interface/> element of a <component/> <reference/> has an interface that is marked bidirectional, that the invoked <service/> <implementation/> can invoke an operation of the callback interface on the <implementation/> which has the <reference/>
Artifacts	ASM_8003_TestCase.java Test_ASM_8003.composite TestInvocation.wsdl

	TestClient_0002.composite TestComposite54.composite TestComposite55.composite Service1.wsdl Service5.wsdl Service5Callback.wsdl
Expected output	Positive test: "ASM_8003 request service1 operation1 invoked service2 operation1 invoked service1 callback1 invoked"

470

471 **ASM_8004_TestCase**

472

Testcase ID	ASM_8004_TestCase
Test Assertion	ASM-TA-8005
Description	Tests that when an <interface/> element of a <component/> <reference/> has an interface that is marked bidirectional, where the forward interface is marked remotable, then the callback interface is also marked remotable
Artifacts	ASM_8004_TestCase.java Test_ASM_8004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite56.composite TestComposite57.composite Service1.wsdl Service6Callback.java Service6.java
Expected output	Negative test: "exception"

473

474 **ASM_8005_TestCase**

475

Testcase ID	ASM_8005_TestCase
Test Assertion	ASM-TA-8006
Description	Tests that when an <interface/> element of a <component/> <reference/> has an interface that is marked bidirectional, where the forward interface is

	marked local, then the callback interface is also marked local
Artifacts	ASM_8005_TestCase.java Test_ASM_8005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite58.composite TestComposite59.composite Service1.wsdl Service7Callback.java Service7.java
Expected output	Negative test: "exception"

476

477 **ASM_8006_TestCase**

478

Testcase ID	ASM_8006_TestCase
Test Assertion	ASM-TA-8008
Description	Tests that when an <interface/> element of a <component/> <reference/> explicitly declares both a forward interface and a callback interface and the forward interface document also declares a callback interface, that the two callback interfaces are compatible
Artifacts	ASM_8006_TestCase.java Test_ASM_8006.composite TestInvocation.wsdl TestClient_0002.composite TestComposite60.composite TestComposite61.composite Service1.wsdl Service9.wsdl Service8Callback.wsdl
Expected output	Negative test: "exception"

479

ASM_8007_TestCase

Testcase ID	ASM_8007_TestCase
Test Assertion	ASM-TA-8009
Description	Tests that when an <interface/> element of a <component/> <service/> explicitly declares both a forward interface and a callback interface that the callback interface is compatible with the callback interface of the <componentType/> <service/> element of the <implementation/> used by the <component/>
Artifacts	ASM_8007_TestCase.java Test_ASM_8007.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite TestComposite55.composite Service1.wsdl Service5.wsdl Service5Callback.wsdl Service8Callback.wsdl
Expected output	Negative test: "exception"

ASM_8008_TestCase

Testcase ID	ASM_8008_TestCase
Test Assertion	ASM-TA-8010
Description	Tests that when an <interface/> element of a <component/> <reference/> explicitly declares both a forward interface and a callback interface that the callback interface is compatible with the callback interface of the <componentType/> <reference/> element of the <implementation/> used by the <component/>
Artifacts	ASM_8008_TestCase.java Test_ASM_8008.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite TestComposite55.composite

	Service1.wsdl Service5.wsdl Service5Callback.wsdl Service8Callback.wsdl
Expected output	Negative test: "exception"

485

486 **ASM_8009_TestCase**

487

Testcase ID	ASM_8009_TestCase
Test Assertion	ASM-TA-8011
Description	Tests that when an <interface/> element of an <implementation/> <componentType/> <reference/> or <service/> does not declare a callback interface that the <component/> <reference/> or <service/> element of the <component/> using the <implementation/> also does not declare a callback interfaceTests that where the <interface/> element of a reference in the componentType of an implementation does not declare a callback interface, then the corresponding reference element of the component using the implementation also does not declare a callback interface in its <interface/> subelement
Artifacts	ASM_8009_TestCase.java Test_ASM_8009.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite TestComposite55.composite Service1.wsdl Service5Callback.java
Expected output	Negative test: "exception"

488

489 **ASM_8010_TestCase**

490

Testcase ID	ASM_8010_TestCase
Test Assertion	ASM-TA-8012
Description	Tests that when an <interface/> element of a <component/> <service/> declares a callback interface that the <composite/> <service/> element

	which promotes the component service declares a compatible callback interface
Artifacts	ASM_8010_TestCase.java Test_ASM_8010.composite TestInvocation.wsdlTestClient_0002.composite TestComposite54.composite TestComposite63.composite Service1.wsdl Service5.wsdl Service5Callback.wsdl Service8Callback.wsdl
Expected output	Negative test: "exception"

491

492 **ASM_8011_TestCase**

493

Testcase ID	ASM_8011_TestCase
Test Assertion	ASM-TA-8013
Description	Tests that when an <interface/> element of a <component/> <reference/> declares a callback interface that the <composite/> <reference/> element which promotes the component service declares a compatible callback interface
Artifacts	ASM_8011_TestCase.java Test_ASM_8011.composite TestInvocation.wsdl TestClient_0002.composite TestComposite62.composite TestComposite55.composite Service5.wsdl Service5Callback.wsdl
Expected output	Negative test: "exception"

494

495 **ASM_8012_TestCase**

496

Testcase ID	ASM_8012_TestCase
Test Assertion	ASM-TA-8014
Description	Tests that when an <interface/> element of a <component/> <reference/> does not declare a callback interface that the <composite/> <reference/> element which promotes the <component/> <reference/> also does not declare a callback interface
Artifacts	ASM_8012_TestCase.java Test_ASM_8012.composite TestInvocation.wsdl TestClient_0002.composite TestComposite64.composite Service1.wsdl
Expected output	Negative test: "exception"

497

498 ASM_8013_TestCase

499

Testcase ID	ASM_8013_TestCase
Test Assertion	ASM-TA-8015
Description	Tests that where the <implementation/> of a component with a <service/> with an <interface/> with a callback, the implementation can invoke operations of the callback interface an arbitrary number of times
Artifacts	ASM_8013_TestCase.java Test_ASM_8013.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite TestComposite65.composite Service1.wsdl Service5.wsdl Service5Callback.wsdl
Expected output	Positive test: "ASM_8013 request service1 operation1 invoked service2 operation1 invoked service1 callback1 invoked service1 callback1 invoked service1 callback1 invoked"

500

ASM_8014_TestCase

Testcase ID	ASM_8014_TestCase
Test Assertion	ASM-TA-8018
Description	Tests that when an <interface/> element of a <service/> or <reference/> has one or more intents attached to it, those intents apply to the <service/> or <reference/>
Artifacts	ASM_8014_TestCase.java Test_ASM_8014.composite TestInvocation.wsdl TestClient_0002.composite TestComposite4.composite TestComposite66.composite Service1.wsdl Service1_Intent.wsdl
Expected output	Positive test: "ASM_8018 request service1 operation1 invoked service2 operation1 invoked"

ASM_8015_TestCase

Testcase ID	ASM_8015_TestCase
Test Assertion	ASM-TA-8020
Description	Tests that when an <interface.wsdl/> element is used to declare an interface, that the @interface attribute of the element references a WSDL 1.1 portType
Artifacts	ASM_8015_TestCase.java Test_ASM_8015.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite TestInvocation.wsdl Service1.wsdl
Expected output	Positive test: "ASM_8015 request service1 operation1 invoked"

ASM_8016_TestCase

Testcase ID	ASM_8016_TestCase
Test Assertion	ASM-TA-8020
Description	Tests that when an <interface.wsdl/> element is used to declare a callback interface, that the @callbackInterface attribute of the element references a WSDL 1.1 portType
Artifacts	ASM_8016_TestCase.java Test_ASM_8016.composite TestInvocation.wsdl TestClient_0002.composite TestComposite54.composite TestComposite55.composite ASM_0002_Client.java Service1.wsdl Service5.wsdl Service5Callback.wsdl
Expected output	Positive test: "ASM_8016 request service1 operation1 invoked service2 operation1 invoked service1 callback1 invoked"

ASM_8017_TestCase

Testcase ID	ASM_8017_TestCase
Test Assertion	ASM-TA-8014
Description	Tests that when an <interface/> element of a <component/> <service/> does not declare a callback interface that the <composite/> <service/> element which promotes the <component/> <service/> also does not declare a callback interface
Artifacts	ASM_8017_TestCase.java Test_ASM_8017.composite TestInvocation.wsdl TestClient_0002.composite TestComposite64.composite Service1.wsdl
Expected output	Negative test:

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

512

513 **ASM_8018_TestCase**

514

Testcase ID	ASM_8018_TestCase
Test Assertion	ASM-TA-8011
Description	Tests that where the <interface/> element of a service in the componentType of an implementation does not declare a callback interface, then the corresponding service element of the component using the implementation also does not declare a callback interface in its <interface/> subelement
Artifacts	ASM_8018_TestCase.java Test_ASM_8018.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl Service5Callback.wsdl
Expected output	Negative test: "exception"

515

516 **ASM_8019_TestCase**

517

Testcase ID	ASM_8019_TestCase
Test Assertion	ASM-TA-8022
Description	Tests that an <interface.wsdl/> element does not contain @remotable="false"
Artifacts	ASM_8019_TestCase.java Test_ASM_8019.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

518

519

520 **2.7 Section 9**521 **ASM_9001_TestCase**

522

Testcase ID	ASM_9001_TestCase
Test Assertion	ASM-TA-9002
Description	Tests that if a <binding/> element of a <reference/> element has a @uri attribute set, that it points to a service endpoint.
Artifacts	ASM_9001_TestCase.java Test_ASM_9001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_9001 request service1 operation1 invoked"

523

524 **ASM_9002_TestCase**

525

Testcase ID	ASM_9002_TestCase
Test Assertion	ASM-TA-9003
Description	Tests that if a <service/> element has multiple <binding/> subelements, that the @name attribute of each <binding/> element must be unique amongst the binding elements of that <service/>
Artifacts	ASM_9002_TestCase.java Test_ASM_9002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

526

ASM_9003_TestCase

Testcase ID	ASM_9003_TestCase
Test Assertion	ASM-TA-9004
Description	Tests that if a <binding/> element of a <reference/> element does not identify a target service using binding specific mechanisms, that it has a @uri attribute which resolves to a valid service endpoint
Artifacts	ASM_9003_TestCase.java Test_ASM_9003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_9003 request service1 operation1 invoked"

ASM_9004_TestCase

Testcase ID	ASM_9004_TestCase
Test Assertion	ASM-TA-9005
Description	Tests that if a <binding/> element of a <reference/> element is wired to a <service/> element which has multiple <binding/> subelements, each with a @name attribute set, that the @target attribute of the <reference/> uses the form "componentName/serviceName/bindingName" to select one specific binding for the wire
Artifacts	ASM_9004_TestCase.java Test_ASM_9004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Positive test: "ASM_9004 request service1 operation1 invoked"

ASM_9005_TestCase

Testcase ID	ASM_9005_TestCase
Test Assertion	ASM-TA-9006
Description	Tests that if a component service has a <binding.sca/> subelement, that the binding.sca element does not declare a @uri attribute
Artifacts	ASM_9005_TestCase.java Test_ASM_9005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

2.8 Section 10

ASM_10001_TestCase

Testcase ID	ASM_10001_TestCase
Test Assertion	ASM-TA-10001
Description	Tests that the Qname of an <intent/> <policySet/> <bindingType/> or <implementationType/> subelement of a <definitions/> element is unique in the Domain
Artifacts	ASM_10001_TestCase.java Test_ASM_5002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl ASM_10001/definitions.xml
Expected output	Negative test: "exception"

ASM_10002_TestCase

Testcase ID	ASM_10002_TestCase
Test Assertion	ASM-TA-10002
Description	Tests that when a <definitions/> file contains an <intent/> subelement, that the intent is available for use by any element in the SCA Domain
Artifacts	ASM_10002_TestCase.java Test_ASM_10002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl General/META-INF/definitions.xml
Expected output	Positive test: "ASM_10002 request service1 operation1 invoked"

543

544 ASM_10003_TestCase

545

Testcase ID	ASM_10003_TestCase
Test Assertion	ASM-TA-10003
Description	Tests that when a <definitions/> file has content that does not conform to the sca-definitions.xsd, it is rejected by the SCA runtime
Artifacts	ASM_10003_TestCase.java Test_ASM_10003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl ASM_10003/META-INF/definitions.xml
Expected output	Negative test: "exception"

546

547 2.9 Section 12

548 ASM_12001_TestCase

549

Testcase ID	ASM_12001_TestCase
Test Assertion	ASM-TA-12002 ASM-TA-12003
Description	Tests that a contribution with a META-INF directory at the root, containing a file called sca-contribution.xml containing a <deployable/> element referencing a composite in the contribution, has that composite run when the contribution is deployed
Artifacts	ASM_12001_TestCase.java Test_ASM_12001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite- Service1.wsdl ASM_12001/META-INF/sca-contribution.xml
Expected output	Positive test: "ASM_12001 request service1 operation1 invoked"

550

551 ASM_12002_TestCase

552

Testcase ID	ASM_12002_TestCase
Test Assertion	ASM-TA-12005 ASM-TA-12002
Description	Tests that where a WSDL document references its schema using @schemaLocation attribute (schema artifact specific artifact resolution mechanism), that this is used to resolve the schema rather than SCA mechanisms.
Artifacts	ASM_12002_TestCase.java Test_ASM_12002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1a.wsdl Service1a.xsd (at location http://www.oasis-open.org/committees/download.php/32378/Service1a.xsd)
Expected output	Positive test: "ASM_12002 request service1 operation1 invoked"

553

554 **ASM_12003_TestCase**

555

Testcase ID	ASM_12003_TestCase
Test Assertion	ASM-TA-12006 ASM-TA-12012
Description	Tests that where a WSDL document references its schema using @schemaLocation attribute (schema artifact specific artifact resolution mechanism), and that this mechanism fails, then the SCA runtime reports an error
Artifacts	ASM_12003_TestCase.java Test_ASM_12003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1b.wsdl
Expected output	Negative test: "exception"

556

557 **ASM_12004_TestCase**

558

Testcase ID	ASM_12004_TestCase
Test Assertion	ASM-TA-12007
Description	Tests that the SCA runtime accepts and runs a contribution in ZIP format
Artifacts	ASM_12004_TestCase.java Test_ASM_12004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl ASM_12004.zip
Expected output	Positive test: "ASM_12004 request service1 operation1 invoked"

559

560 **ASM_12005_TestCase**

561

Testcase ID	ASM_12005_TestCase
Test Assertion	ASM-TA-12010
Description	<p>Tests that where multiple indirect dependent contributions export conflicting artifacts, the conflict is resolved by explicit import statements in the root contribution</p> <p>The concept of the test is that two composite files with the same name in 2 different contributions have different contents - and that only 1 of them will work with the root composite file</p>
Artifacts	ASM_12005_TestCase.java Test_ASM_12005.composite TestInvocation.wsdl TestClient_0002.composite TestComposite5.composite (version 1 - contribution = General) TestComposite5.composite (version 2 - contribution = Contribution1) Service1.wsdl
Expected output	Positive test: "ASM_12005 request servicex operation1 invoked"

562

563 **ASM_12006_TestCase**

564

Testcase ID	ASM_12006_TestCase
Test Assertion	ASM-TA-12013
Description	Tests that where a <component/> <reference/> is not wired by any explicit means, where the component is at the SCA Domain level that the reference remains unwired and cannot be invoked by the <component/> <implementation/>
Artifacts	ASM_12006_TestCase.java Test_ASM_12006.composite TestClient_0004.composite TestComposite1.composite ASM_0002_Client.java Service1.wsdl
Expected output	Positive test: "ASM_12006 request no invocation"

565

566 **ASM_12007_TestCase**

567

Testcase ID	ASM_12007_TestCase
Test Assertion	ASM-TA-12027
Description	Tests that where a contribution contains multiple <import/> declarations which refer to the namespace of a referenced artifact, that the artifact is found by searching target locations in the order in which the <import/> statements occur in the <contribution/> document
Artifacts	ASM_12007_TestCase.java Test_ASM_12007.composite TestInvocation.wsdl TestClient_0002.composite TestComposite5.composite (version 1 - contribution = General) TestComposite5.composite (version 2 - contribution = Contribution1) Service1.wsdl ASM_12007/META-INF/sca-contribution.xml
Expected output	Negative test: "exception"

568

569 **ASM_12008_TestCase**

570

Testcase ID	ASM_12008_TestCase
Test Assertion	ASM-TA-12028
Description	Tests that where a contribution contains an <import/> declaration which refers to the namespace of a referenced artifact and where the artifact is present in a second contribution which exports the namespace, that the artifact is resolved in the second contribution even though a copy of the artifact exists in the original contribution
Artifacts	ASM_12008_TestCase.java Test_ASM_12008.composite TestInvocation.wsdl TestClient_0002.composite TestComposite5a.composite (version 1 - contribution = General) TestComposite5.composite (version 2 - contribution = ASM_12008) Service1.wsdl

	ASM_12008/META-IF/sca-contribution.xml
Expected output	Negative test: "exception"

571

572 ASM_12009_TestCase

573

Testcase ID	ASM_12008_TestCase
Test Assertion	ASM-TA-12029
Description	Tests that where a contribution contains an <import/> declaration with a specified target contribution which refers to the namespace of a referenced artifact and where the artifact is NOT present in the target contribution which exports the namespace, that the artifact is resolved in the first contribution when a copy of the artifact exists in the original contribution
Artifacts	ASM_12009_TestCase.java Test_ASM_12009.composite TestInvocation.wsdl TestClient_0002.composite TestComposite66.composite Service1.wsdl Contribution1 ASM_12009/META-IF/sca-contribution.xml
Expected output	Positive test: "ASM_12009 request servicex operation1 invoked"

574

575

576 ASM_12010_TestCase

577

Testcase ID	ASM_12010_TestCase
Test Assertion	ASM-TA-12030
Description	Tests that where a first contribution resolves an artifact reference to an artifact in a second contribution, where the resolved artifact has a further artifact reference, that this reference is resolved through the import statements of the second contribution even if the first contribution has import statements (to different locations) for the same namespace
Artifacts	ASM_12010_TestCase.java Test_ASM_12010.composite TestInvocation.wsdl

	TestClient_0002.composite TestComposite67.composite (Contribution2) TestComposite5.composite (version 1 - contribution = General) TestComposite5.composite (version 2 - contribution = Contribution1) Service1.wsdl ASM_12010/META-IF/sca-contribution.xml Contribution2/META-INF/sca-contribution.xml
Expected output	Positive test: "ASM_12010 request servicex operation1 invoked"

578

579 **ASM_12011_TestCase**

580

Testcase ID	ASM_12011_TestCase
Test Assertion	ASM-TA-12031
Description	Tests that where a contribution contains an <import/> declaration with a specific target contribution which refers to the namespace of a referenced artifact and the artifact is neither present in the target contribution nor the original contribution, that the resolution fails and the SCA runtime reports an error even where that artifact is available in other contributions
Artifacts	ASM_12011_TestCase.java Test_ASM_12011.composite TestInvocation.wsdl TestClient_0002.composite TestComposite68.composite Service1.wsdl ASM_12010/META-IF/sca-contribution.xml
Expected output	Negative test: "exception"

581

582 **ASM_12012_TestCase**

583

Testcase ID	ASM_12012_TestCase
Test Assertion	ASM-TA-12032
Description	Tests that where a contribution contains an sca-contribution.xml file that violates the schema sca-contribution.xsd, that the SCA runtime rejects the file and raises an error

Artifacts	ASM_12012_TestCase.java Test_ASM_12012.composite TestInvocation.wsdl TestClient_0002.composite TestComposite5.composite (version 1 - contribution = General) Service1.wsdl ASM_12012/META-IF/sca-contribution.xml
Expected output	Negative test: "exception"

584

585 ASM_12013_TestCase

586

Testcase ID	ASM_12013_TestCase
Test Assertion	ASM-TA-12035
Description	Tests that where a contribution has an artifact with an artifact dependency which is resolved via an <import/> declaration to an artifact in a second contribution and the resolved artifact has a further dependency on an artifact that is present in both the original contribution and in the second contribution that the dependency of the resolved artifact is resolved to the artifact in the second contribution
Artifacts	ASM_12013_TestCase.java Test_ASM_12013.composite TestInvocation.wsdl TestClient_0002.composite TestComposite69.composite TestComposite1.composite (version 1 - contribution = General) TestComposite1.composite (version 2 - contribution = ASM_12013) Service1.wsdl Service1.java service1Impl.java ASM_12013/META-IF/sca-contribution.xml
Expected output	Positive test: "ASM_12013 request service1 operation1 invoked"

587

588

2.10 Section 13

ASM_13001_TestCase

Testcase ID	ASM_13001_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13001_TestCase.java Test_ASM_13001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_13002_TestCase

Testcase ID	ASM_13002_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13002_TestCase.java Test_ASM_13002.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_13003_TestCase

Testcase ID	ASM_13003_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13003_TestCase.java Test_ASM_13003.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_13004_TestCase

Testcase ID	ASM_13004_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13004_TestCase.java Test_ASM_13004.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

ASM_13005_TestCase

Testcase ID	ASM_13005_TestCase
-------------	--------------------

Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13005_TestCase.java Test_ASM_13005.composite TestInvocation.wsdl TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

604

605 ASM_13006_TestCase

606

Testcase ID	ASM_13006_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13006_TestCase.java Test_ASM_13006.composite TestClient_0002.composite TestComposite1.composite Service1.wsdl ** test.TestInvocation.java **
Expected output	Negative test: "exception"

607

608

609 ASM_13007_TestCase

610

Testcase ID	ASM_13007_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-

	implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13007_TestCase.java Test_ASM_13007.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl
Expected output	Negative test: "exception"

611

612 ASM_13008_TestCase

613

Testcase ID	ASM_13008_TestCase
Test Assertion	ASM-TA-13001
Description	Tests that the SCA runtime rejects an invalid composite file which violates one of the SCA schemas (sca-core.xsd, sca-interface-wsdl.xsd, sca-implementation-composite.xsd and sca-binding-sca.xsd).
Artifacts	ASM_13008_TestCase.java Test_ASM_13008.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl
Expected output	Negative test: "exception"

614

615

616 2.11 Section 14

617 No testcases

3 Cross Mapping of Test Assertions to TestCases

Test Assertion	Test Cases
ASM-TA-4002	ASM_4002_TestCase
ASM-TA-4003	ASM_4003_TestCase
ASM-TA-4004	ASM_4004_TestCase
ASM-TA-4005	Untestable - wiredByImpl
ASM-TA-4006	ASM_4005_TestCase
ASM-TA-4007	ASM_4006_TestCase
ASM-TA-4008	ASM_5002_TestCase
ASM-TA-4009	ASM_4007_TestCase
ASM-TA-4010	ASM_4007_TestCase
ASM-TA-4011	ASM_4008_TestCase

Test Assertion	Test Cases
ASM-TA-5001	ASM_5005_TestCase
ASM-TA-5002	ASM_5006_TestCase
ASM-TA-5003	ASM_5007_TestCase
ASM-TA-5004	ASM_5008_TestCase
ASM-TA-5005	ASM_5009_TestCase
ASM-TA-5006	ASM_5010_TestCase
ASM-TA-5007	ASM_5032_TestCase
ASM-TA-5008	ASM_5033_TestCase
ASM-TA-5009	ASM_5034_TestCase
ASM-TA-5010	ASM_5011_TestCase
ASM-TA-5011	ASM_5012_TestCase
ASM-TA-5012	ASM_5013_TestCase
	ASM_5014_TestCase
ASM-TA-5013	ASM_5015_TestCase
ASM-TA-5014	Untestable - wiredByImpl
ASM-TA-5015	Untestable - wiredByImpl
ASM-TA-5016	ASM_5016_TestCase

ASM-TA-5017	ASM_5017_TestCase
ASM-TA-5018	ASM_5018_TestCase
ASM-TA-5019	Untestable - wiredByImpl
ASM-TA-5020	Untestable - wiredByImpl
ASM-TA-5021	ASM_5019_TestCase
ASM-TA-5022	ASM_5020_TestCase
ASM-TA-5023	ASM_5021_TestCase
ASM-TA-5024	ASM_5018_TestCase
ASM-TA-5025	ASM_5002_TestCase
ASM-TA-5026	ASM_5003_TestCase
ASM-TA-5027	ASM_5004_TestCase
ASM-TA-5028	ASM_5022_TestCase
ASM-TA-5029	ASM_5023_TestCase
ASM-TA-5030	ASM_5024_TestCase
ASM-TA-5031	ASM_5002_TestCase
ASM-TA-5032	ASM_5025_TestCase
ASM-TA-5033	ASM_5026_TestCase
ASM-TA-5034	ASM_5027_TestCase
ASM-TA-5036	ASM_5029_TestCase
ASM-TA-5037	ASM_5030_TestCase
ASM-TA-5038	ASM_5031_TestCase
ASM-TA-5039	ASM_5035_TestCase
ASM-TA-5040	ASM_5035_TestCase
ASM-TA-5041	ASM_5036_TestCase
ASM-TA-5042	ASM_5037_TestCase
ASM-TA-5043	Requires multiple bindings - leave to Bindings TC to devise a test
ASM-TA-5031	ASM_5038_TestCase
ASM-TA-5045	ASM_5039_TestCase
ASM-TA-5046	ASM_5025_TestCase ASM_5036_TestCase
ASM-TA-5047	ASM_5040_TestCase
ASM-TA-5048	ASM_5041_TestCase
ASM-TA-5049	ASM_5042_TestCase
ASM-TA-5050	Untestable

Test Assertion	Test Cases
ASM-TA-5001	ASM_5005_TestCase
ASM-TA-5002	ASM_5006_TestCase
ASM-TA-5003	ASM_5007_TestCase
ASM-TA-5004	ASM_5008_TestCase
ASM-TA-5005	ASM_5009_TestCase
ASM-TA-5006	ASM_5010_TestCase
ASM-TA-5007	ASM_5032_TestCase
ASM-TA-5008	ASM_5033_TestCase
ASM-TA-5009	ASM_5034_TestCase
ASM-TA-5010	ASM_5011_TestCase
ASM-TA-5011	ASM_5012_TestCase
ASM-TA-5012	ASM_5013_TestCase ASM_5014_TestCase
ASM-TA-5013	ASM_5015_TestCase
ASM-TA-5014	Untestable -- wiredByImpl
ASM-TA-5015	Untestable -- wiredByImpl
ASM-TA-5016	ASM_5016_TestCase
ASM-TA-5017	ASM_5017_TestCase
ASM-TA-5018	ASM_5018_TestCase
ASM-TA-5019	Untestable -- wiredByImpl
ASM-TA-5020	Untestable -- wiredByImpl
ASM-TA-5021	ASM_5019_TestCase
ASM-TA-5022	ASM_5020_TestCase
ASM-TA-5023	ASM_5021_TestCase
ASM-TA-5024	ASM_5018_TestCase
ASM-TA-5025	ASM_5002_TestCase
ASM-TA-5026	ASM_5003_TestCase
ASM-TA-5027	ASM_5004_TestCase
ASM-TA-5028	ASM_5022_TestCase
ASM-TA-5029	ASM_5023_TestCase
ASM-TA-5030	ASM_5024_TestCase
ASM-TA-5031	ASM_5002_TestCase
ASM-TA-5032	ASM_5025_TestCase

ASM-TA-5033	ASM_5026_TestCase
ASM-TA-5034	ASM_5027_TestCase
ASM-TA-5035	ASM_5028_TestCase
ASM-TA-5036	ASM_5029_TestCase
ASM-TA-5037	ASM_5030_TestCase
ASM-TA-5038	ASM_5031_TestCase
ASM-TA-5039	ASM_5035_TestCase
ASM-TA-5040	ASM_5035_TestCase
ASM-TA-5041	ASM_5036_TestCase
ASM-TA-5042	ASM_5037_TestCase
ASM-TA-5043	Requires multiple bindings -- leave to Bindings-TC to devise a test

621

Test Assertion	Test Cases
ASM-TA-6001	Untestable - interfaces compatible
ASM-TA-6002	Untestable - interfaces - compatible superset
ASM-TA-6003	Untestable - interfaces - compatible subset
ASM-TA-6004	ASM_6001_TestCase
ASM-TA-6005	Untestable - tests that components run local to each other
ASM-TA-6006	ASM_6002_TestCase
ASM-TA-6007	ASM_6003_TestCase
ASM-TA-6008	ASM_6004_TestCase
ASM-TA-6009	ASM_6005_TestCase
ASM-TA-6010	ASM_6006_TestCase
ASM-TA-6011	ASM_6007_TestCase
ASM-TA-6012	ASM_6008_TestCase
ASM-TA-6013	ASM_6031_TestCase
ASM-TA-6014	ASM_6032_TestCase
ASM-TA-6015	ASM_6009_TestCase
ASM-TA-6016	ASM_6033_TestCase
ASM-TA-6017	ASM_6010_TestCase
ASM-TA-6018	ASM_6011_TestCase
ASM-TA-6019	ASM_6012_TestCase
ASM-TA-6020	ASM_6013_TestCase

ASM-TA-6021	ASM_6014_TestCase
ASM-TA-6022	Untestable due to lack of mandatory support for Intents
ASM-TA-6023	ASM_6016_TestCase
ASM-TA-6024	ASM_6017_TestCase
ASM-TA-6025	ASM_6018_TestCase
ASM-TA-6026	ASM_6019_TestCase
ASM-TA-6027	ASM_6020_TestCase
ASM-TA-6028	ASM_6021_TestCase
ASM-TA-6029	ASM_6022_TestCase
ASM-TA-6030	ASM_6023_TestCase
ASM-TA-6031	ASM_6024_TestCase
ASM-TA-6032	ASM_6025_TestCase
ASM-TA-6033	Untestable - no wiredByImpl implementation
ASM-TA-6034	Untestable - no wiredByImpl implementation
ASM-TA-6035	ASM_6026_TestCase
ASM-TA-6036	ASM_6026_TestCase
ASM-TA-6037	ASM_6027_TestCase
ASM-TA-6038	ASM_6028_TestCase
ASM-TA-6039	ASM_6028_TestCase
ASM-TA-6040	ASM_6029_TestCase
ASM-TA-6041	ASM_6030_TestCase
ASM-TA-6042	ASM_6034_TestCase
ASM-TA-6043	Tests the introspection of componentType for a composite. No specific testcase for this, since almost all the tests in the test suite depend on this introspection and as a result it is heavily tested.
ASM-TA-6044	ASM_6035_TestCase
ASM-TA-6045	ASM_6036_TestCase
ASM-TA-6046	ASM_6037_TestCase
ASM-TA-6047	ASM_6038_TestCase
ASM-TA-6048	ASM_6039_TestCase
ASM-TA-6049	ASM_6040_TestCase
Test Assertion	Test Cases
ASM-TA-6001	Untestable - interfaces compatible
ASM-TA-6002	Untestable - interfaces - compatible superset

ASM-TA-6003	Untestable -- interfaces -- compatible subset
ASM-TA-6004	ASM_6001_TestCase
ASM-TA-6005	Untestable -- tests that components run local to each other
ASM-TA-6006	ASM_6002_TestCase
ASM-TA-6007	ASM_6003_TestCase
ASM-TA-6008	ASM_6004_TestCase
ASM-TA-6009	ASM_6005_TestCase
ASM-TA-6010	ASM_6006_TestCase
ASM-TA-6011	ASM_6007_TestCase
ASM-TA-6012	ASM_6008_TestCase
ASM-TA-6013	ASM_6031_TestCase
ASM-TA-6014	ASM_6032_TestCase
ASM-TA-6015	ASM_6009_TestCase
ASM-TA-6016	ASM_6033_TestCase
ASM-TA-6017	ASM_6010_TestCase
ASM-TA-6018	ASM_6011_TestCase
ASM-TA-6019	ASM_6012_TestCase
ASM-TA-6020	ASM_6013_TestCase
ASM-TA-6021	ASM_6014_TestCase
ASM-TA-6022	ASM_6015_TestCase
ASM-TA-6023	ASM_6016_TestCase
ASM-TA-6024	ASM_6017_TestCase
ASM-TA-6025	ASM_6018_TestCase
ASM-TA-6026	ASM_6019_TestCase
ASM-TA-6027	ASM_6020_TestCase
ASM-TA-6028	ASM_6021_TestCase
ASM-TA-6029	ASM_6022_TestCase
ASM-TA-6030	ASM_6023_TestCase
ASM-TA-6031	ASM_6024_TestCase
ASM-TA-6032	ASM_6025_TestCase
ASM-TA-6033	Untestable -- no wiredByImpl implementation
ASM-TA-6034	Untestable -- no wiredByImpl implementation
ASM-TA-6035	ASM_6026_TestCase
ASM-TA-6036	ASM_6026_TestCase
ASM-TA-6037	ASM_6027_TestCase

ASM-TA-6038	ASM_6028_TestCase
ASM-TA-6039	ASM_6028_TestCase
ASM-TA-6040	ASM_6029_TestCase
ASM-TA-6041	ASM_6030_TestCase

622

623

Test Assertion	Test Cases
ASM-TA-7001	ASM_7001_TestCase
ASM-TA-7002	ASM_7002_TestCase
ASM-TA-7003	ASM_7003_TestCase
ASM-TA-7004	Untestable
ASM-TA-7005	ASM_7010_TestCase
ASM-TA-7006	ASM_7001_TestCase
ASM-TA-7007	ASM_7002_TestCase
ASM-TA-7008	ASM_7003_TestCase
ASM-TA-7009	ASM_7004_TestCase
ASM-TA-7010	ASM_7005_TestCase
ASM-TA-7011	ASM_7006_TestCase
ASM-TA-7012	ASM_7007_TestCase
ASM-TA-7013	ASM_7008_TestCase
ASM-TA-7014	ASM_7009_TestCase

624

Test Assertion	Test Cases
ASM-TA-8001	ASM_8001_TestCase
ASM-TA-8002	ASM_8002_TestCase
ASM-TA-8003	Untestable
ASM-TA-8004	ASM_8003_TestCase
ASM-TA-8006	ASM_8005_TestCase
ASM-TA-8007	ASM_8003_TestCase
ASM-TA-8008	ASM_8006_TestCase
ASM-TA-8009	ASM_8007_TestCase
ASM-TA-8010	ASM_8008_TestCase
ASM-TA-8011	ASM_8009_TestCase

	ASM_8018_TestCase
ASM-TA-8012	ASM_8010_TestCase
ASM-TA-8013	ASM_8011_TestCase
ASM-TA-8014	ASM_8012_TestCase ASM_8017_TestCase
ASM-TA-8015	ASM_8013_TestCase
ASM-TA-8018	ASM_8014_TestCase
ASM-TA-8020	ASM_8015_TestCase
ASM-TA-8021	ASM_8016_TestCase
ASM-TA-8022	ASM_8019_TestCase

625

Test Assertion	Test Cases
ASM-TA-8001	ASM_8001_TestCase
ASM-TA-8002	ASM_8002_TestCase
ASM-TA-8003	Untestable
ASM-TA-8004	ASM_8003_TestCase
ASM-TA-8005	ASM_8004_TestCase
ASM-TA-8006	ASM_8005_TestCase
ASM-TA-8007	ASM_8003_TestCase
ASM-TA-8008	ASM_8006_TestCase
ASM-TA-8009	ASM_8007_TestCase
ASM-TA-8010	ASM_8008_TestCase
ASM-TA-8011	ASM_8009_TestCase ASM_8018_TestCase
ASM-TA-8012	ASM_8010_TestCase
ASM-TA-8013	ASM_8011_TestCase
ASM-TA-8014	ASM_8012_TestCase ASM_8017_TestCase
ASM-TA-8015	ASM_8013_TestCase
ASM-TA-8018	ASM_8014_TestCase
ASM-TA-8020	ASM_8015_TestCase
ASM-TA-8021	ASM_8016_TestCase
ASM-TA-8022	ASM_8019_TestCase

626

Test Assertion	Test Cases
ASM-TA-9002	ASM_9001_TestCase

ASM-TA-9003	ASM_9002_TestCase
ASM-TA-9004	ASM_9003_TestCase
ASM-TA-9005	ASM_9004_TestCase
ASM-TA-9006	ASM_9005_TestCase

627

Test Assertion	Test Cases
ASM-TA-9002	ASM_9001_TestCase
ASM-TA-9003	ASM_9002_TestCase
ASM-TA-9004	ASM_9003_TestCase
ASM-TA-9005	ASM_9004_TestCase

628

629

Test Assertion	Test Cases
ASM-TA-10001	ASM_10001_TestCase
ASM-TA-10002	ASM_10002_TestCase
ASM-TA-10003	ASM_10003_TestCase

630

631

Test Assertion	Test Cases
ASM-TA-12001	Untestable
ASM-TA-12002	Optional - not worth testing
ASM-TA-12003	Removed by resolution of Issue 205
ASM-TA-12005	ASM_12002_TestCase
ASM-TA-12006	ASM_12003_TestCase
ASM-TA-12007	ASM_12004_TestCase
ASM-TA-12008	Optional - not worth testing
ASM-TA-12009	Optional - not worth testing
ASM-TA-12010	ASM_12005_TestCase
ASM-TA-12011	ASM_12002_TestCase
ASM-TA-12012	ASM_12003_TestCase
ASM-TA-12013	ASM_12006_TestCase
ASM-TA-12014	Optional - not worth testing
ASM-TA-12015	Optional - not worth testing
ASM-TA-12016	Optional - not worth testing
ASM-TA-12017	Optional - not worth testing

ASM-TA-12018	Optional - not worth testing
ASM-TA-12019	Requires an SCA runtime configuration update API
ASM-TA-12020	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12021	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12022	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12023	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12024	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12025	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12026	Optional - not worth testing Requires an SCA runtime configuration update API
ASM-TA-12027	ASM_12007_TestCase
ASM-TA-12028	ASM_12008_TestCase
ASM-TA-12029	ASM_12009_TestCase
ASM-TA-12030	ASM_12010_TestCase
ASM-TA-12031	ASM_12011_TestCase
ASM-TA-12032	ASM_12012_TestCase
ASM-TA-12033	Optional - not worth testing
ASM-TA-12034	Optional - not worth testing
ASM-TA-12035	ASM_12013_TestCase

633

Test Assertion	Test Cases
ASM-TA-13001	ASM_13001_TestCase ASM_13002_TestCase ASM_13003_TestCase ASM_13004_TestCase ASM_13005_TestCase ASM_13006_TestCase ASM_13007_TestCase ASM_13008_TestCase
ASM-TA-13002	ASM_12012_TestCase
ASM-TA-13003	ASM_10003_TestCase

634

635

Test Assertion	Test Cases
ASM-TA-14001	Optional - not worth testing
ASM-TA-14002	Optional - not worth testing
ASM-TA-14003	Untestable
ASM-TA-14004	Optional - not worth testing

4 Catalog of Test Artifacts

4.1 Composite Files - lower level

<u>Name</u>	<u>Valid</u>	<u>Description</u>
<u>Marked "Impl Specific" if it is implementation type specific</u>		<u>Services and References use interface Service1 unless described otherwise</u>
<u>TestComposite1.composite</u> <u>"Impl Specific"</u>	<u>Y</u>	<u>1 service</u> <u>service1Impl</u>
<u>TestComposite2.composite</u> <u>-</u>	<u>N</u>	<u>2 services - with same name</u> <u>0 references</u> <u>service1Impl</u> <u>Contribution: ASM_5005</u>
<u>TestComposite3.composite</u>	<u>N</u>	<u>1 service.</u> <u>2 references - with same name</u> <u>service1Impl2</u> <u>Contribution: ASM_5007</u>
<u>TestComposite4.composite</u> <u>"Impl Specific"</u>	<u>Y</u>	<u>1 service.</u> <u>1 reference (1..1)</u> <u>service1Impl2</u>
<u>TestComposite5.composite</u> <u>"Impl Specific"</u>	<u>Y</u>	<u>1 service - interface = Service2</u> <u>0 references</u> <u>service2Impl</u>
<u>TestComposite6.composite</u> <u>"Impl Specific"</u>	<u>Y</u>	<u>1 service.</u> <u>1 reference (1..n)</u> <u>service1Impl3</u>
<u>TestComposite7.composite</u> <u>"Impl Specific"</u>	<u>Y</u>	<u>1 service.</u> <u>1 reference (0..n)</u> <u>service1Impl4</u>
<u>TestComposite8.composite</u> <u>"Impl Specific"</u>	<u>Y</u>	<u>1 service.</u> <u>1 reference (0..1)</u> <u>service1Impl6</u>

TestComposite9.composite <i>"Impl Specific"</i>	Y	1 service - superset interface = Service1Superset 0 references service1SupersetImpl
TestComposite10.composite	Y	1 service 1 reference (1..1) with binding and target
TestComposite11.composite	N	1 service, 1 reference 1 component with 1..1 reference that is wired and promoted Contribution: ASM_5023
TestComposite12.composite	Y	1 service 0 references 1 property of complex type typed with @type uses TestComposite71
TestComposite13.composite	Y	1 service 0 references 1 property of complex type typed with @element uses TestComposite71
TestComposite14.composite	Y	1 service 0 references 1 property of simple type (string)
TestComposite15.composite	Y	1 service - with multiple bindings declared 0 references
TestComposite16.composite	N	3 services - 2 with same name Contributions: ASM_4002, ASM_6002
TestComposite17.composite	N	2 services - 1 with invalid @promote value Contribution: ASM_6003
TestComposite18.composite	Y	1 service - service interface is compatible subset of interface of promoted component service
TestComposite19.composite	N	1 service, 2 references - with same name Contributions: ASM_4003, ASM_6005
TestComposite20.composite	N	1 service,

		<u>1 reference with invalid @promote value</u> <u>Contribution: ASM_6006</u>
<u>TestComposite21.composite</u>	Y	<u>1 service,</u> <u>1 reference (1..1) with interface that is compatible superset of interface of promoted component reference</u>
<u>TestComposite22.composite</u>	Y	<u>1 service,</u> <u>1 reference (1..1) promotes 2 component references with different interfaces declared (Service1, Service1Superset)</u> <u>uses TestComposite70</u>
<u>TestComposite23.composite</u>	Y	<u>1 service,</u> <u>1 reference (1..1) that promotes component reference with (0..1) multiplicity</u> <u>uses TestComposite8</u>
<u>TestComposite24.composite</u>	Y	<u>1 service,</u> <u>1 reference (0..1) that promotes component reference with (0..n) multiplicity</u>
<u>TestComposite25.composite</u>	Y	<u>1 service,</u> <u>1 reference (1..1) that promotes component reference with (1..n) multiplicity</u>
<u>TestComposite26.composite</u>	N	<u>1 service,</u> <u>1 reference,</u> <u>3 properties - all with same name</u> <u>Contributions: ASM_4004, ASM_6012</u>
<u>TestComposite27.composite</u>	Y	<u>1 service,</u> <u>2 components, one with @autowire="true" reference , one with service with compatible interface</u>
<u>TestComposite28.composite</u>	Y	<u>1 service,</u> <u>2 components, one with @autowire="true" reference , one with service with superset interface</u>
<u>TestComposite30.composite</u>	N	<u>1 service,</u> <u>2 components, one with @autowire="true" reference + set of intents, one with service with compatible interface but different set of intents</u> <u>Contribution: ASM_6016</u>
<u>TestComposite31.composite</u>	N	<u>1 service, composite service has a @promote URI that does</u>

		<p><u>not point to a component service within the composite</u></p> <p><u>0 references</u></p> <p><u>Contribution: ASM_6021</u></p>
<u>TestComposite32.composite</u>	N	<p><u>1 service, contains a component with a 1..n reference that is neither promoted nor wired internally in the composite</u></p> <p><u>Contribution: ASM_6022</u></p>
<u>TestComposite33.composite</u>	Y	<p><u>1 service, 2 properties, plus a component with multiple properties which are sourced from the composite properties</u></p>
<u>TestComposite34.composite</u>	Y	<p><u>Composite designed for including within another composite - contains 1 component only</u></p>
<u>TestComposite35.composite</u>	Y	<p><u>1 service and 1 reference, where the component that implements the service & reference is included from a second composite (TestComposite36)</u></p>
<u>TestComposite36.composite</u>	Y	<p><u>Composite designed for inclusion within another composite - contains 1 component that implements 1 service and 1 reference</u></p>
<u>TestComposite37.composite</u>	Y	<p><u>Composite designed for inclusion within another composite - contains 2 components, #1 has 1 service and 1 reference, #2 has 1 service</u></p>
<u>TestComposite38.composite</u>	Y	<p><u>Has @local=true, containing 2 components, #1 has 1 service and 1 reference, #2 has 1 service - reference of #1 is wired to service of #2</u></p>
<u>TestComposite39.composite</u>	N	<p><u>1 service, 1 reference where the reference has an intent that is mutually exclusive with an intent on the component reference which it promotes</u></p> <p><u>Contribution: ASM_6031</u></p>
<u>TestComposite40.composite</u>	N	<p><u>1 service, 1 reference where the reference has @requires containing 2 mutually exclusive intents</u></p> <p><u>Contribution: ASM_6032</u></p>
<u>TestComposite41.composite</u>	N	<p><u>1 service, 1 reference, where the reference multiplicity is 0..1 but the component reference which it promotes has multiplicity 1..1</u></p> <p><u>Contribution: ASM_6033</u></p>
<u>TestComposite42.composite</u>	Y	<p><u>1 service, composite has @constrainingType set to a constrainingType to which the composite does conform</u></p>
<u>TestComposite43.composite</u>	Y	<p><u>1 service,</u></p> <p><u>1 reference,</u></p> <p><u>2 properties. 1 property declares its type using the @type attribute (complex type sca:Callback)</u></p>

TestComposite44.composite	Y	1 service. 1 reference. 2 properties. 1 property declares its type using the @element attribute (complex type sca:callback)
TestComposite45.composite	N	1 service. 1 reference. 2 properties. 1 property declares its type using both an @element attribute and a @type attribute (complex type sca:Callback) Contribution: ASM_4007
TestComposite46.composite	Y	1 service. 1 reference. 2 properties. 1 property declares @mustSupply=true
TestComposite48.composite	Y	2 services
TestComposite49.composite	Y	1 services 2 references, 1 of (0..1), 1 of (0..n)
TestComposite50.composite	Y	1 service, 2 properties with @mustSupply="false"
TestComposite51.composite	N	1 service - with invalid remotable interface - Service3 - that has 2 operations with the same name Contribution: ASM_8001
TestComposite52.composite "Impl Specific"	Y	1 service with Service4 interface service4Impl
TestComposite53.composite "Impl Specific"	Y	1 service 1 reference with Service4 interface contains a component which checks whether an input parameter is modified during invocation of an operation on the Service4 interface service1Impl7
TestComposite54.composite "Impl Specific"	Y	1 service 1 reference with a callback - Service5 & Service5Callback service1Callback5Impl
TestComposite55.composite "Impl Specific"	Y	1 service with callback - Service5 & Service5Callback service5Impl

<u>TestComposite56.composite</u>	Y	(ASM_8004) <u>1 service</u> <u>1 reference with callback - Service6 & Service6Callback</u>
<u>TestComposite57.composite</u>	Y	(ASM_8004) <u>1 service with callback - Service6 & Service6Callback</u>
<u>TestComposite58.composite</u>	Y	(ASM_8005) <u>1 service</u> <u>1 reference with callback - Service7 and Service7Callback</u>
<u>TestComposite59.composite</u>	Y	(ASM_8005) <u>1 service with callback - Service7 and Service7Callback</u>
<u>TestComposite60.composite</u> <u>"Impl Specific"</u>	Y	(ASM_8006) <u>1 service</u> <u>1 reference with callback - Service9 & Service9Callback</u> <u>service1Callback9Impl</u>
<u>TestComposite61.composite</u> <u>"Impl Specific"</u>	Y	(ASM_8006) <u>1 service with callback - Service9 & Service9Callback</u> <u>service9Impl</u>
<u>TestComposite62.composite</u>	N	(ASM_8003) <u>1 service, 1 reference with callback</u> <u>Callback interface of composite reference is not compatible</u> <u>with callback interface of promoted component reference</u> <u>Contribution: ASM_8011</u>
<u>TestComposite63.composite</u>	N	(ASM_8003) <u>1 service with callback - where the callback interface is not</u> <u>compatible with the callback interface on the promoted</u> <u>component service - Service5 & Service8Callback</u> <u>Contribution: ASM_8010</u>
<u>TestComposite64.composite</u>	N	(ASM_8012) <u>1 service</u> <u>1 reference (0..1) with callback - Service1 & Service5Callback -</u> <u>where the promoted component reference has no callback</u> <u>declared</u> <u>Contribution: ASM_8012</u>
<u>TestComposite65.composite</u> <u>"Impl Specific"</u>	Y	(ASM_8003) <u>1 service with a callback - Service5 & Service5Callback - where</u>

		<u>the implementing component invokes the callback operation multiple times for a single invocation of the forward interface service5Impl2</u>
<u>TestComposite66.composite</u> <u>"Impl Specific"</u>	Y	<u>1 service, remotable, service has Confidentiality applied</u> <u>0 references</u> <u>service1Impl</u>
<u>TestComposite67.composite</u>	Y	<u>* copy of TestComposite5 but in namespace http://oasis/tests2</u>
<u>TestComposite68.</u>	Y	<u>Wrapper of TestComposite5 but in namespace http://oasis/tests2</u>
<u>TestComposite69.composite</u>	Y	<u>1 service</u> <u>* wrapper of TestComposite1 in namespace http://oasis/tests2</u>
<u>TestComposite70.composite</u> <u>"Impl Specific"</u>	Y	<u>1 service.</u> <u>1 reference (1..1) - interface Service1Superset</u> <u>service1Impl8</u>
<u>TestComposite71.composite</u> <u>"Impl Specific"</u>	Y	<u>1 service</u> <u>0 references</u> <u>2 properties of type string</u> <u>service1Impl5</u>
<u>TestComposite72.composite</u>	N	<u>1 service</u> <u>1 service with callback - Service1 & Service5Callback - where the promoted component service has no callback declared</u> <u>0 references</u> <u>Contribution: ASM_8017</u>
<u>TestComposite73.composite</u> <u>"Impl Specific"</u>	Y	<u>1 service</u> <u>0 references</u> <u>1 property of type float</u>
<u>TestComposite74.composite</u>	N	<u>1 service</u> <u>1 reference "Reference1"</u> <u>- reference promotes a component reference with multiplicity 1..1 and @nonOverridable="true"</u> <u>Contribution ASM_5040</u>
<u>TestComposite75.composite</u>	Y	<u>3 services all with interface = Service1</u> <u>0 references</u> <u>0 properties</u>

		<u>Contribution: General</u>
Name	Valid	Description
Marked " <i>Impl-Specific</i> " if it is implementation type specific		<i>Services and References use interface Service1 unless described otherwise</i>
TestComposite1.composite <i>"Impl-Specific"</i>	Y	1 service
TestComposite2.composite <i>"Impl-Specific"</i>	N	2 services – with same name 0 references
TestComposite3.composite <i>"Impl-Specific"</i>	N	1 service, 2 references – with same name
TestComposite4.composite <i>"Impl-Specific"</i>	Y	1 service, 1 reference (1..1)
TestComposite5.composite <i>"Impl-Specific"</i>	Y	1 service – interface = Service2 0 references
TestComposite6.composite <i>"Impl-Specific"</i>	Y	1 service, 1 reference (1..n)
TestComposite7.composite <i>"Impl-Specific"</i>	Y	1 service, 1 reference (0..n)
TestComposite8.composite <i>"Impl-Specific"</i>	Y	1 service, 1 reference (0..1)
TestComposite9.composite <i>"Impl-Specific"</i>	Y	1 service – superset interface = Service1Superset 0 references
TestComposite10.composite	Y	1 service 1 reference with binding and target
TestComposite11.composite	N	1 service,

		1 reference 1 component with 1..1 reference that is wired and promoted
TestComposite12.composite	Y	1 service 0 references 1 property of complex type typed with @type
TestComposite13.composite	Y	1 service 0 references 1 property of complex type typed with @element
TestComposite14.composite	Y	1 service 0 references 1 property of simple type
TestComposite15.composite	Y	1 service — with multiple bindings declared 0 references
TestComposite16.composite	N	3 services — 2 with same name
TestComposite17.composite	N	2 services — 1 with invalid @promote value
TestComposite18.composite	Y	1 service — service interface is compatible subset of interface of promoted component service
TestComposite19.composite	N	1 service, 2 references — with same name
TestComposite20.composite	N	1 service, 1 reference with invalid @promote value
TestComposite21.composite	Y	1 service, 1 reference (1..1) with interface that is compatible superset of interface of promoted component reference
TestComposite22.composite	Y	1 service, 1 reference (1..1) promotes 2 component references with different interfaces declared (Service1, Service1Superset)
TestComposite23.composite	Y	1 service, 1 reference (1..1) that promotes component reference with (0..1) multiplicity
TestComposite24.composite	Y	1 service, 1 reference (0..1) that promotes component reference with (0..n) multiplicity
TestComposite25.composite	Y	1 service,

		1 reference (1..1) that promotes component reference with (1..n) multiplicity
TestComposite26.composite	N	1 service, 1 reference, 3 properties – all with same name
TestComposite27.composite	Y	1 service, 2 components, one with @autowire="true" reference , one with service with compatible interface
TestComposite28.composite	Y	1 service, 2 components, one with @autowire="true" reference , one with service with superset interface
TestComposite29.composite	Y	1 service, 4 components, 1 with @autowire="true" reference + set of intents, 1 with service with compatible interface + same set of intents, 2 with service with compatible interface and different set of intents
TestComposite30.composite	N	1 service, 2 components, one with @autowire="true" reference + set of intents, one with service with compatible interface but different set of intents
TestComposite31.composite	N	1 service, composite service has a @promote URI that does not point to a component service within the composite 0 references
TestComposite32.composite	N	1 service, contains a component with a 1..n reference that is neither promoted nor wired internally in the composite
TestComposite33.composite	Y	1 service, 2 properties, plus a component with multiple properties which are sourced from the composite properties
TestComposite34.composite	Y	Composite designed for including within another composite – contains 1 component only
TestComposite35.composite	Y	1 service and 1 reference, where the component that implements the service & reference is included from a second composite (TestComposite36)
TestComposite36.composite	Y	Composite designed for inclusion within another composite – contains 1 component that implements 1 service and 1 reference
TestComposite37.composite	Y	Composite designed for inclusion within another composite – contains 2 components, #1 has 1 service and 1 reference, #2 has 1 service
TestComposite38.composite	Y	Has @local=true, containing 2 components, #1 has 1 service and 1 reference, #2 has 1 service – reference of #1 is wired to

		service-of #2
TestComposite39.composite	N	1 service, 1 reference where the reference has an intent that is mutually exclusive with an intent on the component reference which it promotes
TestComposite40.composite	N	1 service, 1 reference where the reference has @requires-containing 2 mutually exclusive intents
TestComposite41.composite	N	1 service, 1 reference, where the reference multiplicity is 0..1 but the component reference which it promotes has multiplicity 1..1
TestComposite42.composite	Y	1 service, composite has @constrainingType set to a constrainingType to which the composite does conform
TestComposite43.composite	Y	1 service, 1 reference, 2 properties. 1 property declares its type using the @type attribute
TestComposite44.composite	Y	1 service, 1 reference, 2 properties. 1 property declares its type using the @element attribute
TestComposite45.composite	N	1 service, 1 reference, 2 properties. 1 property declares its type using both an @element attribute and a @type attribute
TestComposite46.composite	Y	1 service, 1 reference, 2 properties. 1 property declares @mustSupply=true
TestComposite47.composite	N	1 service, 2 references – the references have the same @name attribute value
TestComposite48.composite	Y	2 services
TestComposite49.composite	Y	1 services 2 references, 1 of (0..1), 1 of (0..n)
TestComposite50.composite	Y	1 service, 2 properties with @mustSupply="false"
TestComposite51.composite	N	1 service – with invalid remotable interface – Service3 – that has 2 operations with the same name
TestComposite52.composite	Y	1 service with Service4 interface
"Impl-Specific"		

TestComposite53.composite <i>"Impl-Specific"</i>	Y	1-service 1-reference-with-Service4-interface contains a component which checks whether an input-parameter is modified during invocation of an operation on the Service4-interface
TestComposite54.composite	Y	(ASM_8003) 1-service- 1-reference-with-a-callback-Service5-&Service5Callback-
TestComposite55.composite	Y	(ASM_8003) 1-service-with-callback-Service5-&Service5Callback
TestComposite56.composite	Y	(ASM_8004) 1-service- 1-reference-with-callback-Service6-&Service6Callback
TestComposite57.composite	Y	(ASM_8004) 1-service-with-callback-Service6-&Service6Callback
TestComposite58.composite	Y	(ASM_8005) 1-service- 1-reference-with-callback-Service7-and-Service7Callback
TestComposite59.composite	Y	(ASM_8005) 1-service-with-callback-Service7-and-Service7Callback
TestComposite60.composite	Y	(ASM_8006)- 1-service 1-reference-with-callback-Service9-&Service9Callback
TestComposite61.composite	Y	(ASM_8006) 1-service-with-callback-Service9-&Service9Callback
TestComposite62.composite	N	(ASM_8003) 1-service, 1-reference-with-callback Callback interface of composite-reference is not compatible-with-callback interface of promoted-component-reference
TestComposite63.composite	N	(ASM_8003) 1-service-with-callback-where-the-callback-interface-is-not-compatible-with-the-callback-interface-on-the-promoted-component-service-Service5-&Service8Callback-
TestComposite64.composite	N	(ASM_8012) 1-service-

		1 reference (0..1) with callback – Service1 & Service5Callback – where the promoted component reference has no callback declared
TestComposite65.composite	Y	(ASM_8003) 1 service with a callback – Service5 & Service5Callback – where the implementing component invokes the callback operation multiple times for a single invocation of the forward interface
TestComposite66.composite "Impl-Specific"		1 service, remotable, service has Confidentiality applied 0 references
TestComposite67.composite	Y	* copy of TestComposite5 but in namespace http://oasis/tests2
TestComposite68.	Y	Wrapper of TestComposite5 but in namespace http://oasis/tests2
TestComposite69.composite	Y	1 service * wrapper of TestComposite1 in namespace http://oasis/tests2
TestComposite70.composite "Impl-Specific"	Y	1 service, 1 reference (1..1) – interface Service1Superset
TestComposite71.composite "Impl-Specific"	Y	1 service 0 references 2 properties of type string
TestComposite72.composite	N	(ASM_8017) 1 service 1 service with callback – Service1 & Service5Callback – where the promoted component service has no callback declared 0 references

4.2 Constraint Files

Name	Description
Test_ASM_7001.constraint	2 services 0 references 1 property
Test_ASM_7002.constraint	1 service 2 references 2 properties

Test_ASM_7003.constraint	1-service 0-references 2-properties
Test_ASM_7004.constraint	1-service 1-property
TestComposite42.constraint	1-service 1-property
Test_ASM_7010_A.constraint	1-service 1-reference 1-property Same @name-value as Test_ASM_7010_B.constraint
Test_ASM_7010_B.constraint	1-service 0-references 1-property Same @name-value as Test_ASM_7010_A.constraint

4.3 Java Interfaces

Name	Description
test.Service1.java	1 operation with 1 input, 1 output parameter
test.Service2.java	Remotable service interface that is not compatible with Service1 interface
test.Service1Superset.java	Remotable service interface which is a superset of Service1 interface 2 operations with 1 input, 1 output parameter
test.Service3.java	(ASM_8001) Remotable interface with 2 operations with the same name - it is also a superset of Service1
test.Service4.java	Remotable service interface 1 operation with 1 input, 1 output parameter - input parameter is a Java mutable type = List of Strings
test.Service5.java	Remotable service interface with a Callback

	interface (Service5Callback) 1 operation with 1 input, 1 output parameter
test.Service5Callback.java	Remotable callback interface for Service5, 1 operation with 1 input, 1 output parameter
test.Service1_Intent	Service1 interface + Confidentiality intent 1 operation with 1 input, 1 output parameter
test.Service6.java	Remotable service interface with a local callback interface (Service6Callback) 1 operation with 1 input, 1 output parameter
test.Service7.java	Local service interface with a remotable callback interface (Service7Callback) 1 operation with 1 input, 1 output parameter
test.Service6Callback.java	Callback interface for Service6 - local 1 operation with 1 input, 1 output parameter
test.Service7Callback.java	Callback interface for Service7 - remotable 1 operation with 1 input, 1 output parameter
test.Service8Callback	Remotable callback interface incompatible with Service5Callback interface

4.4 Java Implementation Classes

<u>Name</u>	<u>Description</u>
	<u>Services and references use interface Service1 unless otherwise described</u>
<u>test.Service1Impl.java</u>	<u>1 service</u> <u>0 references</u>
<u>test..Service1Impl2.java</u>	<u>1 service</u> <u>1 reference (1..1)</u>
<u>test.Service1Impl3.java</u>	<u>1 service</u> <u>1 reference (1..n)</u>

test.Service1Impl4.java	1 service 1 reference (0..n)
test.Service1Impl5.java	1 service 0 references 3 properties
test.Service1Impl6.java	1 service 1 reference (0..1)
test.Service1impl7.java	1 service 1 reference (1..1) with interface Service4 - operation1 method calls reference operation and checks for modification of the input parameter
test.Service1Impl8.java	1 service 1 reference (1..1) - interface Service1Superset
test.Service1Impl9	1 service 0 references 1 property of type float
test.Service2Impl.java	1 service with interface Service2
test.Service1SupersetImpl.java	1 service with interface Service1Superset
test.Service3Impl.java	1 service with interface Service3
test.Service4impl.java	1 service with interface Service4 - service operation modifies input parameter
test.Service6Impl.java	1 service with interface Service6
test.Service7Impl.java	1 service with interface Service7
test.Service1Callback6Impl.java	1 service 1 reference with Service6/Service6Callback interfaces
test.Service1Callback7Impl.java	1 service 1 reference with Service7/Service7Callback
TestException.java	Exception returned if test application receives an exception during execution
TestInvocation.java	Client testcase invocation interface
ASM_0001_Client.java	Client invocation component

	<u>1 service</u> <u>0 references</u>
<u>ASM_0002_Client.java</u>	<u>"Standard" client invocation component</u> <u>1 service</u> <u>1 reference (0..1)</u>
<u>ASM_0003_Client.java</u>	<u>Client invocation component</u> <u>1 service</u> <u>1 reference (1..n)</u>
Name	Description
	Services and references use interface Service1 unless otherwise described
test.service1Impl.java	1 service 0 references
test.service1Impl2.java	1 service 1 reference (1..1)
test.service1Impl3.java	1 service 1 reference (1..n)
test.service1Impl4.java	1 service 1 reference (0..n)
test.service1Impl5.java	1 service 0 references 3 properties
test.service1Impl6.java	1 service 1 reference (0..1)
test.service1impl7.java	1 service 1 reference (1..1) with interface Service4 —operation1 method calls reference operation and checks for modification of the input parameter
test.Service1Impl8.java	1 service 1 reference (1..1) — interface Service1Superset

test.service2Impl.java	1 service with interface Service2
test.service1SupersetImpl.java	1 service with interface Service1Superset
test.service3Impl.java	1 service with interface Service3
test.service4impl.java	1 service with interface Service4 —service operation modifies input parameter
test.service6Impl.java	1 service with interface Service6
test.service7Impl.java	1 service with interface Service7
test.service1Callback6Impl.java	1 service 1 reference with Service6/Service6Callback interfaces
test.service1Callback7Impl.java	1 service 1 reference with Service7/Service7Callback
TestException.java	Exception returned if test application receives an exception during execution
TestInvocation.java	Client testcase invocation interface
ASM_0001_Client.java	Client invocation component 1 service 0 references
ASM_0002_Client.java	"Standard" client invocation component 1 service 1 reference (0..1)
ASM_0003_Client.java	Client invocation component 1 service 1 reference (1..n)

4.5 WSDL Interface Files

Name	Description
Service1.wsdl	portType = Service1

	1 operation with 1 input, 1 output parameter
Service1Superset.wsdl	portType = Service1Superset - superset of Service1 2 operations - 1 input, 1 output parameter
Service1_Intent.wsdl	portType = Service1_Intent equivalent to Service1, but with @requires="sca:confidentiality" applied to the portType element 1 operation with 1 input, 1 output parameter
Service1a.wsdl	Version of Service1.wsdl containing no XML type declarations, but instead containing a @schemaLocation attribute, but one which points to an valid XSD file
Service1b.wsdl	Version of Service1.wsdl containing no XML type declarations, but instead containing a @schemaLocation attribute, but one which points to an invalid file
Service2.wsdl	1 service - incompatible with Service1 1 operation with 1 input, 1 output parameter
Service3.wsdl	portType = Service3 2 operations - overloaded with the same name operation1 has string in string out operation1a has int in string out
Service4.wsdl	portType = Service4 1 operation with 1 input, 1 output parameter - input parameter is list of Strings
Service5.wsdl	Remotable service interface with a single operation with 1 input and 1 output parameter and a Callback interface
Service9.wsdl	Remotable service interface, with @callback attribute applied to the portType element 1 operation with 1 input and 1 output parameter
Service5Callback.wsdl	Remotable callback interface for Service5, with a single operation with 1 input and 1 output parameter
Service8Callback.wsdl	Callback interface which is not compatible with other Callback interfaces (uses a different operation name) 1 operation with 1 input and 1 output parameter

Service9Callback.wsdl	Callback interface for Service9 1 operation with 1 input and 1 output parameter

648

5 Conformance

649

650 There are no conformance statements relating to the TestCases.

651

Appendix A. Acknowledgments

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

Participants:

Participant Name	Affiliation
Bryan Aupperle	IBM
Vladislav Bezrukov	SAP AG*
David Booz	IBM
Martin Chapman	Oracle Corporation
Vamsavardhana Reddy Chillakuru	IBM
Mark Combellack	Avaya, Inc.
Mike Edwards	IBM
Anish Karmarkar	Oracle Corporation
Ashok Malhotra	Oracle Corporation
Plamen Pavlov	SAP AG*
Eric Wells	Hitachi, Ltd.

660

Appendix B. ~~Non-Normative Text~~

661

Appendix C. Revision History

Revision	Date	Editor	Changes Made
17	2009-04-15	Mike Edwards	Section 4 testcases added
18	2009-04-16	Mike Edwards	Section 7 testcases added Tables of Artifacts added
19	04/05/09	Mike Edwards	Section 8, 9, 10 testcases added
20	06/05/09	Mike Edwards	Section 12 testcases added
21	06/05/09	Mike Edwards	Section 13, 14 testcases added. All sections now covered.
22	19/05/09	Mike Edwards	Most testcases converted to WSDL interfaces
23	02/06/09	Mike Edwards	Added Acknowledgements Added missing descriptions of artifacts Added extra testcases to cover previously untreated Test Assertions
24	10/06/09	Mike Edwards	Added ASM_7010_TestCase Added ASM_8017_TestCase Added ASM_8018_TestCase Added ASM_8019_TestCase Added Namespace and Package Name declarations
25	16/06/09	Mike Edwards	CD01 / Public Review draft 01 All changes accepted, file name changed to match OASIS guidelines.
cd01_rev1	08/03/10	Mike Edwards	Issue 205 - deleted ASM_12001_TestCase Issue 165 - descriptions of ASM_8009_TestCase and ASM_8018_TestCase updated Issue 150 - marked ASM_8004_TestCase as language specific and removed from Assembly test suite (given to Java TC for their test suite)
cd01_rev2	10/03/10	Mike Edwards	Issue 172 - Added new testcase ASM_5038 and associated artifacts
cd01_rev3	12/04/10	Mike Edwards	Issue 159 - Testcases deleted, added and reassigned to align with latest CD of the Assembly specification Issue 154 Issue 155 Issue 161

			Issue 162 Issue 163 Issue 164 Issue 165 Issue 166 Issue 167 Issue 168 Issue 169 Issue 170 Issue 171 Issue 172 - Added new testcase ASM_5038_TestCase Issue 173 Issue 174 Issue 184 Issue 185 Issue 186 Issue 190 Issue 191 Issue 193 Issue 196 Issue 197 Issue 198 Issue 199 Issue 202 Issue 205 Issue 207 Issue 208 Issue 209 Issue 216 Issue 219 Issue 226
cd01-Rev4	19/04/10	Mike Edwards	Issue 220 - Remove testcase ASM_6015 and associated artifacts (TestComposite29)

			Issue 224
cd01-Rev5	18/05/10	Mike Edwards	Issue 206 Issue 230
cd01-rev6	08/06/10	Bryan Aupperle	Editorial fixes to front matter including links and references
cd02	08/06/10	Mike Edwards	Create CD02 All changes accepted

664