



TestCases for the SCA POJO Component Implementation Specification Version 1.1_J ~~Common Annotations and APIs Version 1.1~~ Specification

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

15 August 2011~~19 July 2010~~

Specification URIs:

This vVersion:

<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-csprd02.pdf>
(Authoritative)<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-csprd02.htmleaa-1.1-testcases-ed01.html>
<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-csprd02.odteaa-1.1-testcases-ed01.pdf>(Authoritative)

Previous vVersion:

<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-1.0-csprd01.pdf>
(Authoritative)
<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-1.0-csprd01.html>
<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-1.0-csprd01.odt>

Latest version:

<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases.pdf> (Authoritative)
<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases.html>
<http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases.odt>
N/A

Latest Version:

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

Technical Committee:

[OASIS Service Component Architecture / J \(SCA-J\) TC](#)

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sca-j

Chair(s):

David Booz (booz@us.ibm.com), IBM, IBM
Mark Gombellaack Avaya Inc
Anish Karmarkar (Anish.Karmarkar@oracle.com), Oracle

Editors:

David Booz (booz@us.ibm.com), IBM
Mike Edwards (mike_edwards@uk.ibm.com), IBM
Anish Karmarkar (Anish.Karmarkar@oracle.com), Oracle

Additional artifacts:

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

- [Test suite artifacts ZIP archive: http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-csprd02/testcases.zip](http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-csprd02/testcases.zip)

Related work:

This specification is related to:

- [Service Component Architecture SCA-J Common Annotations and APIs Specification Version 1.1](http://docs.oasis-open.org/opencsa/sca-j/sca-j-javacaa-1.1-spec.html). Latest version. <http://docs.oasis-open.org/opencsa/sca-j/sca-j-javacaa-1.1-spec.html>
- [Service Component Architecture POJO Component Implementation Specification Version 1.1](http://docs.oasis-open.org/opencsa/sca-j/sca-j-javaci-1.1-spec.html). Latest version. <http://docs.oasis-open.org/opencsa/sca-j/sca-j-javaci-1.1-spec.html>

Declared XML namespaces:**Editor(s):**

Mike Edwards, IBM
David Booz, IBM

Related Work:

This document is related to:

- [Service Component Architecture Java Common Annotations and APIs Specification Version 1.1](http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec-ed04.pdf)
<http://docs.oasis-open.org/opencsa/sca-j/sca-javacaa-1.1-spec-ed04.pdf>

Declared XML Namespace(s):

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

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

Abstract:

This document defines the TestCases for the SCA [POJO Component Implementation Specification Version 1.1](#), [Java Common Annotations and APIs 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 Java Common Annotations and APIs 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 POJO Component Implementation Specification Version 1.1.~~

Status:

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

~~Technical Committee members should send comments on this Work Product 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-j/>.~~

~~For information on whether any patents have been disclosed that may be essential to implementing this Work Product, 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-j/ipr.php>).

Citation format:

When referencing this Work Product the following citation format should be used:

[SCA-POJO-CI-V1.1-Testcases]

TestCases for the SCA POJO Component Implementation Specification Version 1.1. 15 August 2011. OASIS Committee Specification Draft 02 / Public Review Draft 02. <http://docs.oasis-open.org/opencsa/sca-j/sca-j-pojo-ci-1.1-testcases-csprd02.html>.

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-j/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/sca-j/>

Notices

Copyright © OASIS ~~Open 2011@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 name "[OASIS](#)" is a trademarks "[OASIS](#)", "[Service Component Architecture](#)" are trademarks of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction.....	5
1.1	TestCase Structure.....	5
1.2	Namespaces and Java Package Names.....	6
1.3	Terminology.....	7
1.4	Normative References.....	7
1.5	Non-normative References.....	7
2	TestCases.....	8
2.1	Section 2.....	8
2.2	Section 5.....	9
2.3	Section 8.....	13
2.4	Section 8.....	14
2.5	Section 9.....	34
2.6	Section 10.....	36
3	Catalog of Test Artifacts.....	42
3.1	Java Interfaces.....	42
3.2	Java Implementation Classes.....	42
3.3	WSDL Interface Files.....	47
4	Conformance.....	48
Appendix A	Test Assertions for the SCA POJO Component Implementation Specification.....	49
Appendix A.1	Example Test Assertion.....	49
Appendix A.2	Test Assertions for SCA POJO Component Implementation Specification Section 2.	49
Appendix A.3	Test Assertions for SCA POJO Component Implementation Specification Section 4.	50
Appendix A.4	Test Assertions for SCA POJO Component Implementation Specification Section 5.	51
Appendix A.5	Test Assertions for SCA POJO Component Implementation Specification Section 6.	54
Appendix A.6	Test Assertions for SCA POJO Component Implementation Specification Section 8.	55
Appendix A.7	Test Assertions for SCA POJO Component Implementation Specification Section 9.	68
Appendix A.8	Test Assertions for SCA POJO Component Implementation Specification Section 10	71
Appendix B	Cross Mapping of Normative Statements to Test Assertions.....	75
Appendix C	Cross Mapping of Test Assertions to TestCases.....	77
Appendix D	Acknowledgments.....	80
Appendix E	Revision History.....	81

1 Introduction

This document defines the TestCases for the SCA [SCA POJO Component Implementation Specification Version 1.1 Assembly](#) specification.

The tests described in this document are [derived from the normative statements in SCA POJO Component Implementation Specification Version 1.1 via Test Assertions which are described in Appendix A. Test Assertions for the SCA POJO Component Implementation Specification related to the Test Assertions described in the SCA Assembly Test Assertions document \[CAA-TA\]](#).

1.1 TestCase Structure

The SCA J [POJO-CIGAA](#) 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) [in the case that should](#) the SCA runtime under test not support Web services as a binding technology.

39 Test Application

40 Each Test Application consists of one top level SCA Composite file and one or more other SCA Compos-
41 ite files and their associated artifacts (implementations, interface files), plus test client invocation applica-
42 tion described above.

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

47 The components in the composites are typically implemented by Java POJO classes.

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

51 ~~The application is structured so that alternative technologies can be used. For example, replacing the~~
52 ~~contents of the second level or third level composites allows different Java implementation technologies~~
53 ~~to be tested—eg POJOs or Spring Application Contexts may be used. Similarly, the binding used to con-~~
54 ~~nect from the top level composite to the client application may be changed from Web services to JMS if~~
55 ~~required, simply by changing the binding on the <service/> of the top level composite.~~

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

58 ~~This variable can have one of the following values:~~

- 59 • ~~"POJO"————— for Java implementations~~
- 60 • ~~"Spring"————— for Spring implementations~~

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

62 Test Artifacts Organization

63 Note that the design of these testcases promotes reuse of artifacts between testcases, so that many test-
64 cases share components. For example, components implementing simple invocable services are all im-
65 plemented using a single parameterized implementation artifact.

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

71 Broadly, Contribution names are as follows:

72 • POJOJCA_nnnn - a contribution that is specific for a particular testcase, where "nnnn" is
73 the number of the testcase. Often this is required because a particular testcase involves artifacts
74 that contain errors that are statically checkable - an SCA runtime is permitted to reject such arti-
75 facts when they are contributed and deployed and it is important to ensure that contributions con-
76 taining deliberate errors for one testcase do not interfere with the operation of other testcases.

77 • POJO_General - a shared contribution containing implementation type independent artifacts that
78 can be used by many testcases.

79 • JCA_nnnn_POJO—a contribution for a specific testcase where there is a need for language spe-
80 cific artifacts that relate to that testcase alone

81 • JCA_General—a shared contribution containing implementation type independent artifacts that
82 can be used by many testcases.

83 • JCA_General_POJO—a shared contribution containing implementation type dependent artifacts
84 for Java POJOs. These artifacts can include both Java classes and also SCA composites that
85 directly use Java classes.

86 ~~Note that the names of Contributions containing implementation specific artifacts ends with a name that is~~
87 ~~specific to the implementation type—so "_POJO" is used for Java POJO implementations, "_Spring" is~~
88 ~~used for Spring implementations (and so on). Note that the name following the underscore matches the~~
89 ~~name used in the "OASIS_TESTENV_IMPL_LANG" variable used to control execution of the test client.~~

90 The concept is that where there is an implementation type specific contribution, each implementation type
91 must provide its own versions of the same basic artifacts. Typically, this means that each contribution
92 must contain the same set of Composites, but that the implementation type dependent artifacts that these
93 composites use will differ from implementation type to implementation type.

94 Basically, the setting of the variable is used to select the suffix used for implementation type dependent
95 contributions. If the variable is set to "POJO" then the contribution "JCA_General_POJO" is selected,
96 whereas if the variable is set to "Spring", the contribution "JCA_General_Spring" is selected.

- 97 • Namespaces and Java Package Names

98 The SCA [POJO-CI](#) Assembly testcase suite makes use of some XML namespaces and Java package
99 names, as follows:

100 SCA Artifact Namespaces

101 These apply to artifacts such as Composites

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

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

104 WSDL Namespace

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

106 Java Package name

107 For Java interface classes and for Java implementation classes

108 `org.oasisopen.sca.test`

109 1.2 Terminology

110 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
111 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as de-
112 scribed in [IETF RFC 2119 \[RFC 2119\]](#)

113 1.3 Normative References

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

[JAVACI] _____ OASIS Committee Specification Draft 04, *SCA-J POJO Component
Implementation Specification V1.1, August 2011*
<http://docs.oasis-open.org/opencsa/sca-j/sca-javaci-spec-v1.1-csd04.pdf> [GAA-
TA] _____ OASIS Committee Draft 01, "Test Assertions for the SCA_J Common
Annotations and APIs Version 1.1 Specification", July 2010.
<http://docs.oasis-open.org/opencsa/sca-j/sca-j-cao-1.1-test-assertions-cd01.pdf>

[JAX-B] _____ JAXB 2.1 Specification,
<http://www.jcp.org/en/jsr/detail?id=222>

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

114 1.4 Non-normative References

115 N/A

116

117 2 TestCases

118 2.1 Section 2

119 POJOJGA_2001_TestCase

120

<u>Testcase ID</u>	<u>POJO_2001_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-2001</u>
<u>Description</u>	<u>Tests that a Java POJO implementation can define a service interface using a Java interface</u>
<u>Artifacts</u>	<u>POJO_2001_TestCase.java</u> <u>Test_POJO_2001.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>service1Impl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_2001 request service1 operation1 invoked"</u>

121

122 POJO_2002_TestCase

123

<u>Testcase ID</u>	<u>POJO_2002_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-2001</u>
<u>Description</u>	<u>Tests that a Java POJO implementation can define a service interface using a Java class</u>
<u>Artifacts</u>	<u>POJO_2002_TestCase.java</u> <u>Test_POJO_2002.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>serviceImplClass.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_2002 request service1 operation1 invoked"</u>

124

125 **POJO_2003_TestCase**

126

<u>Testcase ID</u>	<u>POJO_2003_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-2002</u>
<u>Description</u>	<u>Tests that a Java POJO implementation class implements all the operations defined by its service interface</u>
<u>Artifacts</u>	<u>POJO_2003_TestCase.java</u> <u>Test_POJO_2003.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>MultiOperationService1.java</u> <u>incompleteOperationsImpl.java</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

127

128 **2.2 Section 5**

129 **POJO_5001_TestCase**

130

<u>Testcase ID</u>	<u>POJO_5001_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-5001</u>
<u>Description</u>	<u>Tests that a Java POJO implementation class has either a public or a protected constructor</u>
<u>Artifacts</u>	<u>POJO_5001_TestCase.java</u> <u>Test_POJO_5001.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>service1PrivateConstructorImpl.java</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

131

132 | **POJO_5002_TestCase**

133 |

<u>Testcase ID</u>	<u>POJO_5002_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-5002</u>
<u>Description</u>	<u>Tests that where a POJO implementation has a constructor annotated with @Constructor, that this constructor is invoked when the implementation class is instantiated</u>
<u>Artifacts</u>	<u>POJO_5002_TestCase.java</u> <u>Test_POJO_5002.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.java</u> <u>service1AnnotatedConstructorImpl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_5002 request service1 operation1 invoked @Constructor was invoked"</u>

134 |

135 | **POJO_5003_TestCase**

136 |

<u>Testcase ID</u>	<u>POJO_5003_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-5003</u>
<u>Description</u>	<u>Tests that where a POJO implementation has no constructor annotated with @Constructor but has a constructor with all its parameters either annotated with @Property or annotated with @Reference, that this constructor is invoked when the implementation class is instantiated</u>
<u>Artifacts</u>	<u>POJO_5003_TestCase.java</u> <u>Test_POJO_5003.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.java</u> <u>service1AnnotatedParameterConstructorImpl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_5003 request service1 operation1 invoked fullAnnotated constructor was invoked"</u>

137

138 **POJO_5004_TestCase**

139

<u>Testcase ID</u>	<u>POJO_5004_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-5004</u>
<u>Description</u>	<u>Tests that where a POJO implementation has no constructor annotated with @Constructor and no constructor with all its parameters either annotated with @Property or annotated with @Reference, but it has a no-arg constructor, that the no-arg constructor is invoked when the implementation class is instantiated</u>
<u>Artifacts</u>	<u>POJO_5004_TestCase.java</u> <u>Test_POJO_5004.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.java</u> <u>service1NoArgConstructorImpl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_5004 request service1 operation1 invoked no-arg constructor was invoked"</u>

140

141 **POJO_5005_TestCase**

142

<u>Testcase ID</u>	<u>POJO_5005_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-5005</u>
<u>Description</u>	<u>Tests that where a POJO implementation has no constructor annotated with @Constructor and no constructor with all its parameters either annotated with @Property or annotated with @Reference and does not have a no-arg constructor, that no constructor is invoked and an error is thrown by the SCA runtime</u>
<u>Artifacts</u>	<u>POJO_5005_TestCase.java</u> <u>Test_POJO_5005.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.java</u>

	service1NoValidConstructorsImpl.java
Expected output	Negative test: "exception"

143

144 **[POJO_5006_TestCase](#)**

145

Testcase ID	POJO_5006_TestCase
Test Assertion	JCI-TA-5006
Description	Tests that a POJO implementation class has no more than 1 constructor annotated with @Constructor
Artifacts	POJO_5006_TestCase.java Test_POJO_5006.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1MultipleConstructorAnnotationsImpl.java
Expected output	Negative test: "exception"

146

147 **[POJO_5007_TestCase](#)**

148

Testcase ID	POJO_5007_TestCase
Test Assertion	JCI-TA-5007
Description	Tests that a POJO implementation class containing no @Constructor annotations has no more than 1 constructor with a non-empty parameter list where all the parameters are annotated with either @Property or with @Reference
Artifacts	POJO_5007_TestCase.java Test_POJO_5007.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java

	service1MultipleAnnotatedConstructorsImpl.java
Expected output	Negative test: "exception"

149

150 **2.3 [Section 8](#)**

151 **[POJO_6001_TestCase](#)**

152

Testcase ID	POJO_6001_TestCase
Test Assertion	JCA-TA-6001
Description	Tests that a POJO implementation annotated with @Scope("STATELESS") runs with the operational characteristics of a STATELESS scoped implementation.
Artifacts	POJO_6001_TestCase.java Test_POJO_6001.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl2java
Expected output	Positive test: "POJO_6001 request service1 operation1 invoked target stateless scope service invoked successfully"

153

154

Testcase ID	POJO_6002_TestCase
Test Assertion	Tests that a POJO implementation annotated with @Scope("COMPOSITE") runs with the operational characteristics of a COMPOSITE scoped implementation.
Description	Tests that
Artifacts	POJO_6002_TestCase.java Test_POJO_6002.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java

	service1CoordinatorImpl.java service1CompositeImpl.java
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_6002 request service1 operation1 invoked target composite scope service invoked successfully"</u>

155

156 **2.4 Section 8**

157 **POJO_8001_TestCase**

158

<u>Testcase ID</u>	POJO_8001_TestCase
<u>Test Assertion</u>	JCI-TA-8001
<u>Description</u>	<u>Tests that when a POJO implementation class has a @Service annotation with a single interface class in its value attribute and no name attribute, that its componentType has a <service/> subelement with a @name attribute which is the simple name of the Java interface class</u>
<u>Artifacts</u>	POJO_8001_TestCase.java Test_POJO_8001.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1Superset.java service1SupersetImpl.java
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_8001 request service1 operation1 invoked"</u>

159

160 **POJO_8002_TestCase**

161

<u>Testcase ID</u>	POJO_8002_TestCase
<u>Test Assertion</u>	JCA-TA-8002
<u>Description</u>	<u>Tests that when a POJO implementation class has a @Service annotation with a single interface class in its value attribute and no name attribute, that its componentType has a <service/> subelement with an <interface.-java/> subelement with the @interface attribute set to the fully qualified name of the interface class in the @Service annotation</u>
<u>Artifacts</u>	POJO_8002_TestCase.java

	Test_POJO_8002.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1Superset.java (1st copy in org.oasisopen.sca.test package) Service1Superset.java (2nd copy in default package) service1SupersetImpl.java
Expected output	Negative test: "exception"

162

163 **[POJO_8003_TestCase](#)**

164

Testcase ID	POJO_8003_TestCase
Test Assertion	JCI-TA-8003
Description	Tests that an SCA POJO implementation with an @Requires annotation on the class and a @Service annotation, has a <service/> element in its componentType which has a @requires attribute containing the set of intents contained in the @Requires annotation
Artifacts	POJO_8003_TestCase.java Test_POJO_8003.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java PropagatesTransaction.java service1WithIntentImpl.java
Expected output	Negative test: "exception"

165

166 **[POJO_8004_TestCase](#)**

167

Testcase ID	POJO_8004_TestCase
Test Assertion	JCI-TA-8004
Description	Tests that a POJO implementation with no @Service annotation that implements an interface where the interface class is annotated with @Remot-

	able has a componentType with a <service/> subelement with @name set to the simple name of the interface class and with an <interface.java/> subelement that is set to the fully qualified name of the interface class
Artifacts	POJO_8004_TestCase.java Test_POJO_8004.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1NoServiceAnnotationImpl.java
Expected output	Positive test: "POJO_8004 request service1 operation1 invoked"

168

169 **POJO_8005_TestCase**

170

Testcase ID	POJO_8005_TestCase
Test Assertion	JCI-TA-8005
Description	Tests that where a POJO implementation class has no @Service annotations and does not implement an interface class which is annotated with @Remotable, that the componentType of the implementation has a single <service/> element with @name set to the simple name of the implementation class and an <interface.java/> subelement with @interface set to the fully qualified name of the implementation class
Artifacts	POJO_8005_TestCase.java Test_POJO_8005.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java plainClassImpl.java
Expected output	Positive test: "POJO_8005 request service1 operation1 invoked"

171

172 **POJO_8006_TestCase**

Testcase ID	JCA_2001_TestCase
Test Assertion	JCA-TA-2001
Description	Tests that a Java service which is marked remotable does not use method-

	overloading
Artifacts	JCA_2001_TestCase.java Test_JCA_2001.composite TestInvocation.wsdl TestClient_0002.composite Service1Overload.java service1OverloadImpl.java
Expected output	Negative test: "exception"

173

Testcase ID	POJO_8006_TestCase
Test Assertion	JCI-TA-8006
Description	Tests that where a POJO implementation class has a @Reference annotation with the @name parameter, that the componentType of the implementation has a <reference/> element with its @name attribute set to the value of the @name parameter of the @Reference annotation
Artifacts	POJO_8006_TestCase.java Test_POJO_8006.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java service1RefWithNameImpl.java
Expected output	Positive test: "POJO_8006 request service1 operation1 invoked service2 operation1 invoked"

174 **JCA_2002_TestCase**

175

176 **POJO_8007_TestCase**

Testcase ID	JCA_2002_TestCase
Test Assertion	JCA-TA-2002
Description	Tests that a stateless-scoped Java implementation instance is not dispatched on more than a single thread at one time
Artifacts	JCA_2002_TestCase.java Test_JCA_2002.composite TestInvocation.wsdl TestClient_0002.composite Service1.java ParallelService.java parallelServiceClientImpl.java parallelServiceImpl.java
Expected output	Positive test: "JCA_2002 request service1 parallel service invocation successful"

177

<u>Testcase ID</u>	<u>POJO_8007_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-8008</u>
<u>Description</u>	<u>Tests that where a POJO implementation has a @Reference annotation with @required=false and the annotation annotates a field with an interface type, that the componentType of the implementation has a <reference/> element with @multiplicity set to "0..1"</u>
<u>Artifacts</u>	<u>POJO_8007_TestCase.java</u> <u>Test_POJO_8007.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.java</u> <u>service1Impl6.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_8007 request service1 operation1 invoked reference is null"</u>

178 **JCA_2003_TestCase**

179

180 **POJO 8008 TestCase**

<u>Testcase ID</u>	<u>JCA_2003_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-2003</u>
<u>Description</u>	<u>Tests that an implementation instance annotated with @Scope("STATE-LESS") is invoked only once through one business method during the life-time of the implementation instance</u>
<u>Artifacts</u>	<u>JCA_2003_TestCase.java</u> <u>Test_JCA_2003.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>MultipleService.java</u> <u>multipleServiceClientImpl.java</u> <u>multipleServiceImpl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"JCA_2003 request service1 multiple service invocation successful"</u>

181

<u>Testcase ID</u>	<u>POJO_8008_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-8009</u>
<u>Description</u>	<u>Tests that where a POJO implementation has a @Reference annotation which annotates a field which also has a @Requires annotation that the componentType of the implementation as a <reference/> element with a @requires attribute set to the value of the @Requires annotation</u>
<u>Artifacts</u>	<u>POJO_8008_TestCase.java</u>

	Test_POJO_8008.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java service1WithRefIntentImpl.java
Expected output	Negative test: "exception"

182 **JCA_2004_TestCase**

183

184 **POJO_8009_TestCase**

Testcase ID	JCA_2004_TestCase
Test Assertion	JCA-TA-2004
Description	Tests that where there is a Domain-level component implementation marked as COMPOSITE scope, that all its clients appear to interact with a single runtime instance of the class.
Artifacts	JCA_2004_TestCase.java Test_JCA_2004.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1CoordinatorImpl.java service1Impl2.java service1CompositeImpl.java
Expected output	Positive test: "JCA_2004 request serviceCoordinator operation1 invoked service1 operation1 invoked serviceComposite operation1 Initial service2 operation1 invoked serviceComposite operation1 1 service3 operation1 invoked serviceComposite operation1 2 service4 operation1 invoked serviceComposite operation1 3"

185

Testcase ID	POJO_8009_TestCase
Test Assertion	JCI-TA-8010
Description	Tests that a POJO implementation with a @Property annotation which annotates a setter method and the setter method parameter has a type which is not an array or a collection type has a componentType <property/> element with @name set to the JavaBeans property name derived from the name of the setter method and has @type set to the JAXB mapping of the parameter of the setter method
Artifacts	POJO_8009_TestCase.java Test_POJO_8009.composite TestInvocation.wsdl

	TestClient_0002.composite ASM_0002_Client.java Service1.java service1PropertySetterImpl.java
Expected output	Positive test: "POJO_8009 request service1 operation1 invoked 2002.09.13 09:00:00"

186 **JCA_2005_TestCase**

187

188 **POJO_8010_TestCase**

Testcase ID	JCA_2005_TestCase
Test Assertion	JCA-TA-2005
Description	Tests that where a component implementation is marked with COMPOSITE scope and with @EagerInit, that the implementation instance is created and initialized when the component is started.
Artifacts	JCA_2005_TestCase.java Test_JCA_2005.composite TestInvocation.wsdl TestClient_0002.composite Service1.java compositeEagerInitImpl.java service1CompositeImpl.java
Expected output	Positive test: "JCA_2005 request serviceComposite2 operation1 EagerInit called"

189

Testcase ID	POJO_8010_TestCase
Test Assertion	JCI-TA-8011
Description	Tests that a POJO implementation with no @Reference annotations and also with no @Property annotations and which has a public setter method which is not part of a service interface and which has a parameter typed by an interface which is annotated with @Remotable has a componentType with a <reference/> element with @name set to the Javabeans property name derived from the setter method name and @multiplicity of 1..1 and with an <interface.java/> subelement which references the fully qualified name of the interface class
Artifacts	POJO_8010_TestCase.java Test_POJO_8010.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java

	service1UnannotatedRefSetterImpl.java
Expected output	Positive test: "POJO_8010 request service1 operation1 invoked service2 operation1 invoked"

190 | **POJO_8011_TestCase**

191 | **JCA_2006_TestCase**

192 |

Testcase ID	POJO_8011_TestCase
Test Assertion	JCI-TA-8012
Description	Tests that a POJO implementation with no @Reference annotations and also with no @Property annotations and which has a public setter method which is not part of a service interface and which has a parameter which is not typed by an interface annotated with @Remotable or an array or a parameterized java.util.Collection has a componentType with a <property/> element with @name set to the Javabeans property name derived from the setter method name and @type set to the JAXB mapping of the type of the setter method and with @many set to "false"
Artifacts	POJO_8011_TestCase.java Test_POJO_8011.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1UnannotatedPropertySetterImpl.java
Expected output	Positive test: "POJO_8011 request service1 operation1 invoked"
Testcase ID	JCA_2006_TestCase
Test Assertion	JCA-TA-2006
Description	Tests that where a component implementation has a method marked with @Init, that this method is called when the implementation instance is created
Artifacts	JCA_2006_TestCase.java Test_JCA_2006.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1InitCheckerImpl.java service1InitImpl.java
Expected output	Positive test: "JCA_2006 request serviceComposite1 operation1 Init check succeeded"

193 |

194 | **POJO_8012JCA_2007_TestCase**

195

Testcase ID	POJO_8012_TestCase
Test Assertion	JCI-TA-8013
Description	Tests that a POJO implementation with a @Requires attribute on the class that the componentType has an <implementation.java/> subelement with a @requires attribute with the same value as the @Requires attribute
Artifacts	POJO_8012_TestCase.java Test_POJO_8012.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1WithIntentImpl.java
Expected output	Negative test: "exception"
Testcase ID	JCA_2007_TestCase
Test Assertion	JCA-TA-2007
Description	Tests that for a Java implementation with COMPOSITE scope, that multiple invocations of service operations which overlap in time can run within a single instance of the implementation on separate Java threads.
Artifacts	JCA_2007_TestCase.java Test_JCA_2007.composite TestInvocation.wsdl TestClient_0002.composite Service1.java ParallelService.java parallelCompositeClientImpl.java parallelCompositeServiceImpl.java
Expected output	Positive test: "JCA_2007 request serviceCompositeClient COMPOSITE service invocation successfully used by multiple threads simultaneously"

196

197 | **JCA_2008_TestCase**

198

199 | **POJO_8013_TestCase**

Testcase ID	JCA_2008_TestCase
Test Assertion	JCA-TA-2008
Description	Tests that for a Java implementation class with COMPOSITE scope offering a service, used as the implementation of a component within a composite which is itself the implementation of a Domain level component, all clients of the component service appear to interact with a single instance of the implementation class
Artifacts	JCA_2008_TestCase.java

	Test_JCA_2008.composite TestInvocation.wsdl TestClient_0002.composite CompositeScope.composite Service1.java service1CoordinatorImpl.java service1Impl2.java service1CompositeImpl.java
Expected output	Positive test: "JCA_2008 request serviceComposite-operation1 invoked service1-operation1 invoked serviceComposite-operation1 Initial service2-operation1 invoked serviceComposite-operation1-1 service3-operation1 invoked serviceComposite-operation1-2 service4-operation1 invoked serviceComposite-operation1-3"

200

Testcase ID	POJO_8013_TestCase
Test Assertion	JCI-TA-8014
Description	Tests that a POJO implementation class with 2 setter methods annotated with @Property have unique JavaBeans property names for the methods
Artifacts	POJO_8013_TestCase.java Test_POJO_8013.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1DupPropertySettersImpl.java
Expected output	Negative test: "exception"

201 **JCA_2009_TestCase**

202

203 **POJO_8014_TestCase**

Testcase ID	JCA_2009_TestCase
Test Assertion	JCA-TA-2011, JCA-TA-10004
Description	Tests that where one component is a client of a service provided by a second component, both with Java implementations and which both run in the same JVM, and the client reference is marked with @AllowsPassByReference but the service implementation methods are not marked with @AllowsPassByReference that invocations of the service use "pass by value" semantics.
Artifacts	JCA_2009_TestCase.java Test_JCA_2009.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service4.java

	service1Impl7.java service4Impl.java
Expected output	Positive test: "JCA_2009-request service1-operation1-invoked service2-operation1-in- voked-request+1"

204

Testcase ID	POJO_8014_TestCase
Test Assertion	JCI-TA-8015
Description	Tests that a POJO implementation class with 2 setter methods annotated with @Reference have unique JavaBeans property names for the methods
Artifacts	POJO_8014_TestCase.java Test_POJO_8014.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1DupReferenceSettersImpl.java
Expected output	Negative test: "exception"

205 **JCA_2010_TestCase**

206

207 **POJO_8015_TestCase**

Testcase ID	JCA_2010_TestCase
Test Assertion	JCA-TA-2011, JCA-TA-10004
Description	Tests that where one component is a client of a service provided by a second component, both with Java implementations and which both run in the same JVM, and the service implementation methods are not marked with @AllowsPassByReference but the client reference is not marked with @AllowsPassByReference that invocations of the service use "pass by value" semantics.
Artifacts	JCA_2010_TestCase.java Test_JCA_2010.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service4.java service1Impl7b.java service4Impl1.java
Expected output	Positive test: "JCA_2010-request service1-operation1-invoked service2-operation1-in- voked-request+1"

208

Testcase ID	POJO_8015_TestCase
-------------	--------------------

Test Assertion	JCI-TA-8007
Description	Tests that where a POJO implementation has a @Reference annotation with @required=true and the annotation annotates a field which is an array type, that the componentType of the implementation has a <reference/> element with @multiplicity set to "1..n"
Artifacts	POJO_8015_TestCase.java Test_POJO_8015.composite TestInvocation.wsdI TestClient_0002.composite ASM_0002_Client.java Service1.java Service1Impl.java Service1Impl3.java
Expected output	Positive test: "POJO_8015 request service1 operation1 invoked service2 operation1 invokedservice3 operation1 invoked"

209

210 **[POJO_8016_TestCase](#)**211 **2.5 Section 3**212 **[JCA_3001_TestCase](#)**

213

Testcase ID	POJO_8016_TestCase
Test Assertion	JCI-TA-8016
Description	Tests that a POJO implementation class with @Service annotation and @Remotable annotation, where the @Service annotation references an interface which does not contain an @Remotable annotation has a componentType with a <service/> element with <interface.java/> set with @interface set to the fully qualified name of the interface class in the value attribute of the @Service annotation and the @remotable attribute set to "true"
Artifacts	POJO_8016_TestCase.java Test_POJO_8016.composite TestInvocation.wsdI TestClient_0002.composite ASM_0002_Client.java Service1.java Service1NoRemotable.java

	Service1NoRemotableImpl.java
Expected output	Positive test: "POJO_8016 request service1 operation1 invoked"
Testcase ID	JCA_3001_TestCase
Test Assertion	JCA-TA-3001
Description	Tests that a Java Interface class name is fully qualified on a service <interface.java> element
Artifacts	JCA_3001_TestCase.java Test_JCA_3001.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1Impl.java
Expected output	Negative test: "exception"

214

215 **[POJO_8017JCA_3002_TestCase](#)**

216

Testcase ID	POJO_8017_TestCase
Test Assertion	JCI-TA-8017
Description	Tests that a POJO implementation class with @Reference annotation on a setter method where the interface type of the method has no @Remotable annotation but the setter method has a @Remotable annotation. the implementation has a componentType with a <reference/> element with <interface.java/> set with @interface set to the fully qualified name of the interface class which types the parameter of the setter method and the @remotable attribute set to "true"
Artifacts	POJO_8017_TestCase.java Test_POJO_8017.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1NoRemotable.java Service1NoRemotableRefImpl.java
Expected output	Positive test: "POJO_8017 request service1 operation1 invoked service2 operation1 invoked"
Testcase ID	JCA_3002_TestCase
Test Assertion	JCA-TA-3001
Description	Tests that a Java Interface is fully qualified on a reference <interface.java>

	element
Artifacts	JCA_3002_TestCase.java Test_JCA_3002.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1Impl.java Service1Impl2.java
Expected-output	Negative test: "exception"

217

218 **POJO_8018JCA_3003_TestCase**

219

Testcase ID	POJO_8018_TestCase
Test Assertion	JCI-TA-8018
Description	Tests that an implementation class with a setter method that takes an interface type parameter and is annotated with @Reference has a component-Type with a <reference/> element with an <interface.java/> subelement with @interface set to the fully qualified name of the interface type
Artifacts	POJO_8018_TestCase.java Test_POJO_8018.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ReferenceSetterImpl.java
Expected output	Positive test: "POJO_8018 request service1 operation1 invoked service2 operation1 invoked"
Testcase-ID	JCA_3003_TestCase
Test Assertion	JCA-TA-3002, JCA-TA-10031
Description	Tests that callback interfaces on a service are specified in their fully qualified form
Artifacts	JCA_3003_TestCase.java Test_JCA_3003.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1CallbackImpl.java Service3WithCallback.java Service3Callback.java service3Impl1.java

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

220

221 **POJO_8019JCA_3004_TestCase**

222

Testcase ID	POJO_8019_TestCase
Test Assertion	JCI-TA-8019
Description	Tests that an implementation class with a constructor parameter which has an interface type and is annotated with @Reference has a componentType with a <reference/> element with an <interface.java/> subelement with @interface set to the fully qualified name of the interface type
Artifacts	POJO_8019_TestCase.java Test_POJO_8019.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ReferenceConstructorImpl.java
Expected output	Positive test: "POJO_8019 request service1 operation1 invoked service2 operation1 invoked"
Testcase ID	JCA_3004_TestCase
Test Assertion	JCA-TA-3002, JCA-TA-10031
Description	Tests that callback interfaces on a reference are specified in their fully-qualified form
Artifacts	JCA_3004_TestCase.java Test_JCA_3004.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1CallbackImpl.java Service3WithCallback.java Service3Callback.java service3Impl1.java
Expected output	Negative test: "exception"

223

224 **POJO_8020JCA_3005_TestCase**

225

Testcase ID	POJO_8020_TestCase
-------------	--------------------

Test Assertion	JCI-TA-8020
Description	Tests that an implementation class with a @Service annotation and a @PolicySets annotation with a policySet declared has a componentType with a <service/> element with the @policySets attribute containing the policySet from the @PolicySets annotation
Artifacts	POJO_8020_TestCase.java Test_POJO_8020.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ServicePolicySetsImpl.java test:PolicySet3 in definitions.xml of contribution
Expected output	Positive test: "POJO_8020 request service1 operation1 invoked"
Testcase ID	JCA_3005_TestCase
Test Assertion	JCA-TA-3003, JCA-TA-10031
Description	Tests that callback interfaces specified in the composite match the callback interface specified in the service interface class
Artifacts	JCA_3005_TestCase.java Test_JCA_3005.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1CallbackImpl.java Service3WithCallback.java Service3Callback.java service3Impl1.java
Expected output	Positive test: "JCA_3005 request service1 operation1 invoked service3 operation1 invoked service1 callback1 invoked"

226

227 **[POJO_8021JCA_3006_TestCase](#)**

228

Testcase ID	POJO_8021_TestCase
Test Assertion	JCI-TA-8021
Description	Tests that an implementation class with a @Reference annotation and a @PolicySets annotation with a policySet declared has a componentType with a <reference/> element with the @policySets attribute containing the policySet from the @PolicySets annotation
Artifacts	POJO_8021_TestCase.java

	Test_POJO_8021.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ReferencePolicySetsImpl.java
Expected output	Positive test: "POJO_8021 request service1 operation1 invoked service2 operation1 invoked"
Testcase ID	JCA_3006_TestCase
Test Assertion	JCA-TA-3003, JCA-TA-10031
Description	Tests that callback interfaces specified in the composite match the callback interface specified in the service interface class
Artifacts	JCA_3006_TestCase.java Test_JCA_3006.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1CallbackImpl.java Service3WithCallback.java Service3Callback.java service3Impl1.java
Expected output	Negative test: "exception"

229

230 **[POJO_8022JCA_3007_TestCase](#)**

231

Testcase ID	POJO_8022_TestCase
Test Assertion	JCI-TA-8022
Description	Tests that an implementation class with zero @Service, @Reference, @Property annotations and a @PolicySets annotation on the class with one policy set declared, has a componentType with a <service/> element with a @policySets attribute present containing the policy set declared in the @PolicySets annotation
Artifacts	POJO_8022_TestCase.java Test_POJO_8022.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java

	Service1UnannotatedWithPolicySetsImpl.java
<u>Expected output</u>	Positive test: "POJO_8022 request service1 operation1 invoked service2 operation1 invoked"
<u>Testcase ID</u>	JCA_3007_TestCase
<u>Test Assertion</u>	JCA-TA-3003, JCA-TA-10031
<u>Description</u>	Tests that callback interfaces specified in the composite match the callback interface specified in the reference interface class
<u>Artifacts</u>	JCA_3007_TestCase.java Test_JCA_3007.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1CallbackImpl.java Service3WithCallback.java Service3Callback.java service3Impl1.java
<u>Expected output</u>	Negative test: "exception"

232

233 **POJO_8023JCA_3008_TestCase**

234

<u>Testcase ID</u>	POJO_8023_TestCase
<u>Test Assertion</u>	JCI-TA-8023
<u>Description</u>	Tests that an implementation class with zero @Service, @Reference, @Property annotations and a @PolicySets annotation on a public field, typed by an interface annotated with @Remotable, with one policy set declared, has a componentType with a <reference/> element with a @policySets attribute present containing the policy set declared in the @PolicySets annotation
<u>Artifacts</u>	POJO_8023_TestCase.java Test_POJO_8023.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1UnannotatedWithRefPolicySetsImpl.java
<u>Expected output</u>	Positive test: "POJO_8023 request service1 operation1 invoked service2 operation1 invoked"
<u>Testcase ID</u>	JCA_3008_TestCase

Test Assertion	JCA-TA-3004
Description	Tests that <interface.java/> conforms to the schema
Artifacts	JCA_3008_TestCase.java Test_JCA_3008.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java
Expected output	Negative test: "exception"

235

236 **POJO_8024JCA_3009_TestCase**

237

Testcase ID	POJO_8024_TestCase
Test Assertion	JCI-TA-8024
Description	Tests that an implementation class has a @PolicySets annotation on the class with one policy set declared, has a componentType with a <implementation.java/> element with a @policySets attribute present containing the policy set declared in the @PolicySets annotation
Artifacts	POJO_8024_TestCase.java Test_POJO_8024.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ReferencePolicySetsImpl.java
Expected output	Positive test: "POJO_8024 request service1 operation1 invoked service2 operation1 invoked"
Testcase ID	JCA_3009_TestCase
Test Assertion	JCA-TA-3005
Description	Tests that remotable attribute matches @Remotable annotation
Artifacts	JCA_3009_TestCase.java Test_JCA_3009.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java
Expected output	Negative test: "exception"

238

239 | **POJO_8025JCA_3010_TestCase**

240

Testcase ID	POJO_8025_TestCase
Test Assertion	JCI-TA-8025
Description	Tests that an implementation class with zero @Service, @Reference, @Property annotations, which has a public setter method, not part of a service interface, with a parameter type which is an array of interface classes annotated with @Remotable has a componentType with a <reference/> element with @name set to the JavaBeans property name from the setter method name, @multiplicity set to 1..n and an <interface.java/> subelement which references the fully qualified name of the interface class
Artifacts	POJO_8025_TestCase.java Test_POJO_8025.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ReferencePolicySetsImpl.java
Expected output	Positive test: "POJO_8025 request service1 operation1 invoked service2 operation1 invoked"
Testcase ID	JCA_3010_TestCase
Test Assertion	JCA-TA-3005
Description	Tests that remotable attribute matches @Remotable annotation
Artifacts	JCA_3010_TestCase.java Test_JCA_3010.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java
Expected output	Positive test: "JCA_3010 request service1 operation1 invoked"

241

242 | **POJO_8026JCA_3011_TestCase**

243

Testcase ID	POJO_8026_TestCase
Test Assertion	JCI-TA-8026
Description	Tests that an implementation class with zero @Service, @Reference, @Property annotations, which has a public setter method, not part of a service interface, with a parameter type which is a java.util.Collection parameterized by an interface class annotated with @Remotable has a com-

	ponentType with a <reference/> element with @name set to the Java-Beans property name from the setter method name, @multiplicity set to 1..n and an <interface.java/> subelement which references the fully qualified name of the interface class
Artifacts	POJO_8026_TestCase.java Test_POJO_8026.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1ReferencePolicySetsImpl.java
Expected output	Positive test: "POJO_8026 request service1 operation1 invoked service2 operation1 invoked"
Testcase ID	JCA_3011_TestCase
Test Assertion	JCA-TA-3006
Description	Tests that a service interfaces doesn't contain forbidden annotations
Artifacts	JCA_3011_TestCase.java Test_JCA_3011.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java JCA3011Service.java JCA3011serviceImpl.java
Expected output	Negative test: "exception"

244

245 **[POJO_8027JCA_3012_TestCase](#)**

246

Testcase ID	POJO_8027_TestCase
Test Assertion	JCI-TA-8027
Description	Tests that an implementation class with zero @Service, @Reference, @Property annotations, which has a public setter method, not part of a service interface, with a parameter type which is a java.util.Collection not parameterized by an interface class annotated with @Remotable has a componentType with a <property/> element with @name set to the Java-Beans property name from the setter method name, @type set to the JAXB mapping of the type of the setter parameter and @many set to "true"
Artifacts	POJO_8027_TestCase.java Test_POJO_8027.composite TestInvocation.wsdl

	TestClient_0002.composite ASM_0002_Client.java Service1.java service1Impl.java Service1UnannotatedManyPropertySetterImpl.java
<u>Expected output</u>	<u>Positive test:</u> "POJO_8027 request service1 operation1 invoked 2002.09.13 09:00:00 2010.10.28 23:05:13":
<u>Testcase ID</u>	JCA_3012_TestCase
<u>Test Assertion</u>	JCA-TA-3007
<u>Description</u>	Tests that callback interfaces don't contain forbidden annotations
<u>Artifacts</u>	JCA_3012_TestCase.java Test_JCA_3012.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service3WithCallback.java Service1.java service1CallbackImpl.java Service3Callback.java JCA3012Service3WithCallback.java JCA3012Service3Callback.java JCA3012service3Impl1.java
<u>Expected output</u>	<u>Negative test:</u> "exception"

247

248 **[POJO_8028JCA_3013_TestCase](#)**

249

<u>Testcase ID</u>	POJO_8028_TestCase
<u>Test Assertion</u>	JCI-TA-8028
<u>Description</u>	Tests that an implementation class with a javax.jws.WebService annotation and no @Service annotation has a component type with a <service/> element with @name set to the name property on the @WebService annotation and an <interface.java/> subelement with the @interface attribute set to the fully qualified name of the implementation class
<u>Artifacts</u>	POJO_8028_TestCase.java Test_POJO_8028.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebServiceImpl.java

Expected output	Positive test: "POJO_8028 request service1 operation1 invoked";
Testcase ID	JCA_3013_TestCase
Test Assertion	JCA-TA-3008
Description	Tests that where two Java interfaces are otherwise compatible, every method present in both interfaces where annotated with @OneWay in one interface is also annotated with @OneWay in the other interface
Artifacts	JCA_3013_TestCase.java Test_JCA_3013.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java ServiceOneWayMissing.java serviceOneWayImpl.java Service1OneWayMissingCallerImpl.java ServiceOneWay.java
Expected output	Negative test: "exception"

250

251 **POJO_8029JCA_3014_TestCase**

252

Testcase ID	POJO_8029_TestCase
Test Assertion	JCI-TA-8029
Description	Tests that an implementation class with a javax.jws.WebService annotation and no @Service annotation, where the @EndpointInterface annotation property is set to the name of a Java interface class, has a component type with a <service/> element with @name set to the name property on the @WebService annotation and an <interface.java/> subelement with the @interface attribute set to the value of the EndpointInterface property
Artifacts	POJO_8029_TestCase.java Test_POJO_8029.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebServiceEPIImpl.java
Expected output	Positive test: "POJO_8029 request service1 operation1 invoked";
Testcase ID	JCA_3014_TestCase
Test Assertion	JCA-TA-3009
Description	Tests that where an <interface.java/> element references a Java interface class which contains a @WebService annotation with a non-empty wsdlLocation property pointing at a WSDL document, it is treated as if it were an <interface.wsdl/> element with an @interface attribute identifying

	the portType in the WSDL document mapped from the Java interface class-
Artifacts	JCA_3014_TestCase.java Test_JCA_3014.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service3Operations.java Service3OperationsWSDL.java Service3OperationsWSDLImpl.java Service1Calls3OperationsImpl.java Service3OperationsWSDL.wsdl
Expected output	Negative test: "exception"

253

254 **POJO_8030_TestCase**

255

Testcase ID	POJO_8030_TestCase
Test Assertion	JCI-TA-8030
Description	Tests that an implementation class with a javax.jws.WebService annotation and no @Service annotation, where the @WsdLocation annotation property is set to a WSDL document, has a component type with a <service/> element with @name set to the name property on the @WebService annotation and an <interface.wsdl/> subelement with the @interface attribute set to the portType referenced by the value of the WsdLocation property
Artifacts	POJO_8030_TestCase.java Test_POJO_8030.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebServiceWSDLImpl.java
Expected output	Negative test: "exception"

256 **2.6 Section 4**

257 **JCA_4001_TestCase**

258

259 **POJO_8031_TestCase**

Testcase ID	JCA_4001_TestCase
Test Assertion	JCA-TA-4001, JCA-TA-10005, JCA-TA-10006
Description	Tests that for a stateless Java implementation, that all lifecycle stages are-

	performed in the correct sequence and that no stage occurs out of sequence.
Artifacts	JCA_4001_TestCase.java Test_JCA_4001.composite TestInvocation.wsdl TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1StatelessLifecycleImpl.java service1Impl.java dataStoreCompositeImpl.java
Expected output	"JCA_4001 request Constructing property1 injected reference1 injected property2 injected reference2 injected Init invoked Init completed serviceLifecycle operation1 invoked service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked prop1 prop2 prop3 Destroy invoked"

260

Testcase ID	POJO_8031_TestCase
Test Assertion	JCI-TA-8031
Description	Tests that an implementation class with a service method with <u>javax.jws.WebParam</u> annotation on a parameter with the <u>@header</u> property set to "true", has a component type with a <u><service/></u> element with <u>@requires</u> attribute containing the "sca:SOAP" intent
Artifacts	POJO_8031_TestCase.java Test_POJO_8031.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebParamImpl.java
Expected output	Negative test: "exception"

261 **JCA_4002_TestCase**

262

263 **POJO_8032_TestCase**

Testcase ID	JCA_4002_TestCase
Test Assertion	JCA-TA-4004
Description	Tests that where a stateless Java implementation throws an exception from its Constructor method, that the implementation transitions to the Terminating state
Artifacts	JCA_4002_TestCase.java Test_JCA_4002.composite TestInvocation.wsdl

	TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1LifecycleExceptionsImpl.java service1Impl.java dataStoreCompositeImpl.java
Expected output	"JCA_4002 request Constructing exception thrown"

264

Testcase ID	POJO_8032_TestCase
Test Assertion	JCI-TA-8032
Description	Tests that an implementation class with a service method with javax.jws.WebResult annotation on a parameter with the @header property set to "true", has a component type with a <service/> element with @requires attribute containing the "sca:SOAP" intent
Artifacts	POJO_8032_TestCase.java Test_POJO_8032.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebResultImpl.java
Expected output	Negative test: "exception"

265 **JCA_4003_TestCase**

266

267 **POJO_8033_TestCase**

Testcase ID	JCA_4003_TestCase
Test Assertion	JCA-TA-4010
Description	Tests that when a stateless Java implementation throws an exception when a property is injected, that the implementation transitions to the Destroying state
Artifacts	JCA_4003_TestCase.java Test_JCA_4003.composite TestInvocation.wsdl TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1LifecycleExceptionsImpl.java service1Impl.java dataStoreCompositeImpl.java
Expected output	"JCA_4003 request Constructing property1 injected reference1 injected-

	property2 injected exception thrown Destroy invoked"
268	
<u>Testcase ID</u>	<u>POJO_8033_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-8033</u>
<u>Description</u>	<u>Tests that an implementation class annotated with a javax.jws.soap.SOAP-Binding annotation has a component type with a <service/> element with @requires attribute containing the "sca:SOAP" intent</u>
<u>Artifacts</u>	<u>POJO_8033_TestCase.java</u> <u>Test_POJO_8033.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>ASM_0002_Client.java</u> <u>Service1.java</u> <u>Service1JWSSOAPBindingImpl.java</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

269 **JCA_4004_TestCase**

270

271 **POJO 8034 TestCase**

<u>Testcase ID</u>	<u>JCA_4004_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-4012</u>
<u>Description</u>	<u>Tests that where a Java implementation invokes an injected reference while in the initializing phase and the target of the reference has not been initialized, that the implementation receives a ServiceUnavailableException</u>
<u>Artifacts</u>	<u>JCA_4004_TestCase.java</u> <u>Test_JCA_4004.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>JCA_0002_Client.java</u> <u>Service1.wsdl</u> <u>DataStore.java</u> <u>lifecycleControllerImpl.java</u> <u>service1LifecycleExceptionsImpl.java</u> <u>service1UninitImpl.java</u> <u>service1Impl.java</u> <u>dataStoreCompositeImpl.java</u>
<u>Expected output</u>	<u>"JCA_4004 request Constructing property1 injected reference1 injected property2 injected reference2 injected Init invoked calling uninitialized service ServiceUnavailable exception received Init completed serviceLifecycle operation1 invoked service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked prop1 prop2 prop3 Destroy invoked"</u>

272

<u>Testcase ID</u>	<u>POJO_8034_TestCase</u>
--------------------	---------------------------

<u>Test Assertion</u>	JCI-TA-8034
<u>Description</u>	Tests that an implementation class with a <code>javax.xml.ws.WebServiceProvider</code> annotation, has a component type with a <code><service/></code> element with an <code><interface.java/></code> subelement set to the fully qualified name of the interface class
<u>Artifacts</u>	POJO_8034_TestCase.java Test_POJO_8034.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebServiceProviderImpl.java
<u>Expected output</u>	Positive test: "POJO_8034 request service1 operation1 invoked";

273 **JCA_4005_TestCase**

274

275 **POJO_8035_TestCase**

<u>Testcase ID</u>	JCA_4005_TestCase
<u>Test Assertion</u>	JCA-TA-4015
<u>Description</u>	Tests that where a Java implementation throws an exception from the method marked with the <code>@Init</code> annotation, that the implementation transitions to the Destroying state
<u>Artifacts</u>	JCA_4005_TestCase.java Test_JCA_4005.composite TestInvocation.wsdl TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1LifecycleExceptionsImpl.java service1Impl.java dataStoreCompositeImpl.java
<u>Expected output</u>	"JCA_4005 request Constructing property1 injected reference1 injected property2 injected reference2 injected Init invoked exception thrown Destroy invoked"

276

<u>Testcase ID</u>	POJO_8035_TestCase
<u>Test Assertion</u>	JCI-TA-8035
<u>Description</u>	Tests that an implementation class with a <code>javax.xml.ws.WebServiceProvider</code> annotation which has a <code>wsdlLocation</code> property set, pointing to a WSDL document, has a component type with a <code><service/></code> element with an <code><interface.wsdl/></code> subelement set to the portType referenced by the value of the <code>wsdlLocation</code> property

<u>Artifacts</u>	POJO_8035_TestCase.java Test_POJO_8035.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebServiceProviderImpl.java
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

277 | **JCA_4006_TestCase**

278 |

279 | **POJO_8036_TestCase**

<u>Testcase ID</u>	JCA_4006_TestCase
<u>Test Assertion</u>	JCA-TA-4019
<u>Description</u>	Tests that a Java implementation in the Destroying state which invokes a method on a reference whose target service has already been destroyed receives an InvalidServiceException
<u>Artifacts</u>	JCA_4006_TestCase.java Test_JCA_4006.composite TestInvocation.wsdl TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1StatelessLifecycleImpl.java service1Impl.java dataStoreCompositImpl.java
<u>Expected output</u>	"JCA_4006 request Constructing property1 injected reference1 injected-property2 injected reference2 injected Init invoked Init completed serviceLifecycle operation1 invoked service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked prop1 prop2 prop3 Destroy invoked"

280 |

<u>Testcase ID</u>	POJO_8036_TestCase
<u>Test Assertion</u>	JCI-TA-8036
<u>Description</u>	Tests that an implementation class annotated with any JAX-WS annotation has a component type with a <service/> element with a <binding.ws/> subelement with the @wsdlElement attribute set to point to a WSDL document that uses the SOAP/HTTP binding
<u>Artifacts</u>	POJO_8036_TestCase.java Test_POJO_8036.composite TestInvocation.wsdl

	TestClient_0002.composite ASM_0002_Client.java Service1.java Service1JWSWebServiceImpl.java
Expected output	Positive test: "POJO_8036 request service1 operation1 invoked";

281 **JCA_4007_TestCase**

282

283 **POJO_8037_TestCase**

Testcase ID	JCA_4007_TestCase
Test Assertion	JCA-TA-4022
Description	Tests that where a Java implementation throws an exception from its method marked with the @Destroy annotation, that the implementation transitions to the terminated state
Artifacts	JCA_4007_TestCase.java Test_JCA_4007.composite TestInvocation.wsdl TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1LifecycleExceptionsImpl.java service1Impl.java dataStoreCompositeImpl.java
Expected output	"JCA_4007 request Constructing property1 injected reference1 injected property2 injected reference2 injected Init invoked Init completed serviceLifecycle operation1 invoked service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked prop1 prop2 prop3 Destroy invoked exception thrown"

284

Testcase ID	POJO_8037_TestCase
Test Assertion	JCI-TA-8037
Description	Tests that an implementation class annotated with a javax.xml.ws.BindingType annotation, where the value of the @BindingType annotation is http://www.w3.org/2003/05/soap/bindings/HTTP/, has a component type with a <binding.ws/> element with an @requires attribute which contains the intent "SOAP.v1_2"
Artifacts	POJO_8037_TestCase.java Test_POJO_8037.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java

	Service1.java Service1JWSBindingTypeImpl.java
Expected output	Negative test: "exception"

285 **JCA_4008_TestCase**

286

Testcase ID	JCA_4008_TestCase
Test Assertion	JCA-TA-4024
Description	Tests that when a stateless Java implementation throws an exception when a reference is injected, that the implementation transitions to the Destroying state
Artifacts	JCA_4008_TestCase.java Test_JCA_4008.composite TestInvocation.wsdl TestClient_0002.composite JCA_0002_Client.java Service1.wsdl DataStore.java lifecycleControllerImpl.java service1LifecycleExceptionsImpl.java service1Impl.java dataStoreCompositeImpl.java
Expected output	"JCA_4008 request Constructing property1 injected reference1 injected-property2 injected reference2 injected-exception thrown Destroy invoked"

287

288 **2.7 Section 95**

289 **POJO_9JCA_7001_TestCase**

290

Testcase ID	POJO_9001_TestCase
Test Assertion	JCI-TA-9001
Description	Tests that if an SCA <component/> element has an <implementation.java/> subelement, that the <implementation.java/> element conforms to the sca-implementation-java.xsd schema
Artifacts	POJO_9001_TestCase.java Test_POJO_9001.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1Impl.java
Expected output	Negative test:

	<u>"exception"</u>
--	--------------------

291

292 **POJO_9002_TestCase**

293

<u>Testcase ID</u>	<u>POJO_9002_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-9002</u>
<u>Description</u>	<u>Tests that where a <component/> has an <implementation.java/> subelement with a @class attribute that has the fully qualified name of a Java class that is contained in the same contribution as the composite file containing the <component/>, that the Java class is found and instantiated</u>
<u>Artifacts</u>	<u>POJO_9002_TestCase.java</u> <u>Test_POJO_9002.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>x.service1Impl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_9002 request service1 operation1 invoked"</u>

294

295 **POJO_9003_TestCase**

296

<u>Testcase ID</u>	<u>POJO_9003_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-9003</u>
<u>Description</u>	<u>Tests that if an <implementation.java/> element has a @class attribute which references a Java POJO implementation class and that class is imported by the contribution containing the composite from a second contribution, the version of the class from the second contribution is loaded and used even where a version of the same class is available in the contribution containing the composite</u>
<u>Artifacts</u>	<u>POJO_9003_TestCase.java</u> <u>Test_POJO_9003.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>x.service1.java (2 copies #1 = contribution POJO_General; #2 = contribution POJO_9003)</u>

<u>Expected output</u>	<u>Positive test:</u> "POJO_9003 request service1 operation1"
------------------------	--

297

298 **POJO_9004_TestCase**

299

<u>Testcase ID</u>	<u>POJO_9004_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-9004</u>
<u>Description</u>	<u>Tests that for an <implementation.java/> element in a composite file that specifies the name of a Java POJO class in its @class attribute, where a Java class of that fully qualified name exists a) in the same contribution (A) as the composite file b) in two other contributions in the Domain (B & C), both of which export the namespace of the Java class and where contribution A imports the namespace with <import.java/> specifying @location referencing only one of the Contributions (C) that the Java class is loaded from Contribution C.</u>
<u>Artifacts</u>	<u>POJO_9004_TestCase.java</u> <u>Test_POJO_9004.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>service1ContributionImpl.java</u> <u>contribution POJO_9004</u> <u>contribution ContributionB</u> <u>contribution ContributionC</u>
<u>Expected output</u>	<u>Positive test:</u> "POJO_9004 request service1 operation1 invoked contributionC"

300

301

302 **2.8 Section 10**

303 **POJO_10001_TestCase**

304

<u>Testcase ID</u>	<u>POJO_10001_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10001</u>
<u>Description</u>	<u>Tests that where an sca-contribution.xml file contains multiple <import.java/> elements, that the value of the @package attribute of each element is unique across all the <import.java/> elements</u>

<u>Artifacts</u>	<u>POJO_10001_TestCase.java</u> <u>Test_POJO_10001.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite1.composite</u> <u>Service1.java</u> <u>service1Impl.java</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

305

306 **POJO_10002_TestCase**

307

<u>Testcase ID</u>	<u>POJO_10002_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10002</u>
<u>Description</u>	<u>Tests that where a contribution contains an SCA component that refer-</u> <u>ences classes from a Java package not contained in the contribution, and</u> <u>where the sca-contribution.xml contains an <import.java/> for that Java</u> <u>package, and there is a second contribution in the Domain that contains</u> <u>the classes of the package and which exports the package, that the SCA</u> <u>component can be instantiated and that it uses the Java classes from the</u> <u>second contribution.</u>
<u>Artifacts</u>	<u>POJO_10002_TestCase.java</u> <u>Test_POJO_10002.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>service1Impl.java</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_10002 request service1 operation1 invoked"</u>

308 **POJO_10003_TestCase**

309

<u>Testcase ID</u>	<u>POJO_10003_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10003</u>
<u>Description</u>	<u>Tests that where a contribution A contains a component with an imple-</u> <u>mentation from a Java package not found in contribution A and where the</u> <u>sca-contribution.xml has an <import.java/> for that package with a @loca-</u> <u>tion attribute pointing at a contribution B and contribution B both contains</u>

	<u>classes for that Java package and also exports that package, that the classes from contribution B satisfy the import from contribution A and that the component is instantiated.</u>
<u>Artifacts</u>	<u>POJO_10003_TestCase.java</u> <u>Test_POJO_10003.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>service1ContributionImpl.java</u> <u><POJO_ContributionB></u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_10003 request service1 operation1 invoked ContributionB"</u>

310

311 **POJO_10004_TestCase**

312

<u>Testcase ID</u>	<u>POJO_10004_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10004</u>
<u>Description</u>	<u>Tests that where a contribution A has references to Java artifacts from 2 different Java packages X and Y and where both of these packages are imported, and where package X is exported by a contribution B with a uses statement specifying a specific version of package Y, that import of Y in contribution A must import the same version of Y as specified in the uses attribute in contribution B.</u>
<u>Artifacts</u>	<u>POJO_10004_TestCase.java</u> <u>Test_POJO_10004.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>service1ContributionImpl.java</u> <u><POJO_ContributionC></u> <u><POJO_ContributionD></u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

313

314 **POJO_10005_TestCase**

315

<u>Testcase ID</u>	<u>POJO_10005_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10005</u>
<u>Description</u>	<u>Tests that where a contribution has an sca-contribution.xml file with multiple <export.java/> elements, that the @package attribute value of each element is unique</u>
<u>Artifacts</u>	<u>POJO_10005_TestCase.java</u> <u>Test_POJO_10005.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite1.composite</u> <u>Service1.java</u> <u>service1Impl.java</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

316

317 **POJO_10006_TestCase**

318

<u>Testcase ID</u>	<u>POJO_10006_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10006</u>
<u>Description</u>	<u>Tests that where a contribution has an sca-contribution.xml file with an <export.java/> element which references a given Java package, that the files of that package are contained within the contribution</u>
<u>Artifacts</u>	<u>POJO_10006_TestCase.java</u> <u>Test_POJO_10006.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite1.composite</u> <u>Service1.java</u> <u>service1Impl.java</u>
<u>Expected output</u>	<u>Negative test:</u> <u>"exception"</u>

319

320 **POJO_10007_TestCase**

321

<u>Testcase ID</u>	<u>POJO_10007_TestCase</u>
--------------------	----------------------------

<u>Test Assertion</u>	<u>JCI-TA-10007</u>
<u>Description</u>	<u>Tests that all Java classes loaded from a Contribution are loaded by a class loader that is unique to the contribution</u>
<u>Artifacts</u>	<u>POJO_10007_TestCase.java</u> <u>Test_POJO_10007.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>TestComposite1.composite</u> <u>Service1.java</u> <u>x.service1ComplexClient.java</u> <u>x.serviceComplex1Impl.java</u> <u>y.ServiceComplex1.java</u> <u>z.ComplexClass.java</u> <u><POJO_10007 contribution></u> <u><POJO_ContributionE contribution></u> <u><POJO_ContributionF contribution></u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_10007 request service1 operation1 invoked service2 operation1 invoked Service interface classloader same as Impl class classloader ComplexClass classloader different from Impl class classloader"</u>

322

323 **POJO_10008_TestCase**

324

<u>Testcase ID</u>	<u>POJO_10008_TestCase</u>
<u>Test Assertion</u>	<u>JCI-TA-10008</u>
<u>Description</u>	<u>Tests that where a contribution A uses one or more Java classes that are imported from a contribution B, that contribution B's classloader is used to load the classes imported into contribution A</u>
<u>Artifacts</u>	<u>POJO_10008_TestCase.java</u> <u>Test_POJO_10008.composite</u> <u>TestComposite1.composite (contribution POJO_ContributionE)</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u>

	service1ComplexClient.java (contribution POJO_10008) x.serviceComplex1Impl.java y.ServiceComplex1.java z.ComplexClass.java <POJO_10008 contribution> <POJO_ContributionE contribution>
<u>Expected output</u>	<u>Positive test:</u> "POJO_10008 request service1 operation1 invoked classloaders set correctly"
<u>Testcase ID</u>	JCA_7001_TestCase
<u>Test Assertion</u>	JCA-TA-7001 JCA-TA-7002
<u>Description</u>	Tests that where a Java implementation offers a bidirectional service and has fields and setter methods annotated with @Callback, when the bidirectional service is invoked, a callback reference object is injected into those fields and setter methods which have a type which is the interface for the callback service but null is injected into fields and setter methods which have a different type.
<u>Artifacts</u>	JCA_7001_TestCase.java Test_JCA_7001.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service3WithCallback.java Service3Callback.java Service7WithCallback.java Service7Callback.java service1CallbackImpl.java MultipleCallbacksImpl.java
<u>Expected-output</u>	"JCA_7001 request service1 operation1 invoked service2 operation1 invoked callback1 invoked other callbacks null";

325

326 **POJO_10009JCA_7002_TestCase**

327

<u>Testcase ID</u>	POJO_10009_TestCase
<u>Test Assertion</u>	JCI-TA-10009
<u>Description</u>	Tests that the thread context classloader of the thread used to invoke an operation of a Java POJO component implementation is the class loader of the contribution that contains the POJO implementation class.
<u>Artifacts</u>	POJO_10009_TestCase.java Test_POJO_10009.composite TestInvocation.wsdl TestClient_0002.composite

	<u>TestComposite1.composite</u> <u>Service1.wsdl</u>
<u>Expected output</u>	<u>Positive test:</u> <u>"POJO_10009 request service1 operation1 invoked thread context class-loader is correct"</u>
<u>Testcase ID</u>	<u>JCA_7002_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-7002</u>
<u>Description</u>	<u>Tests that where a Java implementation offers a bidirectional service and also has one non-bidirectional service and has fields and setter methods annotated with @Callback, when the non-bidirectional service is invoked, null is injected into all the fields and setter methods annotated with @Callback</u>
<u>Artifacts</u>	<u>JCA_7002_TestCase.java</u> <u>Test_JCA_7002.composite</u> <u>TestInvocation.wsdl</u> <u>TestClient_0002.composite</u> <u>Service1.java</u> <u>Service3WithCallback.java</u> <u>Service3Callback.java</u> <u>Service7WithCallback.java</u> <u>Service7Callback.java</u> <u>service1CallbackImpl.java</u> <u>MultipleCallbacksImpl.java</u>
<u>Expected output</u>	<u>"JCA_7002 request service1 operation1 invoked service2 operation1 invoked callback1 invoked other callbacks null";</u>

329 **3 Catalog of Test Artifacts**

330 **JCA_7003_TestCase**

331

332 **3.1 Java Interfaces**

Testcase ID	JCA_7003_TestCase
Test Assertion	JCA-TA-7003 JCA-TA-7004 JCA-TA-7005 JCA-TA-7007
Description	Tests that where a Java interface class is an asynchronous service mapping of a WSDL portType containing request/response operations that the class is annotated with the "asyncInvocation" intent
Artifacts	JCA_7003_TestCase.java Test_JCA_7003.composite TestInvocation.wsdl TestClient_0002.composite Service1AsyncServer.java Service1AsyncServerImpl.java
Expected output	"JCA_7003 request service1 operation1 invoked asynchronously"

333

<u>Name</u>	<u>Description</u>
<u>MultipleService.java</u>	<u>Service interface for checking multiple service invocations of a stateless Java implementation</u>
<u>Service1.java</u>	<u>Service1 interface</u>
<u>Service1NoRemotable.java</u>	<u>Service1 interface without a @Remotable annotation</u>
<u>Service1Superset.java</u>	<u>Interface with 2 operations</u> <u>- each with String as input and String response</u>
<u>ServiceComplex1.java</u>	<u>Interface with 2 operations</u> <u>- 1 with Java class as input and String response</u> <u>- 1 with void input and String response</u>

334 **3.2 Java Implementation Classes**

335 **JCA_7004_TestCase**

336

<u>Name</u>	<u>Description</u>
-------------	--------------------

<u>All classes in the org.oasisopen.sca.test package unless noted</u>	
<u>incompleteOperationsImpl.java</u> <POJO_2003>	<u>1 service with Service1Superset. interface</u> <u>- operation2 is not implemented</u>
<u>multipleServiceClientImpl.java</u>	<u>1 service with Service1 interface</u> <u>1 reference with ParallelService interface</u>
<u>multipleServiceImpl.java</u>	<u>1 service with MultipleService interface</u>
<u>plainClassImpl.java</u>	<u>1 service with interface defined by the class itself</u>
<u>service1AnnotatedConstructorImpl.java</u>	<u>1 service with Service1 interface</u> <u>one constructor annotated with @Constructor</u>
<u>service1AnnotatedParameterConstructorImpl.java</u>	<u>1 service with Service1 interface</u> <u>one constructor annotated with @Constructor and all parameters annotated with either @Property or @Reference</u>
<u>service1ClassLoaderCheckImpl.java</u>	<u>1 service with Service1 interface</u> <u>Performs a check on the ThreadContextClassLoader used to invoke a service operation</u>
<u>service1ComplexClientImpl.java</u> <POJO_10008>	<u>1 service with Service1 interface</u> <u>1 reference with ServiceComplex1 interface</u>
<u>x.service1ComplexClientImpl.java</u> <POJO_ContributionE>	<u>1 service with Service1 interface</u> <u>1 reference with ServiceComplex1 interface</u>
<u>x.serviceComplex1Impl</u> <POJO_ContributionE>	<u>1 service with ServiceComplex1 interface</u> <u>- operation1 checks the classloaders of the class, the interface class and the complex Java class</u> <u>- operation2 returns the hashCode of the class-loader of the class</u>
<u>service1CompositImpl.java</u>	<u>1 service with Service1 interface</u> <u>The implementation is COMPOSITE scope</u> <u>- it holds state information from one invocation of Service1.operation1 to the next invocation</u> <u>- the input String from one invocation is returned in the result of the next invocation</u>
<u>x.service1ContributionImpl.java</u> <POJO_ContributionD>	<u>1 service with Service1 interface</u> <u>- operation1 responds with a string ending in the contribution name</u>
<u>service1ContributionImpl.java</u>	<u>1 service with Service1 interface</u>

<POJO_ContributionC>	- operation1 responds with a string ending in the contribution name
service1ContributionImpl.java	1 service with Service1 interface
<POJO_ContributionB>	- operation1 responds with a string ending in the contribution name
service1ContributionImpl.java	1 service with Service1 interface
<POJO_9004>	- operation1 responds with a string ending in the name = "ContributionA"
service1CoordinatorImpl.java	1 service with Service1 interface 1 reference (1..n) with Service1 interface all configured wires get called when service1 operation1 is invoked This implementation acts as a coordinator between each of the invocations - it sends different data as input to each reference invocation - it records all the output data from all the invocations
x.service1Impl.java	1 service with Service1 interface
<POJO_9003>	
<POJO_General>	
service1Impl.java	1 service with Service1 interface
service1Impl3.java	1 service with Service1 interface 1 reference (1..n) with Service1 interface all configured wires get called when service1 operation1 is invoked
service1Imp6l.java	1 service with Service1 interface 1 reference (0..1) with Service1 interface configured wire gets called when service1 operation1 is invoked, if present otherwise, this implementation reports its absence
service1DupPropertySettersImpl.java	1 service with Service1 interface
<POJO_8013>	has a 2 setter methods annotated with @Property and the Javabeans names of these two methods is the same
service1DupReferenceSettersImpl.java	1 service with Service1 interface
<POJO_8014>	has a 2 setter methods annotated with @Reference and the Javabeans names of these two methods is the same

Service1JWSSOAPBindingImpl.java	1 service with Service1 interface class is annotated with @SOAPBinding
Service1JWSWebParamImpl.java	1 service with Service1 interface method is annotated with @WebParam
Service1JWSWebResultImpl.java	1 service with Service1 interface class is annotated with @WebResult
Service1JWSWebServiceEPIImpl.java	1 service with Service1 interface class is annotated with @WebService with the EndPointInterface attribute set
Service1JWSWebServiceImpl.java	1 service with Service1 interface class is annotated with @WebService
Service1JWSWebServiceProviderImpl.java	1 service with Service1 interface class is annotated with @WebServiceProvider
Service1JWSWebServiceProviderWSDLImpl.java	1 service with Service1 interface class is annotated with @WebServiceProvider with the wsdlLocation attribute set
Service1JWSWebServiceWSDLImpl.java	1 service with Service1 interface class is annotated with @WebService with the wsdlLocation attribute set
Service1MultipleAnnotatedConstructors.java <POJO_5007>	1 service with Service1 interface multiple constructors annotated with @Constructor and all parameters annotated with either @Property or @Reference
service1MultipleConstructorAnnotations.java <POJO_5006>	1 service with Service1 interface multiple constructors annotated with @Constructor
service1NoArgConstructorImpl.java	1 service with Service1 interface has a no argument constructor
Service1NoRemotableImpl.java	1 service with Service1 interface class is annotated with @Remotable
Service1NoRemotableRefImpl.java	1 service with Service1 interface 1 reference with Service1NoRemotable interface reference setter is annotated with @Remotable
service1NoServiceAnnotationImpl.java	1 service with Service1 interface no @Service annotation implements a @Remotable interface
service1NoValidConstructor.java	1 service with Service1 interface

<POJO_5005>	only constructor is not invokable by SCA
service1PrivateConstructor.java	1 service with Service1 interface
<POJO_5001>	only constructor is private
service1PropertySetterImpl.java	1 service with Service1 interface setter is annotated with @Property
Service1ReferenceConstructorImpl.java	1 service with Service1 interface 1 reference with Service1 interface constructor parameter annotated with @Reference
Service1ReferencePolicySetsImpl.java	1 service with Service1 interface 1 reference with Service1 interface reference is annotated with @PolicySets
Service1ReferenceSetterImpl.java	1 service with Service1 interface 1 reference with Service1 interface reference is set by a setter method
Service1RefWithNameImpl.java	1 service with Service1 interface 1 reference with Service1 interface @Reference has name value
Service1ServicePolicySetsImpl.java	1 service with Service1 interface class is annotated with @PolicySets
service1SupersetImpl.java	1 service with Service1Superset interface
Service1UnannotatedManyPropertySetterImpl.java	1 service with Service1 interface No @Service, @Property or @Reference annotations but a setter with a collection type that that does not have a @Remotable annotation
Service1UnannotatedMultiRefCollSetterImpl.java	1 service with Service1 interface 1 reference (1..n) with Service1 interface No @Service, @Property or @Reference annotations but a setter with a collection type that has a @Remotable annotation
Service1UnannotatedMultiRefSetterImpl.java	1 service with Service1 interface 1 reference (1..n) with Service1 interface No @Service, @Property or @Reference annotations but a setter with an array type that has a @Remotable annotation
service1UnannotatedPropertySetterImpl.java	1 service with Service1 interface No @Service, @Property or @Reference annotations but a setter with a type that that does not

	have a @Remotable annotation
service1UnannotatedRefSetterImpl.java	1 service with Service1 interface 1 reference with Service1 interface No @Service , @Property or @Reference annotations but a setter with a type that has a @Remotable annotation
Service1UnannotatedWithPolicySetsImpl.java	1 service with Service1 interface 1 reference with Service1 interface No @Service , @Property or @Reference annotations but a setter with a type that has a @Remotable annotation and class has a @PolicySets annotation
Service1UnannotatedWithRefPolicySetsImpl.java	1 service with Service1 interface 1 reference with Service1 interface No @Service , @Property or @Reference annotations but a public field with a type that has a @Remotable annotation and class has a @PolicySets annotation
Service1WithImplPolicySetsImpl.java	1 service with Service1 interface class is annotated with @PolicySets
Service1WithIntentImpl.java <POJO_8003>	1 service with Service1 interface class @Requires PropagatesTransaction
Service1WithIntentImpl.java <POJO_8012>	1 service with Service1 interface class @Requires managedTransaction
Service1WithRefIntentImpl.java <POJO_8008>	1 service with Service1 interface 1 reference with Service1 interface reference @Requires PropagatesTransaction
serviceImplClass.java	1 service with interface defined by the class itself that is compatible with Service 1 interface
z.ComplexClass.java <POJO_ContributionF>	ComplexClass class - used in a service operation with a Java class as its input parameter

337

3.3 WSDL Interface Files

Testcase ID	JCA_7004_TestCase
Test Assertion	JCA-TA-7003
Description	Tests that where a Java interface class is an asynchronous service mapping of a WSDL portType containing request/response operations that the class is annotated with the "asynchronous" intent – this is the negative version of JCA_7003_Testcase where the asynchronous intent is left off the interface class

Artifacts	JCA_7004_TestCase.java Test_JCA_7004.composite TestInvocation.wsdl TestClient_0002.composite Service1AsyncServerError.java Service1AsyncServerErrorImpl.java
Expected output	"exception"

338

Name	Description
Service1.wsdl	Service1 interface

339

JCA_7005_TestCase

340

341 | **4 Conformance**

342 | The artifacts contained in the sca-j-pojo-ci-1.1-testcases-package.zip file are considered to be authoritative and take precedence over the artifacts described in this document.

344 | An implementation that claims to conform to this specification MUST be able to run all test cases in Section 2 TestCases, producing the 'Expected Output'.

Testcase-ID	JCA_7005_TestCase
Test Assertion	JCA-TA-7006
Description	Tests that when an asynchronous service implementation invokes the sendResponse method of the supplied ResponseDispatch object when that method has already been called once, the method throws an IllegalStateException
Artifacts	JCA_7005_TestCase.java Test_JCA_7005.composite TestInvocation.wsdl TestClient_0002.composite Service1AsyncServer.java Service1AsyncServerMultiReponseImpl.java asyncControllerImpl.java dataStoreCompositeImpl.java
Expected-output	"JCA_7005 request service1 operation1 invoked asynchronously IllegalStateException thrown on second invocation of ResponseDispatch.sendResponse() method"

346 |

347 **Appendix A Test Assertions for the SCA POJO**
 348 **Component Implementation Specification**

349 This document defines the Test Assertions for the SCA POJO Component Implementation Specification
 350 Version 1.1 [JAVACI].

351 The test assertions in this document follow the format defined in the OASIS Test Assertion Guidelines
 352 specification [TA-GUIDE].

353 **Appendix A.1 Example Test Assertion**

354 Test assertions are presented in a tabular format with rows corresponding to the entry types defined in
 355 [TA-GUIDE].

356 **JCA_7006_TestCase**

357

Assertion ID	JCI-TA-xxxx
Source	[JCIx00yy]
Target	<kitchenSink/> element of composite file
Prerequisites	The <kitchenSink/> element has a @drain attribute
Predicate	The @drain attribute value of the <kitchenSink/> element is a URI that identifies a portal into the drainage system of the Domain.
Prescription Level	Mandatory
Tags	kitchenSink drain Domain
Testcase ID	JCA_7006_TestCase
Test Assertion	JCA-TA-9068 JCA-TA-9069
Description	Tests that when an asynchronous service implementation invokes the sendFault method of the supplied ResponseDispatch object that the exception is sent to the client of the service operation and if the sendFault method is subsequently invoked a second time when that method has already been called once for a given ResponseDispatch object, the method throws an IllegalStateException
Artifacts	JCA_7006_TestCase.java Test_JCA_7006.composite TestInvocation.wsdl TestClient_0002.composite Service1AsyncServer.java Service1AsyncServerMultiFaultImpl.java asyncControllerImpl.java dataStoreCompositeImpl.java
Expected output	"JCA_7005-request service1-operation1 invoked asynchronously IllegalStateException thrown on second invocation of ResponseDispatch.sendResponse() method"

358

359 **Assertion ID:** Is a unique ID for the test assertion. Its format starts with a 3 letter string that identifies the
 360 specification to which it relates - "JCI" is for the SCA POJO Component Implementation specification.

361 | This is followed by "-TA-" to indicate that this identifier is for a test assertion. This is then followed by a
 362 | unique 4 digit number.

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

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

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

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

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

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

374 | **Appendix A.2 Test Assertions for SCA POJO Component**
 375 | **Implementation Specification Section 2**
 376 |
 377 |

<u>Assertion ID</u>	<u>JCI-TA-2001</u>
<u>Source</u>	<u>[JCI20001]</u>
<u>Target</u>	<u>Introspected service interface of an SCA POJO component implementation</u>
<u>Prerequisites</u>	
<u>Predicate</u>	<u>The introspected interface is one of:</u> <ul style="list-style-type: none"> • <u>Java Interface</u> • <u>Java Class</u> • <u>Java Interface generated from a WSDL document</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"interface"</u>

378 | **4.1 Section 8**

379 | **JCA_8001_TestCase**
 380 |

<u>Testcase ID</u>	<u>JCA_8001_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-8001</u>
<u>Description</u>	<u>Tests that intent annotations use the @Intent annotation in the definition of the annotation. In this testcase, two mutually exclusive intents are defined in the domain. One of them is also defined as a Java annotation using the @Intent definition. TEST_JCA_8001Component1 requires both mutually exclusive intents (one in the SCDL, one in the Java implementation class) and should flag an error that mutually exclusive intents are not allowed.</u>

Artifacts	JCA_8001_TestCase.java Test_JCA_8001.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java Service1Intent.java TestIntent2.java definitions.xml
Expected-output	Negative-test: "exception"

381

Assertion ID	JCI-TA-2002
Source	[JCI20002]
Target	SCA POJO component implementation class
Prerequisites	SCA component provides a service interface
Predicate	The implementation class implements all the methods in the service interface.
Prescription Level	mandatory
Tags	"implementation" "interface"

382

JCA_8002_TestCase

383

384

Appendix A.3 Test Assertions for SCA POJO Component Implementation Specification Section 4

385

Testcase ID	JCA_8002_TestCase
Test Assertion	JCA-TA-8002
Description	Tests that intent annotations cannot be used on methods which are not reference setter methods.
Artifacts	JCA_8002_TestCase.java Test_JCA_8002.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java TestIntent2.java service1BadIntent.java
Expected-output	Negative-test: "exception"

386

Assertion ID	JCI-TA-4001
Source	[JCI40001]
Target	SCA Runtime
Prerequisites	The implementation class with an unannotated field that is introspected as

	a property and the Java type of the field is a JAXB annotated class
Predicate	The SCA Runtime converts the property value into an instance of the property's Java type as defined by the XML to Java mappings in the JAXB specification [JAX-B] with XML schema validation enabled.
Prescription Level	mandatory
Tags	"implementation" "property"

387 **JCA_8003_TestCase**

388

Testcase ID	JCA_8003_TestCase
Test Assertion	JCA-TA-8003
Description	Tests that intent annotations cannot be used on fields which are not references.
Artifacts	JCA_8003_TestCase.java Test_JCA_8003.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java TestIntent2.java service1BadIntent.java
Expected output	Negative test: "exception"

389

Assertion ID	JCI-TA-4002
Source	[JCI40001]
Target	SCA Runtime
Prerequisites	The implementation class with an unannotated setter method that is introspected as a property and the Java type of the field is a JAXB annotated class
Predicate	The SCA Runtime converts the property value into an instance of the property's Java type as defined by the XML to Java mappings in the JAXB specification [JAX-B] with XML schema validation enabled.
Prescription Level	mandatory
Tags	"implementation" "property"

390 **JCA_8004_TestCase**

391

392 **Appendix A.4 Test Assertions for SCA POJO Component**
393 **Implementation Specification Section 5**

Testcase ID	JCA_8004_TestCase
Test Assertion	JCA-TA-8004
Description	Tests that intent annotations cannot be used on constructor parameters-

	which are not references.
Artifacts	JCA_8004_TestCase.java Test_JCA_8004.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java TestIntent2.java service1BadIntent.java
Expected-output	Negative test: "exception"

394

<u>Assertion ID</u>	JCI-TA-5001
<u>Source</u>	[JCI50001]
<u>Target</u>	SCA POJO component implementation class
<u>Prerequisites</u>	
<u>Predicate</u>	The implementation class has a public or protected constructor.
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"implementation" "constructor"

395

JCA_8005_TestCase

396

Testcase ID	JCA_8005_TestCase
Test Assertion	JCA-TA-8002, JCA-TA-8003, JCA-TA-8004
Description	Tests that intent annotations can be used to annotate every place that a reference is allowed (setter method, field, or constructor parameter).
Artifacts	JCA_8005_TestCase.java Test_JCA_8005.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java TestIntent2.java service1GoodIntent.java
Expected-output	Positive test: "JCA_8005-request service1-operation1-invoked service2-operation1-invoked service3-operation1-invoked service4-operation1-invoked"

397

<u>Assertion ID</u>	JCI-TA-5002
<u>Source</u>	[JCI50004]
<u>Target</u>	SCA POJO component implementation class constructor
<u>Prerequisites</u>	a) SCA Java component implementation class has a constructor annotated with @Constructor b) SCA Java component implementation class has a constructor which is not annotated with @Constructor but where all its parameters are annot-

	<p>ated with either <u>@Property</u> or with <u>@Reference</u></p> <p>c) SCA POJO implementation class has a no-arg constructor which is not annotated with <u>@Constructor</u></p> <p>d) SCA POJO implementation has other constructors with arguments which are not annotated with <u>@Constructor</u> and which do not have all parameters marked with either <u>@Reference</u> or <u>@Property</u></p> <p>e) An SCA component using the implementation class is instantiated</p>
<u>Predicate</u>	The SCA Runtime invokes the constructor annotated with <u>@Constructor</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	"implementation" "constructor"

398 | JCA_8006_TestCase

399 |

<u>Testcase ID</u>	JCA_8006_TestCase
<u>Test Assertion</u>	JCA-TA-8005
<u>Description</u>	Tests that intent annotations are merged together (unioned) when present on the same Java element. In this testcase, two mutually exclusive intents are placed on the same Java element. When the runtime correctly merges them, an error should result.
<u>Artifacts</u>	JCA_8006_TestCase.java Test_JCA_8006.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java TestIntent1.java TestIntent2.java service1BadIntent.java definitions.xml
<u>Expected output</u>	Negative test: "exception"

400 |

<u>Assertion ID</u>	JCI-TA-5003
<u>Source</u>	[JCI50004]
<u>Target</u>	SCA POJO component implementation class constructor
<u>Prerequisites</u>	<p>a) SCA Java component implementation class does not have a constructor annotated with <u>@Constructor</u></p> <p>b) SCA Java component implementation class has a constructor which is not annotated with <u>@Constructor</u> but where all its parameters are annotated with either <u>@Property</u> or with <u>@Reference</u></p> <p>c) SCA POJO implementation class has a no-arg constructor which is not annotated with <u>@Constructor</u></p> <p>d) SCA POJO implementation has other constructors with arguments which are not annotated with <u>@Constructor</u> and which do not have all parameters marked with either <u>@Reference</u> or <u>@Property</u></p>

	<u>e) An SCA component using the implementation class is instantiated</u>
<u>Predicate</u>	<u>The SCA Runtime invokes the constructor which is not marked with @Constructor but which has all of its parameters marked with @Property or with @Reference</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"implementation" "constructor"</u>

401 **JCA_8007_TestCase**

402

<u>Testcase ID</u>	<u>JCA_8007_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-8006</u>
<u>Description</u>	<u>Tests that intent annotations are correctly merged when they appear at different levels in a Java interface class. PolicySet1 satisfies testIntent3 which is the only intent that should be applicable to the service after the intents in Service5Intents are normalized.</u>
<u>Artifacts</u>	<u>JCA_8007_TestCase.java Test_JCA_8007.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java TestIntent3.java TestIntent4.java service5Impl2.java service5Impl.java Service5Intents.java definitions.xml</u>
<u>Expected output</u>	<u>Positive test: "JCA_8007 request service1 operation1 invoked service2 operation1 invoked"</u>

403

<u>Assertion ID</u>	<u>JCI-TA-5004</u>
<u>Source</u>	<u>[JCI50004]</u>
<u>Target</u>	<u>SCA POJO component implementation class constructor</u>
<u>Prerequisites</u>	<u>a) SCA Java component implementation class does not have a constructor annotated with @Constructor</u> <u>b) SCA Java component implementation class does not have a constructor which is not annotated with @Constructor but where all its parameters are annotated with either @Property or with @Reference</u> <u>c) SCA POJO implementation class has a no-arg constructor which is not annotated with @Constructor</u> <u>d) SCA POJO implementation has other constructors with arguments which are not annotated with @Constructor and which do not have all parameters marked with either @Reference or @Property</u> <u>e) An SCA component using the implementation class is instantiated</u>
<u>Predicate</u>	<u>The SCA Runtime invokes the no-arg constructor</u>

<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"implementation" "constructor"</u>

404 **JCA_8008_TestCase**

405

<u>Testcase ID</u>	<u>JCA_8008_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-8007</u>
<u>Description</u>	<u>Tests that policySets cannot be used on methods which are not reference-setter methods:</u>
<u>Artifacts</u>	<u>JCA_8008_TestCase.java Test_JCA_8008.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1BadPolicySet.java definitions.xml</u>
<u>Expected output</u>	<u>Negative test: "exception"</u>

406

<u>Assertion ID</u>	<u>JCI-TA-5005</u>
<u>Source</u>	<u>[JCI50004]</u>
<u>Target</u>	<u>SCA POJO component implementation class constructor</u>
<u>Prerequisites</u>	<u>a) SCA Java component implementation class does not have a constructor annotated with @Constructor b) SCA Java component implementation class does not have a constructor which is not annotated with @Constructor but where all its parameters are annotated with either @Property or with @Reference c) SCA POJO implementation class does not have a no-arg constructor d) SCA POJO implementation has other constructors with arguments which are not annotated with @Constructor and which do not have all parameters marked with either @Reference or @Property e) An SCA component using the implementation class is instantiated</u>
<u>Predicate</u>	<u>The SCA Runtime does not invoke any constructor and an error is thrown</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"implementation" "constructor"</u>

407 **JCA_8009_TestCase**

408

<u>Testcase ID</u>	<u>JCA_8009_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-8008</u>

<u>Description</u>	Tests that policySets cannot be used on fields which are not references.
<u>Artifacts</u>	JCA_8009_TestCase.java Test_JCA_8009.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1BadPolicySet.java definitions.xml
<u>Expected-output</u>	Negative test: "exception"

409

<u>Assertion ID</u>	JCI-TA-5006
<u>Source</u>	[JCI50002]
<u>Target</u>	SCA POJO component implementation class constructor
<u>Prerequisites</u>	
<u>Predicate</u>	At most one constructor is annotated with the <u>@Constructor</u> annotation
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"implementation" "constructor"

410

<u>Testcase ID</u>	JCA_8010_TestCase
<u>Test Assertion</u>	JCA-TA-8009
<u>Description</u>	Tests that policySets cannot be used on constructor parameters which are not references.
<u>Artifacts</u>	JCA_8010_TestCase.java Test_JCA_8010.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service1BadPolicySet.java definitions.xml
<u>Expected-output</u>	Negative test: "exception"

411

<u>Assertion ID</u>	JCI-TA-5007
<u>Source</u>	[JCI50005]
<u>Target</u>	SCA POJO component implementation class constructor
<u>Prerequisites</u>	a) The implementation class has no <u>@Constructor</u> annotations
<u>Predicate</u>	The implementation class has at most one constructor that has a non-empty parameter list and all parameters are annotated with either <u>@Property</u> or <u>@Reference</u>
<u>Prescription Level</u>	mandatory

Tags	"implementation" "constructor"
------	--------------------------------

412 **JCA_8011_TestCase**

413

414 **Appendix A.5 Test Assertions for SCA POJO Component**
 415 **Implementation Specification Section 6**

Testcase ID	JCA_8011_TestCase
Test Assertion	JCA-TA-8010
Description	Tests that policySet annotations are correctly merged when they appear at different levels in a Java interface class. PolicySet1 satisfies testIntent3- and PolicySet2 satisfies testIntent5-
Artifacts	JCA_8010_TestCase.java Test_JCA_8010.composite TestInvocation.wsdl TestClient_0002.composite ASM_0002_Client.java Service1.java service6Impl.java service6Impl2.java Service6PolicySets.java TestIntent3.java TestIntent4.java definitions.xml
Expected output	Positive test: "JCA_8011 request service1-operation1 invoked service2-operation1 in- voked"

416

Assertion ID	JCI-TA-6001
Source	[JCI60001]
Target	SCA POJO implementation annotated with @Scope("STATELESS")
Prerequisites	
Predicate	POJO implementation is run and has the operational characteristics of a STATELESS scoped implementation
Prescription Level	mandatory
Tags	"implementation" "scope"

417

418 **4.2 Section 9**

419 **JCA_9001_TestCase**

420

Assertion ID	JCI-TA-6002
Source	[JCI60001]

Target	SCA POJO implementation with @Scope("COMPOSITE")
Prerequisites	
Predicate	POJO implementation is run and has the operational characteristics of a COMPOSITE scoped implementation
Prescription Level	mandatory
Tags	"implementation" "scope"
Testcase ID	JCA_9001_TestCase
Test Assertion	JCA-TA-9001
Description	Tests that if an invocation of the getService() method of the Component-Context API is made for a reference that is 0..n or 1..n multiplicity, that the caller receives an IllegalArgumentException
Artifacts	JCA_9001_TestCase.java Test_JCA_9001.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1ContextImpl1.java service1Impl.java
Expected output	Positive test: "JCA_9001 request invokerService-operation1 invoked IllegalArgumentException received"

421

422 **Appendix A.6 Test Assertions for SCA POJO Component**
423 **Implementation Specification Section 8**

424 **JCA_9002_TestCase**

425

Assertion ID	JCI-TA-8001
Source	[JCI80001]
Target	SCA POJO implementation class with a @Service annotation
Prerequisites	@Service annotation has a single interface class in its value attribute and no name attribute
Predicate	Introspected <componentType> has a <service> element with @name equal to the simple name of the interface class in the @Service annotation
Prescription Level	mandatory
Tags	"componentType" "service"
Testcase ID	JCA_9002_TestCase
Test Assertion	JCA-TA-9002
Description	Tests that if an invocation of the getRequestContext() method of the ComponentContext API is made during the execution of a Java business method of a service operation on the same Java thread used to invoke the business method, then a non-null value is returned for the RequestContext

Artifacts	JCA_9002_TestCase.java Test_JCA_9002.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1ContextImpl1.java service1Impl.java
Expected output	Positive test: "JCA_9002 request service1-operation1 invoked RequestContext--service-Name: Service1"

426

427 **JCA_9003_TestCase**

428

Assertion ID	JCI-TA-8002
Source	[JCI80001]
Target	SCA POJO implementation class with a @Service annotation
Prerequisites	<u>@Service annotation has a single interface class in its value attribute and no name attribute</u> <u>The interface class contains the @Remotable annotation</u>
Predicate	<u>Introspected <componentType> has a <service> element with an <interface.java/> subelement with the @interface attribute set to the fully qualified name of the interface class in the value attribute of the @Service annotation</u>
Prescription Level	mandatory
Tags	"componentType" "service"
Testcase ID	JCA_9003_TestCase
Test Assertion	JCA-TA-9003
Description	Tests that if an invocation of the getServiceReference() method of the RequestContext API is made during the execution of a Java business method, then the method returns a ServiceReference object that represents the service that was invoked
Artifacts	JCA_9003_TestCase.java Test_JCA_9003.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1Superset.java service1RequestContextImpl2.java
Expected output	Positive test: "JCA_9003 request service1-operation1 invoked ServiceReference-obtained: service1-checkService-operation2-invoked"

429

430 **JCA_9004_TestCase**

431

Assertion ID	JCI-TA-8003
--------------	-------------

Source	[JCI80001]
Target	SCA POJO implementation class with a @Service annotation and a @Requires annotation with one or more intents declared
Prerequisites	
Predicate	Introspeted <componentType> has a <service> element with @requires attribute containing the set of intents in the @Requires annotation
Prescription Level	mandatory
Tags	"componentType" "service"
Testcase ID	JCA_9004_TestCase
Test Assertion	JCA-TA-9004
Description	Tests that if an invocation of the getServiceReference() method of the RequestContext API is made during the execution of a Java business method of a callback, then the method returns a ServiceReference object that represents the callback that was invoked
Artifacts	JCA_9004_TestCase.java Test_JCA_9004.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service3WithCallback.java Service3Callback.java service1CallbackContextImpl1.java service3Impl1.java
Expected output	Positive test: "JCA_9004 request serviceComposite1 operation1 invoked callbackService operation1 invoked serviceComposite1 callback1 invoked ServiceReference obtained: serviceComposite1 check callback1 invoked"

432

433 **JCA_9005_TestCase**

434

Assertion ID	JCI-TA-8004
Source	[JCI80001]
Target	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations
Prerequisites	Implementation class implements an Interface where the interface class is annotated with @Remotable
Predicate	Introspeted <componentType> has a <service> element with a) @name set to the simple name of the interface class b) an <interface.java/> subelement with @interface attribute set to the fully qualified name of the interface class
Prescription Level	mandatory

Tags	"componentType" "service"
Testcase ID	JCA_9005_TestCase
Test Assertion	JCA-TA-9005
Description	Tests that if an invocation of the getRequestContext() method of the ComponentContext API is made during the execution of a Java business method of a callback operation on the same Java thread used to invoke the business method, then a non-null value is returned for the RequestContext
Artifacts	JCA_9005_TestCase.java Test_JCA_9005.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service3WithCallback.java Service3Callback.java service1CallbackContextImpl2.java service3Impl1.java
Expected output	Positive test: "JCA_9005-request serviceComposite1-operation1-invoked callbackService-operation1-invoked-serviceComposite1-callback1-invoked-RequestContext-serviceName:reference1"

435

436 JCA_9006_TestCase

437

Assertion ID	JCI-TA-8005
Source	[JCI80001]
Target	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations
Prerequisites	Implementation class does not implement any interfaces where the interface class is annotated with @Remotable
Predicate	Introspected <componentType> has a <service> element with a) @name set to the simple name of the implementation class b) an <interface.java/> subelement with @interface attribute set to the fully qualified name of the implementation class
Prescription Level	mandatory
Tags	"componentType" "service"
Testcase ID	JCA_9006_TestCase
Test Assertion	JCA-TA-9070 JCA-TA-9006 JCA-TA-9007 JCA-TA-9008 JCA-TA-9014
Description	Tests invocations of the ComponentContext.getServiceReference() method o invoked with a referenceName parameter for a reference of multiplicity 0..n

	<ul style="list-style-type: none"> o invoked for a reference which does not have the interface of the type in the businessInterface parameter o invoked with a referenceName value that does not match the name of any of the references of the component o invoked for a reference with multiplicity 0..1 which has no target service configured o invoked for a valid reference with a target service configured
Artifacts	JCA_9006_TestCase.java Test_JCA_9006.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1Impl.java Service1CCgetServiceReferenceImpl.java
Expected output	"JCA_9006 request service1 operation1 invoked getServiceReference expected IllegalArgumentException received for 0..n ref expected IllegalArgumentException received for invalid businessInterface expected IllegalArgumentException received for invalid reference name expected null ServiceReference for unwired 0..1 reference expected non-null ServiceReference for wired 1..1 reference service2 operation1 invoked";

438

439 **JCA_9007_TestCase**

440

<u>Assertion ID</u>	JCI-TA-8006
<u>Source</u>	[JCI80001]
<u>Target</u>	SCA POJO implementation class with a @Reference annotation
<u>Prerequisites</u>	@Reference annotation has a @name parameter
<u>Predicate</u>	Introspected <componentType> has a <reference> element with @name attribute set to the value of the @name parameter of the @Reference annotation
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "reference"
<u>Testcase ID</u>	JCA_9007_TestCase
<u>Test Assertion</u>	JCA-TA-9009
<u>Description</u>	Tests invocations of the ComponentContext.getURI() method: <ul style="list-style-type: none"> o invocation returns a String containing the absolute URI of the component in the SCA Domain
Artifacts	JCA_9007_TestCase.java Test_JCA_9007.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CCgetURIImpl.java
Expected output	"JCA_9007 request service1 operation1 invoked getURI getURI() returned URI = TEST_JCA_9007Component1"

441

442 | **JCA_9008_TestCase**

443

Assertion ID	JCI-TA-8007
Source	[JCI80001]
Target	SCA POJO implementation class with a @Reference annotation
Prerequisites	1) @Reference annotation annotates a field which is an array type 2) @Reference annotation has @required=true
Predicate	Introspected <componentType> has a <reference> element with @multiplicity set to 1..n
Prescription Level	mandatory
Tags	"componentType" "reference"
Testcase ID	JCA_8_TestCase
Test Assertion	JCA-TA-9001 JCA-TA-9010 JCA-TA-9011 JCA-TA-9012 JCA-TA-9013
Description	Tests invocations of the ComponentContext.getService() method: o method throws an IllegalArgumentException if the referenceName reference has multiplicity 0..n o method returns a proxy object with the businessInterface for the referenceName, where that reference has a target service configured o method returns null for a 0..1 reference with no target service configured o method throws an IllegalArgumentException where the component does not have a reference with the name referenceName o method throws an IllegalArgumentException where the the referenceName reference does not have the interface defined in the businessInterface parameter
Artifacts	JCA_9008_TestCase.java Test_JCA_9008.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CGgetServiceImpl.java
Expected output	"JCA_9008 request service1 operation1 invoked getService expected IllegalArgumentException received for 0..n ref expected IllegalArgumentException received for invalid businessInterface expected IllegalArgumentException received for invalid reference name expected null Service for unwired 0..1 reference expected non-null Service for wired 1..1 reference service2 operation1 invoked"

444

445 | **JCA_9009_TestCase**

446

Assertion ID	JCI-TA-8008
Source	[JCI80001]

Target	SCA POJO implementation class with a @Reference annotation
Prerequisites	1) @Reference annotation has a @required=false 2) @Reference annotation annotates a field with an interface type
Predicate	Introspected <componentType> has a <reference> element with @multiplicity set to 0..1
Prescription Level	mandatory
Tags	"componentType" "reference"
Testcase ID	JCA_9009_TestCase
Test Assertion	JCA-TA-9015 JCA-TA-9016 JCA-TA-9017 JCA-TA-9018 JCA-TA-9019
Description	Tests invocations of the ComponentContext.getServices() method: o invocation returns a Collection containing a proxy object for each target service of the reference each implementing the interface supplied in the businessInterface parameter o invocation returns an empty Collection where the reference named by referenceName exists but has no target services configured o invocation throws an IllegalArgumentException when the named reference exists and has a multiplicity of 0..1 or 1..1 o invocation throws an IllegalArgumentException when the named reference does not exist o invocation throws an IllegalArgumentException when the named reference does not have an interface compatible with the interface in the businessInterface parameter
Artifacts	JCA_9009_TestCase.java Test_JCA_9009.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CCgetServicesImpl.java
Expected output	"JCA_9009 request service1 operation1 invoked getServices expected IllegalArgumentException received for 1..1 ref expected IllegalArgumentException received for invalid businessInterface expected IllegalArgumentException received for invalid reference name expected empty Collection for unwired 0..n reference expected non-empty Collection for wired 0..n reference service2 operation1 invoked service3 operation1 invoked"

447

448 **JCA_9010_TestCase**

449

Assertion ID	JCI-TA-8009
Source	[JCI80001]
Target	SCA POJO implementation class with a @Reference annotation
Prerequisites	@Reference annotation annotates a field which is also annotated with a @Requires annotation

Predicate	Introspected <componentType> has a <reference> element with @requires set to the value of the @Requires annotation
Prescription Level	mandatory
Tags	"componentType" "reference"
Testcase ID	JCA_9010_TestCase
Test Assertion	JCA-TA-9020 JCA-TA-9021 JCA-TA-9022 JCA-TA-9023 JCA-TA-9024
Description	Tests invocations of the ComponentContext.getServiceReferences() method: <ul style="list-style-type: none"> o invocation returns a Collection containing a ServiceReference object for each target service of the reference each implementing the interface supplied in the businessInterface parameter o invocation returns an empty Collection where the reference named by referenceName exists but has no target services configured o invocation throws an IllegalArgumentException when the named reference exists and has a multiplicity of 0..1 or 1..1 o invocation throws an IllegalArgumentException when the named reference does not exist o invocation throws an IllegalArgumentException when the named reference does not have an interface compatible with the interface in the businessInterface parameter
Artifacts	JCA_9010_TestCase.java Test_JCA_9010.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CCgetServiceReferencesImpl.java
Expected output	"JCA_9010 request service1 operation1 invoked getServiceReferences expected IllegalArgumentException received for 1..1 ref expected IllegalArgumentException received for invalid businessInterface expected IllegalArgumentException received for invalid reference name expected empty Collection for unwired 0..n reference expected non-empty Collection for wired 0..n reference service2 operation1 invoked service3 operation1 invoked"

450

451 JCA_9011_TestCase

452

Assertion ID	JCI-TA-8010
Source	[JCI80001]
Target	SCA POJO implementation class with a @Property annotation
Prerequisites	1) @Property annotation annotates a setter method 2) Setter method parameter has a type which is not an array or collection type
Predicate	Introspected <componentType> has a <property> element with a) @name set to the JavaBeans property name derived from the setter

	<p><u>method name</u></p> <p><u>b) @type set to the JAXB mapping of the type of the parameter of the setter method</u></p>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"componentType" "reference"</u>
<u>Testcase ID</u>	<u>JCA_9011_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-9025</u> <u>JCA-TA-9026</u>
<u>Description</u>	<p>Tests invocations of the <code>ComponentContext.createSelfReference(businessInterface)</code> method:</p> <ul style="list-style-type: none"> o invocation returns a <code>ServiceReference</code> typed by the interface of the <code>businessInterface</code> parameter for one of the services offered by the component that has that interface o method throws an <code>IllegalArgumentException</code> if the component has no service which has the interface specified by the <code>businessInterface</code> parameter
<u>Artifacts</u>	<p><u>JCA_9011_TestCase.java</u></p> <p><u>Test_JCA_9011.composite</u></p> <p><u>TestInvocation.wsdl</u></p> <p><u>TestClient_0002.composite</u></p> <p><u>Service1.java</u></p> <p><u>Service1CCcreateSelfReferenceImpl.java</u></p>
<u>Expected output</u>	<u>"JCA_9011 request service1 operation1 invoked createSelfReference expected IllegalArgumentException received for invalid businessInterface" expected non-null ServiceReference service1 operation2 invoked"</u>

453

454 **JCA_9012_TestCase**

455

<u>Assertion ID</u>	<u>JCI-TA-8011</u>
<u>Source</u>	<u>[JCI80001]</u>
<u>Target</u>	<u>SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations</u>
<u>Prerequisites</u>	<u>Class has a public setter method which is not part of a service interface and which has a parameter typed by an interface class which is annotated with @Remotable</u>
<u>Predicate</u>	<p><u>Inspected <componentType> has a <reference> element with</u></p> <ul style="list-style-type: none"> <u>a) @name set to the JavaBeans property name derived from the setter method name</u> <u>b) @multiplicity is 1..1</u> <u>c) <interface.java/> subelement referencing the fully qualified name of the interface class</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"componentType" "reference"</u>

Testcase ID	JCA_9012_TestCase
Test Assertion	JCA-TA-9027 JCA-TA-9028 JCA-TA-9029
Description	Tests invocations of the ComponentContext.createSelfReference(businessInterface, serviceName) method: o invocation returns a ServiceReference typed by the interface of the businessInterface parameter for the service offered by the component that has that interface and has the name specified by serviceName o method throws an IllegalArgumentException if the service named in the serviceName parameter does not have the interface specified by the businessInterface parameter o method throws an IllegalArgumentException if the component does not have a service with the name specified in the serviceName parameter
Artifacts	JCA_9012_TestCase.java Test_JCA_9012.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CCreateSelfReferenceBImpl.java
Expected output	"JCA_9012 request service1 operation1 invoked createSelfReference expected IllegalArgumentException received for invalid businessInterface expected IllegalArgumentException received for invalid service name expected non-null ServiceReference service1 operation2 invoked"

456

457 **JCA_9013_TestCase**

458

Assertion ID	JCI-TA-8012
Source	[JCI80001]
Target	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations
Prerequisites	Class has a public field or setter with a type which is not a Java interface, array or parameterized java.util.Collection
Predicate	Introspected <componentType> has a <property> element with: a) @name set to the name of the public field or setter method b) @type set to the JAXB mapping of the type of the field or setter c) @many set to "false":
Prescription Level	mandatory
Tags	"componentType" "property"
Testcase ID	JCA_9013_TestCase
Test Assertion	JCA-TA-9030 JCA-TA-9031 JCA-TA-9032 JCA-TA-9033
Description	Tests invocations of the ComponentContext.getProperty method: o invocation returns an object of the type specified by the type parameter

	<p>containing the value in the component configuration for the property named by the <code>propertyName</code> parameter</p> <ul style="list-style-type: none"> o method returns null for a property which has no value specified in the component configuration o method throws an <code>IllegalArgumentException</code> if the component does not have a property with the name specified in the <code>propertyName</code> parameter o method throws an <code>IllegalArgumentException</code> if the property named in the <code>propertyName</code> parameter does not have a type compatible with the supplied <code>type</code> parameter
Artifacts	<p>JCA_9013_TestCase.java Test_JCA_9013.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CCGetPropertyImpl.java</p>
Expected-output	<p>"JCA_9013-request service1-operation1-invoked getProperty expected-IllegalArgumentExcep-tion received for invalid property type" expected-IllegalArgumentExcep-tion received for invalid property name expected null-value for unconfigured optional property expected non-null property value for required property: 2.456"</p>

459

460 **JCA_9014_TestCase**

461

<u>Assertion ID</u>	<u>JCI-TA-8013</u>
<u>Source</u>	<u>[JCI80001]</u>
<u>Target</u>	<u>SCA POJO implementation class</u>
<u>Prerequisites</u>	<u>Implementation class has a @Requires attribute</u>
<u>Predicate</u>	<u>Introspected <componentType> has an <implementation.java/> subelement with @requires attribute present with a value equal to the value of the @Requires annotation</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"componentType" "implementation"</u>
<u>Testcase ID</u>	<u>JCA_9014_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-9034 JCA-TA-9035</u>
<u>Description</u>	<p>Tests invocations of the <code>ComponentContext.cast</code> method:</p> <ul style="list-style-type: none"> o invocation returns a <code>ServiceReference</code> object typed by the same interface as specified by the reference proxy object supplied in the target parameter o method throws an <code>IllegalArgumentException</code> if the supplied target parameter is not an SCA reference proxy object
Artifacts	<p>JCA_9014_TestCase.java Test_JCA_9014.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1CCcastImpl.java</p>
Expected-output	<p>"JCA_9014-request service1-operation1-invoked cast expected-IllegalArgumentExcep-tion received for invalid reference proxy expected non-null Ser-</p>

462

	viceReference ServiceReference has correct business interface"
--	--

463

JCA_9015_TestCase

464

Assertion ID	JCI-TA-8014
Source	[JCI80002]
Target	SCA POJO implementation class with 2 or more setter methods annotated with @Property
Prerequisites	
Predicate	The JavaBeans property name for each property setter method is unique.
Prescription Level	mandatory
Tags	"implementation" "property"
Testcase ID	JCA_5_TestCase
Test Assertion	JCA-TA-9039 JCA-TA-9040 JCA-TA-9041 JCA-TA-9042 JCA-TA-9043 JCA-TA-9044 JCA-TA-9045 JCA-TA-9046 JCA-TA-9047 JCA-TA-9048
Description	<p>Tests invocations of the RequestContext.method:</p> <ul style="list-style-type: none"> o getServiceName invocation returns the name of the service for which the operation is being processed when called from a thread processing a service operation o getServiceName method returns null when called from a thread not processing a service operation or callback operation o getCallbackReference returns a ServiceReference typed by the interface of the callback when invoked by a thread processing an operation of a bidirectional service o getCallbackReference returns null when invoked by a thread processing an operation of a service which is not bidirectional o getCallbackReference returns null when invoked by a thread not processing an operation of a service o getCallback returns a reference proxy typed by the interface of the callback when invoked by a thread processing an operation of a bidirectional service o getCallback returns null when invoked by a thread processing an operation of a service which is not bidirectional o getCallback returns null when invoked by a thread not processing an operation of a service o getServiceReference method returns a ServiceReference representing the callback service when invoked from a thread processing a callback operation o getServiceReference method returns null when called from a thread not involved in processing either a service operation or a callback operation
Artifacts	JCA_9015_TestCase.java

	Test_JCA_9015-composite TestInvocation.wsdl TestClient_0002-composite Service1.java Service1RCgetServiceNameImpl.java
Expected output	

465

466 **JCA_9016_TestCase**

467

Assertion ID	JCI-TA-8015
Source	[JCI80002]
Target	<u>SCA POJO implementation class with 2 or more setter methods annotated with @Reference</u>
Prerequisites	
Predicate	<u>The JavaBeans property name for each reference setter method is unique.</u>
Prescription Level	<u>mandatory</u>
Tags	<u>"implementation" "reference"</u>
Testcase ID	JCA_9016_TestCase
Test Assertion	JCA-TA-9051 JCA-TA-9052 JCA-TA-9053 JCA-TA-9054 JCA-TA-9055 JCA-TA-9056 JCA-TA-9057 JCA-TA-9058 JCA-TA-9059 JCA-TA-9060
Description	Tests invocations of the SCAClientFactory API class: <ul style="list-style-type: none"> o invocation of newInstance(URI) with a valid URI value returns an object which implements the SCAClientFactory class for the SCA Domain identified by the domainURI parameter o invocation of newInstance(URI) with an invalid URI value causes a NoSuchDomainException to be thrown o invocation of newInstance(Properties, URI) with a valid URI value returns an object which implements the SCAClientFactory class for the SCA Domain identified by the domainURI parameter o invocation of newInstance(Properties, URI) with an invalid URI value causes a NoSuchDomainException to be thrown o invocation of newInstance(Classloader, URI) with a valid URI value returns an object which implements the SCAClientFactory class for the SCA Domain identified by the domainURI parameter o invocation of newInstance(Classloader, URI) with an invalid URI value causes a NoSuchDomainException to be thrown o invocation of newInstance(Properties, Classloader, URI) with a valid URI value returns an object which implements the SCAClientFactory class for the SCA Domain identified by the domainURI parameter o invocation of newInstance(Properties, Classloader, URI) with an invalid URI value causes a NoSuchDomainException to be thrown

	<ul style="list-style-type: none"> o invocation of getService with a serviceURI that identifies a service in the Domain which has the business interface identified bby the interfaze parameter returns a reference proxy object which can be used to invoke operations on the identified service o invocation of getService with a serviceURI for which there is no service in the domain throws a NoSuchServiceException
Artifacts	<p>JCA_9016_TestCase.java Test_JCA_9016.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1SCAClientImpl.java</p>
Expected-output	<p>"JCA_9016 request service1 operation1 invoked SCAClient expected non-null client factory expected exception received for invalid URI: NoSuchDomainException expected non-null client factory expected exception received for invalid URI: NoSuchDomainException expected non-null client factory expected exception received for invalid URI: NoSuchDomainException expected non-null client factory expected exception received for invalid URI: NoSuchDomainException expected non-null reference proxyservice2 operation1 invoked expected NoSuchServiceException exception received for getService()"</p>

468

469 **4.3 Section 10**

470 **JCA_10001_TestCase**

471

Assertion ID	JCI-TA-8016
Source	[JCI80001]
Target	SCA POJO implementation class with a @Service annotation and @Remotable annotation
Prerequisites	<p><u>@Service annotation has a single interface class in its value attribute and no name attribute</u></p> <p><u>The interface class does not contain the @Remotable annotation</u></p>
Predicate	<p>Introspected <componentType> has a <service> element with an <interface.java/> subelement where:</p> <ul style="list-style-type: none"> - <u>the @interface attribute set to the fully qualified name of the interface class in the value attribute of the @Service annotation</u> - <u>the @remotable attribute set to "true".</u>
Prescription Level	mandatory
Tags	"componentType" "service"
Testcase ID	JCA_10001_TestCase
Test Assertion	JCA-TA-10001
Description	Tests that Java annotations are not used in improper locations. This test case has an @Property annotation on a method parameter where the method is not a constructor.
Artifacts	JCA_10001_TestCase.java

	Test_JCA_10001.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl service1BadAnnotation.java Service1.java
Expected output	

472

473 **JCA_10002_TestCase**

474

Assertion ID	JCI-TA-8017
Source	[JCI80001]
Target	SCA POJO implementation class with a @Reference annotation
Prerequisites	<ol style="list-style-type: none"> 1) @Reference annotation annotates a field, setter or constructor parameter with an interface type 2) Interface type has no @Remotable annotation 3) @Remotable annotation on the same field, setter or constructor parameter as the @Reference annotation
Predicate	<p>Introspected <componentType> has a <reference> element with @multiplicity set to 0..1 and a child <interface.java> element with</p> <ul style="list-style-type: none"> - @interface attribute set to the fully qualified name of the interface type - @remotable attribute set to true
Prescription Level	mandatory
Tags	"componentType" "reference"
Testcase ID	JCA_10002_TestCase
Test Assertion	JCA-TA-10002
Description	Tests that Java annotations are not used on static methods. This testcase has an @Property annotation on a static setter method.
Artifacts	JCA_10002_TestCase.java Test_JCA_10002.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl service1StaticAnnotation.java Service1.java
Expected output	

475

476 **JCA_10003_TestCase**

477

Assertion ID	JCI-TA-8018
Source	[JCI80001]

<u>Target</u>	<u>SCA POJO implementation class with a @Reference annotation</u>
<u>Prerequisites</u>	<u>@Reference annotation annotates a setter which takes an interface type parameter</u>
<u>Predicate</u>	<u>Introspected <componentType> has a <reference> element with a child <interface.java> element with @interface attribute set to the fully qualified name of the interface type</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"componentType" "reference"</u>
<u>Testcase ID</u>	<u>JCA_10003_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-10003</u>
<u>Description</u>	<u>Tests that Java annotations are not used on static fields. This testcase has an @Property annotation on a static field.</u>
<u>Artifacts</u>	<u>JCA_10003_TestCase.java Test_JCA_10003.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl service1StaticAnnotation.java Service1.java</u>
<u>Expected output</u>	

478

479 **JCA_10004_TestCase**

480

<u>Assertion ID</u>	<u>JCI-TA-8019</u>
<u>Source</u>	<u>[JCI80001]</u>
<u>Target</u>	<u>SCA POJO implementation class with a @Reference annotation</u>
<u>Prerequisites</u>	<u>@Reference annotation annotates a constructor parameter which is an interface type</u>
<u>Predicate</u>	<u>Introspected <componentType> has a <reference> element with a child <interface.java> element with @interface attribute set to the fully qualified name of the interface type</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"componentType" "reference"</u>
<u>Testcase ID</u>	<u>JCA_10004_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-10031</u>
<u>Description</u>	<u>Tests that the @Callback annotation has to be specified with any parameters when used as the injection point of a callback reference.</u>
<u>Artifacts</u>	<u>JCA_10004_TestCase.java Test_JCA_10004.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl</u>

	Service1.java service1CallbackImpl.java service3BadCallbackImpl.java Service3Callback.java Service3WithCallback.java
Expected output	

481

482 **JCA_10005_TestCase**

483

Assertion ID	JCI-TA-8020
Source	[JCI80001]
Target	SCA POJO implementation class with a @Service annotation and a @PolicySets annotation with one or more policySets declared
Prerequisites	
Predicate	Introspected <componentType> has a <service> element with @policySets attribute containing the set of policySets in the @PolicySets annotation
Prescription Level	mandatory
Tags	"componentType" "service"
Testcase ID	JCA_10005_TestCase
Test Assertion	JCA-TA-10005
Description	Tests that @Constructor works correctly with annotated parameters.
Artifacts	JCA_10005_TestCase.java Test_JCA_10005.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1ConstrImpl.java
Expected output	Positive:

484

485 **JCA_10006_TestCase**

486

Assertion ID	JCI-TA-8021
Source	[JCI80001]
Target	SCA POJO implementation class with a @Reference annotation and a @PolicySets annotation with one or more policySets declared
Prerequisites	
Predicate	Introspected <componentType> has a <reference> element with @policySets attribute containing the set of policySets in the @PolicySets annotation
Prescription	mandatory

Level	
Tags	<u>"componentType" "reference"</u>
Testcase ID	JCA_10006_TestCase
Test Assertion	JCA-TA-10005
Description	Tests that the runtime raises an error when an <u>@Constructor</u> annotated constructor does not have all of it's parameters annotated.
Artifacts	JCA_10006_TestCase.java Test_JCA_10006.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1BadConstImpl.java
Expected output	

487

488 **JCA_10007_TestCase**

489

Assertion ID	<u>JCI-TA-8022</u>
Source	<u>[JCI80001]</u>
Target	<u>SCA POJO implementation class with zero @Service annotations, zero @Reference annotations, zero @Property annotations, and a @PolicySets annotation on the class itself with one or more policySets declared</u>
Prerequisites	
Predicate	<u>Introspected <componentType> has a <service> element with @policySets attribute containing the set of policySets in the @PolicySets annotation</u>
Prescription Level	<u>mandatory</u>
Tags	<u>"componentType" "service"</u>
Testcase ID	JCA_10007_TestCase
Test Assertion	JCA-TA-10006
Description	Tests that the runtime raises an error when an <u>@Destroy</u> method has a non-void return and parameters.
Artifacts	JCA_10007_TestCase.java Test_JCA_10007.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1BadDestroyImpl.java
Expected output	

490

491 **JCA_10008_TestCase**

492

Assertion ID	<u>JCI-TA-8023</u>
--------------	--------------------

Source	[JCI80001]
Target	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations, zero @Property annotations, and a @PolicySets annotation on a public field, typed by an interface annotated with @Remotable, with one or more policySets declared
Prerequisites	
Predicate	Introspected <componentType> has a <reference> element with @policySets attribute containing the set of policySets in the @PolicySets annotation
Prescription Level	mandatory
Tags	"componentType" "reference"
Testcase ID	JCA_10008_TestCase
Test Assertion	JCA-TA-10009
Description	Tests that the runtime raises an error when an @Init method has a non-void return and parameters.
Artifacts	JCA_10008_TestCase.java Test_JCA_10008.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1BadInitImpl.java
Expected output	

493

494 **JCA_10009_TestCase**

495

Assertion ID	JCI-TA-8024
Source	[JCI80001]
Target	SCA POJO implementation class
Prerequisites	Implementation class has a @PolicySets attribute
Predicate	Introspected <componentType> has an <implementation.java/> subelement with @policySets attribute present with a value equal to the value of the @PolicySets annotation
Prescription Level	mandatory
Tags	"componentType" "implementation"
Testcase ID	JCA_10009_TestCase
Test Assertion	JCA-TA-10011
Description	Tests that the runtime raises an error when an @Property annotated field is marked as final.
Artifacts	JCA_10009_TestCase.java Test_JCA_10009.composite TestInvocation.wsdl TestClient_0002.composite

	Service1.wsdl Service1.java service1BadPropImpl.java
Expected output	

496

497 **JCA_10010_TestCase**

498

Assertion ID	JCI-TA-8025
Source	[JCI80001]
Target	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations
Prerequisites	Class has a public setter method which is not part of a service interface and which has a parameter typed by an array of interface classes which are annotated with @Remotable
Predicate	Introspected <componentType> has a <reference> element with a) @name set to the JavaBeans property name derived from the setter method name b) @multiplicity is 1..n c) <interface.java/> subelement referencing the fully qualified name of the interface class
Prescription Level	mandatory
Tags	"componentType" "reference"
Testcase ID	JCA_10010_TestCase
Test Assertion	JCA-TA-10012
Description	Tests that @Property specifies a name when used with @Constructor.
Artifacts	JCA_10010_TestCase.java Test_JCA_10010.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1ConstrBadPropImpl.java
Expected output	

499

500 **JCA_10011_TestCase**

501

Assertion ID	JCI-TA-8026
Source	[JCI80001]
Target	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations
Prerequisites	Class has a public setter method which is not part of a service interface

	and which has a parameter typed by a java.util.Collection which is parameterized by an interface class which is annotated with @Remotable
<u>Predicate</u>	<p>Introspected <componentType> has a <reference> element with</p> <p>a) @name set to the JavaBeans property name derived from the setter method name</p> <p>b) @multiplicity is 1..n</p> <p>c) <interface.java/> subelement referencing the fully qualified name of the interface class</p>
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "reference"
<u>Testcase ID</u>	JCA_10011_TestCase
<u>Test Assertion</u>	JCA-TA-10013
<u>Description</u>	Tests that @Property is not required=true when used with @Constructor.
<u>Artifacts</u>	<p>JCA_10011_TestCase.java</p> <p>Test_JCA_10011.composite</p> <p>TestInvocation.wsdl</p> <p>TestClient_0002.composite</p> <p>Service1.wsdl</p> <p>Service1.java</p> <p>service1ConstrBadPropImpl.java</p>
<u>Expected output</u>	

502

503 **JCA_10012_TestCase**

504

<u>Assertion ID</u>	JCI-TA-8027
<u>Source</u>	[JCI80001]
<u>Target</u>	SCA POJO implementation class with zero @Service annotations, zero @Reference annotations and zero @Property annotations
<u>Prerequisites</u>	Class has a public field or setter with a type which is not a Java interface, but is an array or parameterized (non-interface) java.util.Collection
<u>Predicate</u>	<p>Introspected <componentType> has a <property> element with:</p> <p>a) @name set to the name of the public field</p> <p>b) @type set to the JAXB mapping of the type of the field or setter</p> <p>c) @many set to "true"</p>
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "property"
<u>Testcase ID</u>	JCA_10012_TestCase
<u>Test Assertion</u>	JCA-TA-10014
<u>Description</u>	Tests that multi-valued properties are introspected correctly.
<u>Artifacts</u>	JCA_10012_TestCase.java

	Test_JCA_10012.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1MultiPropImpl.java
Expected-output	Positive:

505

506 **JCA_10013_TestCase**

507

Assertion ID	JCI-TA-8028
Source	[JCI80001]
Target	SCA POJO implementation class with javax.jws.WebService annotation
Prerequisites	
Predicate	<p>Inspected <componentType> has a <service> element with:</p> <p>a) @name set to the value of the name property on the @WebService annotation</p> <p>b) <interface.java> child element with the @interface attribute set to the fully qualified name of the implementation class</p>
Prescription Level	mandatory
Tags	"componentType" "service" "jax-ws"
Testcase ID	JCA_10013_TestCase
Test Assertion	JCA-TA-10016
Description	Tests that intent annotations can have qualifiers.
Artifacts	JCA_10013_TestCase.java Test_JCA_10013.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1IntentQualifier.java
Expected-output	Positive:

508

509 **JCA_10014_TestCase**

510

Assertion ID	JCI-TA-8029
Source	[JCI80001]
Target	SCA POJO implementation class with javax.jws.WebService annotation
Prerequisites	EndpointInterface annotation property is set to a Java interface class

<u>Predicate</u>	<p>Introspected <componentType> has a <service> element with:</p> <p>a) @name set to the value of the name property on the @WebService annotation</p> <p>b) <interface.java> child element with the @interface attribute set to the value of the endpointInterface property</p>
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "service" "jax-ws"
<u>Testcase ID</u>	JCA_10014_TestCase
<u>Test Assertion</u>	JCA-TA-10018
<u>Description</u>	Tests that @Reference has a name when used with @Constructor.
<u>Artifacts</u>	<p>JCA_10014_TestCase.java</p> <p>Test_JCA_10014.composite</p> <p>TestInvocation.wsdl</p> <p>TestClient_0002.composite</p> <p>Service1.wsdl</p> <p>Service1.java</p> <p>service1ConstrBadRefImpl.java</p> <p>service1Impl.java</p>
<u>Expected-output</u>	Negative:

511

512 **JCA_10015_TestCase**

513

<u>Assertion ID</u>	JCI-TA-8030
<u>Source</u>	[JCI80001]
<u>Target</u>	SCA POJO implementation class with javax.jws.WebService annotation
<u>Prerequisites</u>	WsdLocation property is set and points to a WSDL document
<u>Predicate</u>	<p>Introspected <componentType> has a <service> element with:</p> <p>a) @name set to the value of the name property on the @WebService annotation</p> <p>b) <interface.wsdl> child element with the @interface attribute set to the portType referenced by the value of the wsdlLocation property</p>
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "service" "jax-ws"
<u>Testcase ID</u>	JCA_10015_TestCase
<u>Test Assertion</u>	JCA-TA-10019
<u>Description</u>	Tests that @Reference has required=true when used with @Constructor.
<u>Artifacts</u>	<p>JCA_10015_TestCase.java</p> <p>Test_JCA_10015.composite</p> <p>TestInvocation.wsdl</p> <p>TestClient_0002.composite</p>

	Service1.wsdl Service1.java service1ConstrBadRefImpl.java service1Impl.java
Expected output	Negative:

514

515 **JCA_10016_TestCase**

516

<u>Assertion ID</u>	JCI-TA-8031
<u>Source</u>	[JCI80001]
<u>Target</u>	SCA POJO implementation class with javax.jws.WebParam annotation that has the header property set to "true"
<u>Prerequisites</u>	
<u>Predicate</u>	Introspected <componentType> has a <service> element with a @requires attribute set to "SOAP".
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "service" "jax-ws"
<u>Testcase ID</u>	JCA_10016_TestCase
<u>Test Assertion</u>	JCA-TA-10022
<u>Description</u>	Tests that @Reference(required=false) is introspected as multiplicity 0..n.
<u>Artifacts</u>	JCA_10016_TestCase.java Test_JCA_10016.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl Service1.java service1OptMultiRef.java
<u>Expected output</u>	Positive:

517

518

520 **JCA_10044_TestCase**

521

<u>Assertion ID</u>	JCI-TA-8032
<u>Source</u>	[JCI80001]
<u>Target</u>	SCA POJO implementation class with javax.jws.WebResult annotation that has the header property set to "true"
<u>Prerequisites</u>	
<u>Predicate</u>	Introspected <componentType> has a <service> element with a @requires attribute set to "SOAP".

<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "service" "jax-ws"
<u>Testcase ID</u>	JCA_10044_TestCase
<u>Test Assertion</u>	JCA-TA-10024
<u>Description</u>	Tests that where a component reference with multiplicity of 0..1 is not-wired, that the reference injected into the implementation is null
<u>Artifacts</u>	JCA_10044_TestCase.java Test_JCA_10044.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1Impl6.java
<u>Expected output</u>	Positive test: "JCA_10044 request service1 operation1 invoked reference is null"

522 | **JCA_10045_TestCase**

523 |

<u>Testcase ID</u>	JCA_10045_TestCase
<u>Test Assertion</u>	JCA-TA-10025
<u>Description</u>	Tests that where a component reference with multiplicity of 0..n is not-wired, that the reference injected into the implementation is an empty array or collection
<u>Artifacts</u>	JCA_10045_TestCase.java Test_JCA_10045.composite TestInvocation.wsdl TestClient_0002.composite Service1.java service1Impl4.java
<u>Expected output</u>	Positive test: "JCA_10045 request service1 operation1 invoked reference array is empty"

524 |

<u>Assertion ID</u>	JCI-TA-8033
<u>Source</u>	[JCI80001]
<u>Target</u>	SCA POJO implementation class with javax.jws.soap.SOAPBinding annotation
<u>Prerequisites</u>	
<u>Predicate</u>	Inspected <componentType> has a <service> element with a @requires attribute set to "SOAP".
<u>Prescription Level</u>	mandatory
<u>Tags</u>	"componentType" "service" "jax-ws"

525 | **JCA_10046_TestCase**

526 |

<u>Testcase ID</u>	JCA_10046_TestCase
<u>Test Assertion</u>	JCA-TA-10045

Description	Tests that where the Java interface of a service is marked remotable, the interface is translatable into a WSDL portType
Artifacts	JCA_10046_TestCase.java Test_JCA_10046.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Unmappable.java unmappableServiceImpl.java
Expected-output	Negative test: "exception"

527

Assertion ID	JCI-TA-8034
Source	[JCI80001]
Target	SCA POJO implementation class with javax.xml.ws.WebServiceProvider annotation
Prerequisites	
Predicate	Introspected <componentType> has a <service> element with an <interface.java> child element with the @interface attribute set to the fully qualified name of the implementation class
Prescription Level	mandatory
Tags	"componentType" "service" "jax-ws"

528

JCA_10047_TestCase

529

Testcase ID	JCA_10047_TestCase
Test Assertion	JCA-TA-10046
Description	Tests that the @Scope annotation is only valid when applied to a component implementation class
Artifacts	JCA_10047_TestCase.java Test_JCA_10047.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite- Service1.wsdl
Expected-output	Negative test: "exception"

530

Assertion ID	JCI-TA-8035
Source	[JCI80001]
Target	SCA POJO implementation class with javax.xml.ws.WebServiceProvider annotation
Prerequisites	WsdlLocation property is set and points to a WSDL document
Predicate	Introspected <componentType> has a <service> element with an <interface.wsdl> child element with the @interface attribute set to the portType

	referenced by the value of the wsdlLocation property
Prescription Level	mandatory
Tags	"componentType" "service" "jax-ws"

531 **JCA_10048_TestCase**

532

Testcase ID	JCA_10048_TestCase
Test Assertion	JCA-TA-10047
Description	Tests that where a class is annotated with a @Service annotation which declares a set of services, that the implementation class implements all of the methods of all the declared service interfaces
Artifacts	JCA_10048_TestCase.java Test_JCA_10048.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite-Service1.wsdl
Expected output	Negative test: "exception"

533

Assertion ID	JCI-TA-8036
Source	[JCI80001]
Target	SCA POJO implementation class with any JAX-WS annotation
Prerequisites	
Predicate	Introspected <componentType> has a <service> element with a <binding.ws> child element with the @wsdlElement attribute set to point to a WSDL document that uses the SOAP/HTTP binding (http://schemas.xmlsoap.org/wsdl/soap/http)
Prescription Level	mandatory
Tags	"componentType" "service" "jax-ws"

534 **JCA_10049_TestCase**

535

Testcase ID	JCA_10049_TestCase
Test Assertion	JCA-TA-10050
Description	Tests that where a @Service annotation has a @names attribute, that the number of entries in the array of @names is the same as the number of entries in the array of the @value attribute
Artifacts	JCA_10049_TestCase.java Test_JCA_10049.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite-Service1.wsdl
Expected output	Negative test:

	"exception"
536	
<u>Assertion ID</u>	<u>JCI-TA-8037</u>
<u>Source</u>	<u>[JCI80001]</u>
<u>Target</u>	<u>SCA POJO implementation class with javax.xml.ws.BindingType annotation</u>
<u>Prerequisites</u>	<u>The value of the @BindingType annotation is http://www.w3.org/2003/05/soap/bindings/HTTP/</u>
<u>Predicate</u>	<u>Introspected <componentType> has a <service> element with a <binding.ws> child element with the @requires attribute set to "SOAP.v1_2"</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"componentType" "service" "jax-ws"</u>

537 **JCA_10050_TestCase**

538

539 **Appendix A.7 Test Assertions for SCA POJO Component**
 540 **Implementation Specification Section 9**

<u>Testcase ID</u>	<u>JCA_10050_TestCase</u>
<u>Test Assertion</u>	<u>JCA-TA-10055</u>
<u>Description</u>	<u>Tests that where an implementation class declares 2 or more services, that each service interface class has a distinct Java simple name</u>
<u>Artifacts</u>	<u>JCA_10050_TestCase.java Test_JCA_10050.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite-Service1.wsdl</u>
<u>Expected output</u>	<u>Negative test: "exception"</u>

541

<u>Assertion ID</u>	<u>JCI-TA-9001</u>
<u>Source</u>	<u>[JCI90001]</u>
<u>Target</u>	<u>SCA <component> using <implementation.java/></u>
<u>Prerequisites</u>	
<u>Predicate</u>	<u>The <implementation.java/> element conforms to the sca-implementation-java.xsd schema</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"implementation" "schema"</u>

542 **JCA_10051_TestCase**

543

Testcase ID	JCA_10051_TestCase
Test Assertion	JCA-TA-10057
Description	Tests that a @Service annotation has at least one element in its @value-array
Artifacts	JCA_10051_TestCase.java Test_JCA_10051.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite- Service1.wsdl
Expected output	Negative test: "exception"

544

Assertion ID	JCI-TA-9002
Source	[JCI90002, JCI100008]
Target	SCA POJO component implementation class
Prerequisites	a) An SCA component that uses the @class attribute to specify its Java implementation class b) The Java implementation class is contained within the same contribution as the component c) The SCA component is instantiated
Predicate	The Java class specified on the @class attribute is found
Prescription Level	mandatory
Tags	"implementation" "resolution"

545 **JCA_10052_TestCase**

546

Testcase ID	JCA_10052_TestCase
Test Assertion	JCA-TA-10058
Description	Tests that all the entries in the @names array of a @Service annotation are unique
Artifacts	JCA_10052_TestCase.java Test_JCA_10052.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite- Service1.wsdl
Expected output	Negative test: "exception"

547

Assertion ID	JCI-TA-9003
Source	[JCI90002, JCI100008]

<u>Target</u>	<u>SCA POJO component implementation class</u>
<u>Prerequisites</u>	<u>a) An SCA component that uses the @class attribute to specify its Java implementation class</u> <u>b) The Java implementation class is exported by a contribution (Contribution B) in the SCA Domain</u> <u>c) The Java implementation class is also contained within the same contribution (Contribution A) as the component</u> <u>d) Contribution A imports the Java package of the implementation class</u> <u>e) The SCA component is instantiated</u>
<u>Predicate</u>	<u>The Java class specified on the @class attribute is loaded from Contribution B</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"implementation" "resolution"</u>

548

549 **4.4 Section 11**

550 **JCA_11001_TestCase**

551

<u>Assertion ID</u>	<u>JCI-TA-9004</u>
<u>Source</u>	<u>[JCI90002, JCI100008]</u>
<u>Target</u>	<u>SCA POJO component implementation class</u>
<u>Prerequisites</u>	<u>a) An SCA component that uses the @class attribute to specify its Java implementation class</u> <u>b) The Java implementation class is contained within another contribution (Contribution C) in the SCA Domain</u> <u>c) The Java implementation class is exported by another contribution (Contribution B) in the SCA Domain</u> <u>d) The Java implementation class is also contained within the same contribution (Contribution A) as the component</u> <u>e) Contribution A imports the Java package of the implementation class and uses the @location attribute of <import.java/> to specify Contribution C</u> <u>f) The SCA component is instantiated</u>
<u>Predicate</u>	<u>The Java class specified on the @class attribute is loaded from Contribution C</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"implementation" "resolution"</u>
<u>Testcase ID</u>	<u>JCA_11001_TestCase</u>

Test Assertion	JCA-TA-11001
Description	Tests that a Java interface which does not have <code>@WebService</code> annotation which is mapped to WSDL by an SCA runtime is treated as if it did contain a <code>@WebService</code> annotation
Artifacts	JCA_11001_TestCase.java Test_JCA_11001.composite TestInvocation.wsdl TestClient_0002.composite TestComposite1.composite- Service1.java Service1ImplWSService.wsdl service1Impl.java service1Impl2.java
Expected output	Positive test: "JCA_11001 request service1 operation1 invoked service2 operation1 invoked"

552

553 **JCA_11002_TestCase**

554

Assertion ID	JCI-TA-9005
Source	[JCI90002, JCI100008]
Target	SCA POJO component implementation class
Prerequisites	<p>a) An SCA component that uses the <code>@class</code> attribute to specify its Java implementation class</p> <p>b) The Java implementation class is contained within another contribution (Contribution C) in the SCA Domain</p> <p>c) The Java implementation class is exported by another contribution (Contribution B) in the SCA Domain</p> <p>d) The Java implementation class is also contained within the same contribution (Contribution A) as the component</p> <p>e) Contribution A uses a Java language specific mechanism to specify the location of the implementation class</p> <p>f) The SCA component is instantiated</p>
Predicate	The Java class specified on the <code>@class</code> attribute is loaded by the Java language specific mechanism
Prescription Level	mandatory
Tags	"implementation" "resolution"
Testcase ID	JCA_11002_TestCase
Test Assertion	JCA-TA-11002
Description	Tests that a Java interface containing a <code>@org.oasisopen.sca.annotation.OneWay</code> annotation is mapped to WSDL as if it contained a <code>@javax.jws.OneWay</code> annotation
Artifacts	JCA_11002_TestCase.java Test_JCA_11002.composite TestInvocation.wsdl

	TestClient_0002.composite Service1.java ServiceOneWay.java service1OneWayCallerImpl.java serviceOneWayImpl.java
Expected output	Positive test: "JCA_11002 request service1-operation1 invoked service2-operation1 invoked OneWay method inonly previously called with testInvoke"

555 **JCA_11003_TestCase**

556

Testcase ID	JCA_11003_TestCase
Test Assertion	JCA-TA-11003
Description	Tests that if a Java interface mapped from a WSDL interface by an SCA runtime contains a @WebService annotation, the interface is treated as if it had a @Remotable annotation
Artifacts	JCA_11003_TestCase.java Test_JCA_11003.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WS.java service1ImplWS.java
Expected output	Positive test: "JCA_11003 request service1-operation1 invoked"

557 **JCA_11004_TestCase**

558

Assertion ID	JCI-TA-9006
Source	[JCI90003]
Target	Java class referenced by @class attribute of <implementation.java/>
Prerequisites	
Predicate	The Java class conforms to Java SE 5.0
Prescription Level	mandatory
Tags	"implementation" "Java"
Testcase ID	JCA_11004_TestCase
Test Assertion	JCA-TA-11004
Description	Tests that if a Java interface used as a service interface contains JAXB 2.1 datatypes and annotations, the SCA runtime is able to map the interface to WSDL.
Artifacts	JCA_11004_TestCase.java Test_JCA_11004.composite TestInvocation.wsdl TestClient_0002.composite Service1.wsdl ServiceJAXB.java service1JAXBCallerImpl.java serviceJAXBImpl.java
Expected output	Positive test:

	"JCA_11004-request-service1-operation1-invoked-service2-operation1-invoked-JAXB-parameters-received"
--	--

559 **JCA_11005_TestCase**

560

561 **Appendix A.8 Test Assertions for SCA POJO Component**
 562 **Implementation Specification Section 10**

Testcase ID	JCA_11005_TestCase
Test Assertion	JCA-TA-11006
Description	Tests that a Java interface used to declare an SCA service interface in an <interface.java/> element does not contain the additional client-side asynchronous polling and callback methods defined by JAX-WS.
Artifacts	JCA_11005_TestCase.java Test_JCA_11005.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithAsyncMethods.java service1Impl.java
Expected output	Negative test: "exception"

563 **JCA_11006_TestCase**

564

Assertion ID	JCI-TA-10001
Source	[JCI100001]
Target	<import.java/> element of sca-contribution.xml
Prerequisites	
Predicate	The value of the @package attribute on the <import.java/> element is unique across all other <import.java/> elements within the contribution.
Prescription Level	mandatory
Tags	"import"
Testcase ID	JCA_11006_TestCase
Test Assertion	JCA-TA-11006
Description	Tests that where a Java interface used to declare an SCA reference interface contains the additional client side asynchronous polling and callback methods defined by JAX-WS, that the interface is treated as valid and is used for the reference.
Artifacts	JCA_11006_TestCase.java Test_JCA_11006.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithAsyncMethods.java service1AsyncReflImpl.java service1Impl.java

Expected output	Positive test: "JCA_11006-request-service1-operation1-invoked-service2-operation1-invoked"
-----------------	---

565 **JCA_11007_TestCase**

566

Testcase ID	JCA_11007_TestCase
Test Assertion	JCA-TA-11008
Description	Tests that where a Java implementation class has an SCA reference which uses an interface which contains the additional client side asynchronous-polling and callback methods defined by JAX-WS, that the introspected reference element in the component type for that implementation has an interface that does not contain the additional asynchronous-polling and callback methods.
Artifacts	JCA_11007_TestCase.java Test_JCA_11007.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithAsyncMethods.java service1AsyncReflmpl.java service1Impl.java
Expected output	Positive test: "JCA_11007-request-service1-operation1-invoked-service2-operation1-invoked"

567 **JCA_11008_TestCase**

568

Assertion ID	JCI-TA-10002
Source	[JCI100002]
Target	Java package with version referenced by <import.java/>
Prerequisites	a) An SCA component and implementation that uses the Java package specified in the <import.java/> element contained with Contribution A b) The Java implementation class is exported by another contribution (Contribution B) in the SCA Domain c) The SCA component is instantiated
Predicate	The Java classes in the package that is loaded, satisfy the <import.java/>
Prescription Level	mandatory
Tags	"import" "resolution"
Testcase ID	JCA_11008_TestCase
Test Assertion	JCA-TA-11009
Description	Tests that where a Java implementation class has an SCA reference which uses an interface which contains the additional client side asynchronous-polling and callback methods defined by JAX-WS, that the implementation class is able to use the asynchronous-polling and callback methods with the semantics described in the JAX-WS 2.1 specification.
Artifacts	JCA_11008_TestCase.java Test_JCA_11008.composite

	TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithAsyncMethods.java service1AsyncReflmpl.java service1Impl.java
Expected output	Positive test: "JCA_11008 request service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked"

569

570 **JCA_11009_TestCase**

571

Assertion ID	JCI-TA-10003
Source	[JCI100002]
Target	Java package with version referenced by <import.java/>
Prerequisites	a) An SCA component and implementation that uses the Java package specified in the <import.java/> element contained with Contribution A b) The Java implementation class is contained within contribution (Contribution B) in the SCA Domain c) The @location attribute on <import.java/> points to Contribution B c) The SCA component is instantiated
Predicate	The Java classes in the package that is loaded from contribution B, satisfy the <import.java/>
Prescription Level	mandatory
Tags	"import" "resolution"
Testcase ID	JCA_11009_TestCase
Test Assertion	JCA-TA-11013
Description	Tests that where a Java implementation class has an @WebService annotation with the @endpointInterface attribute has its (Java) service interface defined by the interface referenced by the @endpointInterface attribute
Artifacts	JCA_11009_TestCase.java Test_JCA_11009.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service3Operations.java Service1Calls3OperationsImpl.java Service3OperationsWebServiceImpl.java
Expected output	Positive test: "JCA_11009 request service1 operation1 invoked"

572

573 **JCA_11010_TestCase**

574

Assertion ID	JCI-TA-10004
Source	[JCI100003]
Target	<export.java> with @package that uses the "uses" directive
Prerequisites	<p>a) An SCA component and implementation contained within Contribution A that imports the Java packages specified in the <import.java/> element</p> <p>b) Contribution A also imports one or more of the packages specified in the uses directive of <import.java/></p> <p>c) Another contribution (Contribution B) contains the <export.java/> element</p> <p>d) A third contribution (Contribution C) also exports the the same Java package as Contribution B, but it does not contain the "uses" directive</p> <p>e) The SCA component is instantiated</p>
Predicate	The Java classes required by Contribution A are all loaded from Contribution B
Prescription Level	mandatory
Tags	"export" "resolution"
Testcase ID	JCA_11010_TestCase
Test Assertion	JCA-TA-11014
Description	Tests that where a Java service interface contains a @WebParam annotation with its @header attribute set to "true", that the interface is treated as if it had the SOAP intent applied
Artifacts	JCA_11010_TestCase.java Test_JCA_11010.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1Impl.java Service1WithWebParam.java definitions.xml (in contribution JCA_General_POJO)
Expected output	Negative test: "exception"

575

576 **JCA_11011_TestCase**

577

Assertion ID	JCI-TA-10005
Source	[JCI100004]
Target	<export.java/> element of a contribution
Prerequisites	
Predicate	The value of the @package attribute on the <export.java/> element is unique across all other <export.java/> elements within the contribution.

Prescription Level	mandatory
Tags	"export"
Testcase ID	JCA_11011_TestCase
Test Assertion	JCA-TA-11015
Description	Tests that where a Java service interface contains a @WebResult annotation with its @header attribute set to "true", that the interface is treated as if it had the SOAP intent applied
Artifacts	JCA_11011_TestCase.java Test_JCA_11011.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1Impl.java Service1WithWebResult.java definitions.xml (in contribution JCA_General_POJO)
Expected output	Negative test: "exception"

578

579 **JCA_11012_TestCase**

580

Assertion ID	JCI-TA-10006
Source	[JCI100007]
Target	@package attribute of <export.java/>
Prerequisites	
Predicate	The Java package specified on the @package attribute is contained with the same contribution as the <export.java/>
Prescription Level	mandatory
Tags	"export" "resolution"
Testcase ID	JCA_11012_TestCase
Test Assertion	JCA-TA-11020
Description	Tests that where a Java service interface contains a @SOAPBinding annotation with its @header attribute set to "true", that the interface is treated as if it had the SOAP intent applied
Artifacts	JCA_11012_TestCase.java Test_JCA_11012.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1Impl.java Service1WithWebSoapBinding.java definitions.xml (in contribution JCA_General_POJO)
Expected output	Negative test: "exception"

581

582 **JCA_11013_TestCase**

583

Assertion ID	JCI-TA-10007
Source	[JCI100010]
Target	SCA Runtime
Prerequisites	
Predicate	All classes loaded from a contribution are loaded by a contribution unique class loader.
Prescription Level	mandatory
Tags	"class-loader"
Testcase ID	JCA_11013_TestCase
Test Assertion	JCA-TA-11016
Description	Tests that where a Java service implementation class contains a @ServiceMode annotation that the service interface is treated as if it had the SOAP intent applied
Artifacts	JCA_11013_TestCase.java Test_JCA_11013.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithServiceModelImpl.java definitions.xml (in contribution JCA_General_POJO)
Expected output	Negative test: "exception"

584

585 **JCA_11014_TestCase**

586

Assertion ID	JCI-TA-10008
Source	[JCI100011]
Target	SCA Runtime
Prerequisites	a) Contribution A imports Java classes b) Contribution B exports the same Java classes
Predicate	Contribution B's class loader loads classes from Contribution A
Prescription Level	mandatory
Tags	"class-loader"
Testcase ID	JCA_11014_TestCase
Test Assertion	JCA-TA-11017
Description	Tests that where a Java interface used to the define the interface of an SCA service does not contains an @WebServiceClient annotation
Artifacts	JCA_11014_TestCase.java

	Test_JCA_11014.composite TestInvocation.wsdl TestClient_0002.composite Service1WithWebServiceClient.java service1Impl.java
Expected output	Negative test: "exception"

587

588 **JCA_11015_TestCase**

589

<u>Assertion ID</u>	JCI-TA-10009
<u>Source</u>	[JCI100009]
<u>Target</u>	SCA Runtime
<u>Prerequisites</u>	a) <u>SCA POJO component contained within Contribution A</u> b) <u>The SCA component is instantiated</u>
<u>Predicate</u>	<u>SCA Runtime's thread context class loader is the class loader for Contribution A</u>
<u>Prescription Level</u>	<u>mandatory</u>
<u>Tags</u>	<u>"class-loader"</u>
<u>Testcase ID</u>	JCA_11015_TestCase
<u>Test Assertion</u>	JCA-TA-11018
<u>Description</u>	Tests that where a Java service implementation class contains a @WebServiceProvider annotation, the class is treated as if it is annotated with an SCA @Remotable annotation
<u>Artifacts</u>	JCA_11015_TestCase.java Test_JCA_11015.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithWebServiceProviderImpl.java
<u>Expected output</u>	Positive test: "JCA_11015 request service1-operation1 invoked"

590

591

Appendix B Cross Mapping of Normative Statements to Test Assertions

592

593

This section contains a list of normative statements in the SCA POJO Component Implementation specification and the corresponding Test Assertions.

594

Conformance statement	Test Assertion
JCI20001	JCI-TA-2001
JCI20002	JCI-TA-2002

595

JCA_11016_TestCase

596

Testcase ID	JCA_11016_TestCase
Test Assertion	JCA-TA-11019
Description	Tests that where a Java service implementation class contains a @WebService annotation and the @wsdlLocation attribute of that annotation is declared, referencing a WSDL document, that the class is treated as if its service interface is defined by the referenced WSDL document
Artifacts	JCA_11016_TestCase.java Test_JCA_11016.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1WithWebServiceProviderImpl.java
Expected output	Negative test: "exception"

597

Conformance statement	Test Assertion
JCI40001	JCI-TA-4001 JCI-TA-4002

598

JCA_11017_TestCase

599

Testcase ID	JCA_11017_TestCase
Test Assertion	JCA-TA-9066 JCA-TA-9068 JCA-TA-11009 JCA-TA-11010
Description	Tests that where a client component Java implementation has a reference that uses the async callback client API to invoke a service which is an "asynchronous" asynchronous service, where the response to an operation is received some arbitrary time after the request is made, the client is able to make the invocation successfully and get back the result - covers both a regular response and a fault response.
Artifacts	JCA_11017_TestCase.java Test_JCA_11017.composite TestInvocation.wsdl TestClient_0002.composite Service1.java Service1AsyncServer.java

	service1AsyncRefImpl.java Service1AsyncServerImpl.java
Expected output	Positive test: "JCA_11017-request-service1-operation1-invoked-Future-returned-service2-operation1-invoked-asynchro- org.oasisopen.sca.test.BusinessFault1:- service2-operation1-invoked-asynchronously"

600

<u>Conformance statement</u>	<u>Test Assertion</u>
<u>JCI50001</u>	<u>JCI-TA-5001</u>
<u>JCI50002</u>	<u>JCI-TA-5006</u>
<u>JCI50004</u>	<u>JCI-TA-5002</u> <u>JCI-TA-5003</u> <u>JCI-TA-5004</u> <u>JCI-TA-5005</u>
<u>JCI50005</u>	<u>JCI-TA-5007</u>

601

5 Cross Mapping of Test Assertions to TestCases

602

Test Assertion	Test Cases
JCA-TA-2001	JCA_2001_TestCase
JCA-TA-2002	JCA_2002_TestCase
JCA-TA-2003	JCA_2003_TestCase
JCA-TA-2004	JCA_2004_TestCase
JCA-TA-2005	JCA_2005_TestCase
JCA-TA-2006	JCA_2006_TestCase
JCA-TA-2007	JCA_2007_TestCase
JCA-TA-2008	JCA_2008_TestCase
JCA-TA-2009	Optional - not tested
JCA-TA-2010	JCA_2009_TestCase
JCA-TA-2011	JCA_2010_TestCase

603

Conformance statement	Test Assertion
JCI60001	JCI-TA-6001 JCI-TA-6002

Test Assertion	Test Cases
JCA-TA-3001	JCA_3001_TestCase JCA_3002_TestCase
JCA-TA-3002	JCA_3003_TestCase JCA_3004_TestCase
JCA-TA-3003	JCA_3005_TestCase JCA_3006_TestCase JCA_3007_TestCase
JCA-TA-3004	JCA_3008_TestCase
JCA-TA-3005	JCA_3009_TestCase JCA_3010_TestCase
JCA-TA-3006	JCA_3011_TestCase
JCA-TA-3007	JCA_3012_TestCase
JCA-TA-3008	JCA_3013_TestCase
JCA-TA-3009	JCA_3014_TestCase

604

Test Assertion	Test Cases
JCA-TA-4001	JCA_4001_TestCase
JCA-TA-4002	JCA_4001_TestCase
JCA-TA-4003	JCA_4001_TestCase
JCA-TA-4004	JCA_4002_TestCase
JCA-TA-4005	JCA_4001_TestCase
JCA-TA-4006	JCA_4001_TestCase
JCA-TA-4007	JCA_4001_TestCase
JCA-TA-4008	JCA_4001_TestCase
JCA-TA-4009	JCA_4001_TestCase

JCA-TA-4010	JCA_4003_TestCase
JCA-TA-4011	JCA_4001_TestCase
JCA-TA-4012	JCA_4004_TestCase
JCA-TA-4013	JCA_4001_TestCase
JCA-TA-4014	JCA_4001_TestCase
JCA-TA-4015	JCA_4005_TestCase
JCA-TA-4016	JCA_4001_TestCase
JCA-TA-4017	JCA_4001_TestCase
JCA-TA-4018	JCA_4001_TestCase
JCA-TA-4019	untestable
JCA-TA-4020	JCA_4001_TestCase
JCA-TA-4021	JCA_4001_TestCase
JCA-TA-4022	JCA_4007_TestCase
JCA-TA-4023	JCA_4001_TestCase
JCA-TA-4024	JCA_4008_TestCase

605

Conformance statement	Test Assertion
<u>JCI8001</u>	<u>JCI-TA-8001, JCI-TA-8002, JCI-TA-8003, JCI-TA-8004, JCI-TA-8005, JCI-TA-8006, JCI-TA-8007, JCI-TA-8008, JCI-TA-8009, JCI-TA-8010, JCI-TA-8011, JCI-TA-8012, JCI-TA-8013, JCI-TA-8016, JCI-TA-8017, JCI-TA-8018, JCI-TA-8019, JCI-TA-8020, JCI-TA-8021, JCI-TA-8022, JCI-TA-8023, JCI-TA-8024, JCI-TA-8025, JCI-TA-8026, JCI-TA-8027, JCI-TA-8028, JCI-TA-8029, JCI-TA-8030, JCI-TA-8031, JCI-TA-8032, JCI-TA-8033, JCI-TA-8034, JCI-TA-8035, JCI-TA-8036, JCI-TA-8037</u>
<u>JCI8002</u>	<u>JCI-TA-8014 JCI-TA-8015</u>

606

Test Assertion	Test Cases
JCA-TA-7001	JCA_7001_TestCase
JCA-TA-7002	JCA_7002_TestCase
JCA-TA-7003	JCA_7003_TestCase JCA_7004_TestCase
JCA-TA-7004	JCA_7003_TestCase
JCA-TA-7005	JCA_7003_TestCase
JCA-TA-7006	JCA_7005_TestCase JCA_7006_TestCase
JCA-TA-7007	JCA_7003_TestCase

607

Conformance statement	Test Assertion
<u>JCI90001</u>	<u>JCI-TA-9001</u>
<u>JCI90002</u>	<u>JCI-TA-9002</u>

Conformance statement	Test Assertion
	<u>JCI-TA-9003</u> <u>JCI-TA-9004</u> <u>JCI-TA-9005</u>
<u>JCI90003</u>	<u>JCI-TA-9006</u>

Test Assertion	Test Cases
<u>JCA-TA-8001</u>	<u>JCA_8001_TestCase</u>
<u>JCA-TA-8002</u>	<u>JCA_8002_TestCase</u> <u>JCA_8005_TestCase</u>
<u>JCA-TA-8003</u>	<u>JCA_8003_TestCase</u> <u>JCA_8005_TestCase</u>
<u>JCA-TA-8004</u>	<u>JCA_8004_TestCase</u> <u>JCA_8005_TestCase</u>
<u>JCA-TA-8005</u>	<u>JCA_8006_TestCase</u>
<u>JCA-TA-8006</u>	<u>JCA_8007_TestCase</u>
<u>JCA-TA-8007</u>	<u>JCA_8008_TestCase</u>
<u>JCA-TA-8008</u>	<u>JCA_8009_TestCase</u>
<u>JCA-TA-8009</u>	<u>JCA_8010_TestCase</u>
<u>JCA-TA-8010</u>	<u>JCA_8011_TestCase</u>

608
609

Conformance statement	Test Assertion
<u>JCI100001</u>	<u>JCI-TA-10001</u>
<u>JCI100002</u>	<u>JCI-TA-10002</u> <u>JCI-TA-10003</u>
<u>JCI100003</u>	<u>JCI-TA-10004</u>
<u>JCI100004</u>	<u>JCI-TA-10005</u>
<u>JCI100007</u>	<u>JCI-TA-10006</u>
<u>JCI100008</u>	<u>JCI-TA-9002</u> <u>JCI-TA-9003</u> <u>JCI-TA-9004</u> <u>JCI-TA-9005</u>
<u>JCI100009</u>	<u>JCI-TA-10009</u>
<u>JCI100010</u>	<u>JCI-TA-10007</u>
<u>JCI100011</u>	<u>JCI-TA-10008</u>

Test Assertion	Test Cases
<u>JCA-TA-9001</u>	<u>JCA_9001_TestCase</u> <u>JCA_9008_TestCase</u>
<u>JCA-TA-9002</u>	<u>JCA_9002_TestCase</u>
<u>JCA-TA-9003</u>	<u>JCA_9003_TestCase</u>
<u>JCA-TA-9004</u>	<u>JCA_9004_TestCase</u>
<u>JCA-TA-9005</u>	<u>JCA_9005_TestCase</u>
<u>JCA-TA-9006</u>	<u>JCA_9006_TestCase</u>

JCA-TA-9007	JCA_9006_TestCase
JCA-TA-9008	JCA_9006_TestCase
JCA-TA-9009	JCA_9007_TestCase
JCA-TA-9010	JCA_9008_TestCase
JCA-TA-9011	JCA_9008_TestCase
JCA-TA-9012	JCA_9008_TestCase
JCA-TA-9013	JCA_9008_TestCase
JCA-TA-9014	JCA_9006_TestCase
JCA-TA-9015	JCA_9009_TestCase
JCA-TA-9016	JCA_9009_TestCase
JCA-TA-9017	JCA_9009_TestCase
JCA-TA-9018	JCA_9009_TestCase
JCA-TA-9019	JCA_9009_TestCase
JCA-TA-9020	JCA_9010_TestCase
JCA-TA-9021	JCA_9010_TestCase
JCA-TA-9022	JCA_9010_TestCase
JCA-TA-9023	JCA_9010_TestCase
JCA-TA-9024	JCA_9010_TestCase
JCA-TA-9025	JCA_9011_TestCase
JCA-TA-9026	JCA_9011_TestCase
JCA-TA-9027	JCA_9012_TestCase
JCA-TA-9028	JCA_9012_TestCase
JCA-TA-9029	JCA_9012_TestCase
JCA-TA-9030	JCA_9013_TestCase
JCA-TA-9031	JCA_9013_TestCase
JCA-TA-9032	JCA_9013_TestCase
JCA-TA-9033	JCA_9013_TestCase
JCA-TA-9034	JCA_9014_TestCase
JCA-TA-9035	JCA_9014_TestCase
JCA-TA-9036	Untestable -- no standard means of configuring a service to have a JAAS subject
JCA-TA-9037	Untestable -- no standard means of configuring a service to have a JAAS subject
JCA-TA-9038	Untestable -- no standard means of configuring a service to have a JAAS subject
JCA-TA-9039	JCA_9015_TestCase
JCA-TA-9040	JCA_9015_TestCase
JCA-TA-9041	JCA_9015_TestCase
JCA-TA-9042	JCA_9015_TestCase
JCA-TA-9043	JCA_9015_TestCase
JCA-TA-9044	JCA_9015_TestCase
JCA-TA-9045	JCA_9015_TestCase
JCA-TA-9046	JCA_9015_TestCase
JCA-TA-9047	JCA_9015_TestCase
JCA-TA-9048	JCA_9015_TestCase
JCA-TA-9049	JCA_9015_TestCase
JCA-TA-9050	JCA_9014_TestCase
JCA-TA-9051	JCA_9016_TestCase

JCA-TA-9052	JCA_9016_TestCase
JCA-TA-9053	JCA_9016_TestCase
JCA-TA-9054	JCA_9016_TestCase
JCA-TA-9055	JCA_9016_TestCase
JCA-TA-9056	JCA_9016_TestCase
JCA-TA-9057	JCA_9016_TestCase
JCA-TA-9058	JCA_9016_TestCase
JCA-TA-9059	JCA_9016_TestCase
JCA-TA-9060	JCA_9016_TestCase
JCA-TA-9061	Impossible to obtain a SCAClientFactory instance with an invalid domainURI - untestable
JCA-TA-9062	Untestable since this is a protected method.
JCA-TA-9063	Impossible to obtain a SCAClientFactory instance with an invalid domainURI - untestable
JCA-TA-9064	Untestable unless we make the vendor implementation of SCAClientFactoryFinder a conformance target
JCA-TA-9065	Untestable unless we make the vendor implementation of SCAClientFactoryFinder a conformance target
JCA-TA-9066	JCA_7003_TestCase
JCA-TA-9067	JCA_7005_TestCase
JCA-TA-9068	JCA_7006_TestCase
JCA-TA-9069	JCA_7006_TestCase
JCA-TA-9070	JCA_9006_TestCase

611
612
613

Appendix C Cross Mapping of Test Assertions to TestCases

<u>Test Assertion</u>	<u>Test Cases</u>
JCI-TA-2001	POJO_2001_TestCase
JCI-TA-2002	POJO_2002_TestCase
<u>Test Assertion</u>	<u>Test Cases</u>
JCA-TA-10001	JCA_10001_TestCase
JCA-TA-10002	JCA_10002_TestCase
JCA-TA-10003	JCA_10003_TestCase
JCA-TA-10004	JCA_2009_TestCase JCA_2010_TestCase
JCA-TA-10031	JCA_10004_TestCase
JCA-TA-10005	JCA_4001_TestCase JCA_10005_TestCase JCA_10006_TestCase
JCA-TA-10006	JCA_4001_TestCase JCA_10007_TestCase
JCA-TA-10007	JCA_4001_TestCase
JCA-TA-10008	JCA_2005_TestCase
JCA-TA-10009	JCA_10008_TestCase
JCA-TA-10010	JCA_4001_TestCase
JCA-TA-10011	JCA_10009_TestCase
JCA-TA-10012	JCA_10005_TestCase JCA_10010_TestCase
JCA-TA-10013	JCA_10005_TestCase JCA_10011_TestCase
JCA-TA-10014	JCA_10012_TestCase
JCA-TA-10015	JCA_2001_TestCase
JCA-TA-10016	JCA_10013_TestCase
JCA-TA-10017	JCA_2001_TestCase
JCA-TA-10018	JCA_10014_TestCase
JCA-TA-10019	JCA_10015_TestCase
JCA-TA-10020	JCA_2001_TestCase
JCA-TA-10021	JCA_2001_TestCase
JCA-TA-10022	JCA_10016_TestCase
JCA-TA-10023	JCA_2008_TestCase
JCA-TA-10024	JCA_10044_TestCase
JCA-TA-10025	JCA_10045_TestCase
JCA-TA-10026	optional requirement -- no test
JCA-TA-10027	optional requirement -- no test
JCA-TA-10028	optional requirement -- no test
JCA-TA-10029	optional requirement -- no test
JCA-TA-10030	requires undeploy API -- no test

JCA-TA-10031	no-TA
JCA-TA-10032	no-TA
JCA-TA-10033	requires redeploy API -- no test
JCA-TA-10034	optional requirement -- no test
JCA-TA-10035	requires undeploy API -- no test
JCA-TA-10036	requires undeploy API -- no test
JCA-TA-10037	requires redeploy API -- no test
JCA-TA-10038	requires deploy/redeploy API -- no test
JCA-TA-10039	requires deploy/redeploy API -- no test
JCA-TA-10040	requires deploy/redeploy API -- no test
JCA-TA-10041	requires deploy/redeploy API -- no test
JCA-TA-10042	requires deploy/redeploy API -- no test
JCA-TA-10043	requires deploy/redeploy API -- no test
JCA-TA-10044	requires optional support for rewiring -- no test
JCA-TA-10045	JCA_10046_TestCase
JCA-TA-10046	JCA_10047_TestCase
JCA-TA-10047	JCA_10048_TestCase
JCA-TA-10050	JCA_10049_TestCase
JCA-TA-10055	JCA_10050_TestCase
JCA-TA-10056	requires deploy/redeploy API -- no test
JCA-TA-10057	JCA_10051_TestCase
JCA-TA-10058	JCA_10052_TestCase

614

<u>Test Assertion</u>	<u>Test Cases</u>
<u>JCI-TA-4001</u>	<u>POJO_4001_TestCase</u>
<u>JCI-TA-4002</u>	<u>POJO_4002_TestCase</u>

615

<u>Test Assertion</u>	<u>Test Cases</u>
<u>JCI-TA-5001</u>	<u>POJO_5001_TestCase</u>
<u>JCI-TA-5002</u>	<u>POJO_5002_TestCase</u>
<u>JCI-TA-5003</u>	<u>POJO_5003_TestCase</u>
<u>JCI-TA-5004</u>	<u>POJO_5004_TestCase</u>
<u>JCI-TA-5005</u>	<u>POJO_5005_TestCase</u>
<u>JCI-TA-5006</u>	<u>POJO_5006_TestCase</u>
<u>JCI-TA-5007</u>	<u>POJO_5007_TestCase</u>
<u>Test Assertion</u>	<u>Test Cases</u>
JCA-TA-11001	JCA_11001_TestCase
JCA-TA-11002	JCA_11002_TestCase
JCA-TA-11003	JCA_11003_TestCase
JCA-TA-11004	JCA_11004_TestCase
JCA-TA-11005	JCA_11005_TestCase

JCA-TA-11006	JCA_11006_TestCase
JCA-TA-11007	optional - no test case
JCA-TA-11008	JCA_11007_TestCase
JCA-TA-11009	JCA_11008_TestCase
JCA-TA-11010	JCA_7003_TestCase JCA_7004_TestCase
JCA-TA-11011	JCA_11003_TestCase
JCA-TA-11012	JCA_3014_TestCase
JCA-TA-11013	JCA_11009_TestCase
JCA-TA-11014	JCA_11010_TestCase
JCA-TA-11015	JCA_11011_TestCase
JCA-TA-11016	JCA_11013_TestCase
JCA-TA-11017	JCA_11014_TestCase
JCA-TA-11018	JCA_11015_TestCase
JCA-TA-11019	JCA_11016_TestCase
JCA-TA-11020	JCA_11012_TestCase
JCA-TA-11021	Tested by a range of the testcases in this test suite - default mappings - mappings influenced by JAXWS annotations - JAXB type mapping are all dealt with in various different testcases, so there is no need to create a new specific testcase for this general requirement.

616

<u>Test Assertion</u>	<u>Test Cases</u>
JCI-TA-6001	POJO_6001_TestCase
JCI-TA-6002	POJO_6002_TestCase

617 | **6 Catalog of Test Artifacts**

618 |

619 | **6.1 Composite Files – lower level**

Name	Valid	Description
Marked "Impl-Specific" if it is implementation-type specific		Services and References use interface Service1 unless described otherwise
CompositeScope.composite	Y	Composite containing a component with a composite-scope implementation (service1CompositeImpl.java) invoked by a variety of other components
TestClient_002.composite	Y	Test invocation composite that presents the TestInvocation interface to the test runner client.

620 |

Test Assertion	Test Cases
JCI-TA-8001	POJO_8001_TestCase
JCI-TA-8002	POJO_8002_TestCase
JCI-TA-8003	POJO_8003_TestCase
JCI-TA-8004	POJO_8004_TestCase
JCI-TA-8005	POJO_8005_TestCase
JCI-TA-8006	POJO_8006_TestCase
JCI-TA-8007	POJO_8015_TestCase
JCI-TA-8008	POJO_8007_TestCase
JCI-TA-8009	POJO_8008_TestCase
JCI-TA-8010	POJO_8009_TestCase
JCI-TA-8011	POJO_8010_TestCase
JCI-TA-8012	POJO_8011_TestCase
JCI-TA-8013	POJO_8012_TestCase
JCI-TA-8014	POJO_8013_TestCase
JCI-TA-8015	POJO_8014_TestCase
JCI-TA-8016	POJO_8016_TestCase

JCI-TA-8017	POJO_8017_TestCase
JCI-TA-8018	POJO_8018_TestCase
JCI-TA-8019	POJO_8019_TestCase
JCI-TA-8020	POJO_8020_TestCase
JCI-TA-8021	POJO_8021_TestCase
JCI-TA-8022	POJO_8022_TestCase
JCI-TA-8023	POJO_8023_TestCase
JCI-TA-8024	POJO_8024_TestCase
JCI-TA-8025	POJO_8025_TestCase
JCI-TA-8026	POJO_8026_TestCase
JCI-TA-8027	POJO_8027_TestCase
JCI-TA-8028	POJO_8028_TestCase
JCI-TA-8029	POJO_8029_TestCase
JCI-TA-8030	POJO_8030_TestCase
JCI-TA-8031	POJO_8031_TestCase
JCI-TA-8032	POJO_8032_TestCase
JCI-TA-8033	POJO_8033_TestCase
JCI-TA-8034	POJO_8034_TestCase
JCI-TA-8035	POJO_8035_TestCase
JCI-TA-8036	POJO_8036_TestCase
JCI-TA-8037	POJO_8037_TestCase

621 | **6.2 Java Interfaces**

622 |

Name	Description
Constants.java	Policy-related constants declared in an interface
DataStore.java	Data-store interface used as service interface for data-store component
MultipleService.java	Interface with a large number of operations, used to check multiple invocations to a service component
ParallelService.java	
Service1.java	Interface with 1 operation

	<ul style="list-style-type: none"> –"operation1", string input, string output
Service1AsyncServer.java	SCA Async Server form of the Service1 interface
Service1Superset.java	<ul style="list-style-type: none"> –A superset of the Service1 interface –"operation1", string input, string output –"operation2", string input, string output
Service1WithAsyncMethods.java	<p>Service1 service interface</p> <ul style="list-style-type: none"> – this version has all the additional JAXWS-async-client methods added
Service1WithFaults.java	Version of the Service1 interface with a series of declared business faults on operation1
Service1WithFaultsAsyncServer.java	<p>Service1 service interface with with a series of declared business faults on operation1</p> <ul style="list-style-type: none"> –Asynchronous server version
Service1WS.java	<p>Service1 service interface</p> <ul style="list-style-type: none"> –a variant which has a @WebService annotation –and no @Remotable annotation
Service2.java	<p>A test service interface, designed to be incompatible with the Service1 interface</p> <ul style="list-style-type: none"> 1 operation "operation2", int input, int output
Service3Callback.java	<p>A remotable service interface used for a Callback from Service3</p> <ul style="list-style-type: none"> 1 operation "callback1", string input, string output
Service3Operations.java	<ul style="list-style-type: none"> –A service with 3 operations –operation1: input String response String –operation2: input String response String –operation3: input String response String
Service3OperationsWSDL.java	<p>A service with 3 operations – with a @WebService annotation that references a WSDL</p> <ul style="list-style-type: none"> –operation1: input String response String –operation2: input String response String –operation3: input String response String
Service3WithCallback.java	<p>A bi-directional service interface with callback interface Service3Callback</p> <ul style="list-style-type: none"> –1 operation "operation3", string input, string output
Service4.java	<p>A remotable service interface containing 1 operation with a (Java) mutable parameter</p> <ul style="list-style-type: none"> 1 operation "operation1" StringBuffer input, string-

	output
Service5Intents.java	<ul style="list-style-type: none"> 1 operation "operation1", string input, string output -entire interface is annotated with TestIntent4 -operation1 is annotated with TestIntent3
Service6PolicySets.java	<ul style="list-style-type: none"> Service interface -1 operation "operation1", string input, string output -interface is annotated with policy set PolicySet1 -operation is annotated with policy set PolicySet2
Service7Callback.java	<ul style="list-style-type: none"> Remotable service interface used for a Callback from Service7 -1 operation "service7Callback1" string input, string output
Service7WithCallback.java	<ul style="list-style-type: none"> A bi-directional service interface with callback interface Service7Callback 1 operation "operation7", string input, string output
ServiceJAXB.java	<ul style="list-style-type: none"> ServiceJAXB service interface -1 operation "operation1", JAXB_Class1 input, JAXB_Class2 output This interface has JAXB classes as parameters
ServiceOneWay.java	<ul style="list-style-type: none"> ServiceOneWay service interface 2 operations: <ul style="list-style-type: none"> -"inonly1" a OneWay method, string input -"operation1" a Request/Response method, string input, string output
ServiceOneWayMissing.java	<ul style="list-style-type: none"> ServiceOneWay service interface but with the inonly1 operation missing its @OneWay annotation 2 operations <ul style="list-style-type: none"> -"inonly1" a OneWay method missing its @OneWay annotation, string input -"operation1" a Request/Response method, string input, string output
TestInvocation.java	<ul style="list-style-type: none"> Basic interface to invoke testcases 1 operation <ul style="list-style-type: none"> -"invokeTest", string input, string output
TestIntent1.java	Intent annotation for test:testIntent1
TestIntent2.java	Intent annotation for test:testIntent2
TestIntent3.java	Intent annotation for test:testIntent3

TestIntent4.java	Intent annotation for test:testIntent4
TestIntent5.java	Intent annotation for test:testIntent5
TestIntent6.java	Intent annotation for test:testIntent6

6.3 Java Implementation Classes

623

624

<u>Test Assertion</u>	<u>Test Cases</u>
JCI-TA-9001	POJO_9001_TestCase
JCI-TA-9002	POJO_9002_TestCase
JCI-TA-9003	POJO_9003_TestCase
JCI-TA-9004	POJO_9004_TestCase
JCI-TA-9005	untestable with current mandatory requirements
JCI-TA-9006	no obvious test available
Name	Description
	Services and references use interface Service1 unless otherwise described
ASM_0002_Client.java	1 service implementing TestInvocation 0 references
asyncControllerImpl.java	1 service 2 references (1..1) -reference1 with interface Service1 -dataStore with interface DataStore Acts as an invoker for an Async service and retrieves results from the DataStore
asyncControllerWithFaultsImpl.java	1 service 2 references (1..1) -reference1 with interface Service1WithFaults -dataStore with interface DataStore Acts as an invoker for an Async service which can throw Faults and retrieves results from the DataStore
BusinessFault1.java	A business fault class
BusinessFault2.java	A business fault class
compositeEagerInitImpl.java	1 service

	<p>1 reference (1..1)</p> <p>Implementation is COMPOSITE scope and is marked with @EagerInit – reference is invoked from method marked with @Init</p>
dataStoreCompositeImpl.java	<p>1 service with DataStore interface</p> <p>0 references</p> <p>COMPOSITE scope – acts as a store for data under a key</p>
lifecycleControllerImpl.java	<p>1 service</p> <p>2 references</p> <ul style="list-style-type: none"> – reference 1 (1..1) with interface Service1 – dataStore (1..1) with interface DataStore <p>Acts as a controller component for a class that undergoes a series of lifecycle tests – the results are stored into a data store component and are retrieved at the end by this implementation</p>
MultipleCallbacksImpl.java	<p>3 services</p> <ul style="list-style-type: none"> – Service1 with interface Service1 – Service3WithCallback with interface Service3WithCallback – Service7WithCallback with interface Service7WithCallback <p>0 references</p>
multipleServiceClientImpl.java	<p>1 service</p> <p>1 reference</p> <ul style="list-style-type: none"> – reference 1 (1..1) with interface MultipleService <p>Invokes all the operations of MultipleService in sequence for a single reference</p>
multipleServiceImpl.java	<p>1 service with interface MultipleService</p> <p>0 references</p> <p>Implementation with many service operations that checks if a stateless instance ever gets invoked more than once</p>
parallelCompositeClientImpl.java	<p>1 service with interface Service1</p> <p>1 reference</p> <p>2 properties</p> <ul style="list-style-type: none"> – "invocationCount" – number of parallel invocations – "maxWaitTime" – max time to wait for a response from the reference invocation – "reference1" (1..1) with interface ParallelService

	reference1 is invoked multiple times in parallel on multiple threads
parallelCompositeServiceImpl.java	1 service with interface ParallelService 0 references Implementation has COMPOSITE scope
parallelServiceClientImpl.java	1 service with interface Service1 1 reference 2 properties —"invocationCount"—number of parallel invocations —"maxWaitTime"—max time to wait for a response from the reference invocation —"reference1" (1..1) with interface ParallelService reference1 is invoked multiple times in parallel on multiple threads
parallelServiceImpl.java	1 service with interface ParallelService 0 references Implementation has STATELESS scope
service1AsyncRefImpl.java	1 service with interface Service1 1 reference (1..1) "reference1" with JAXWS async-client form of Service1 interface 2 properties —"invokeMethods"—array of strings indicating how to invoke methods on reference1 —"invokeParm"—array of strings which are the input parameters to the invocations in invokeMethods
Service1AsyncServerImpl.java	1 service with interface Service1AsyncServer 0 references Async service implementation of Service1, with response sent back on a separate thread.
Service1AsyncServerMultiFaultImpl.java	1 service with interface Service1WithFaultsAsyncServer 1 reference (1..1) with interface DataStore Async service implementation of Service1WithFaults, which calls ResponseDispatch.sendFault() method more than once
Service1AsyncServerMultiResponseImpl.java	1 service with interface Service1AsyncServer 1 reference (1..1) with interface DataStore Async service implementation of Service1, which calls ResponseDispatch.sendResponse() method

	more than once
service1CallbackContextImpl1.java	<p>1 service with interface Service1</p> <p>1 reference (1..1) with interface Service3WithCallback which has a Callback interface Service3Callback</p> <p>Reference gets called when service1 operation1 is invoked</p>
service1CallbackContextImpl2.java	<p>1 service with interface Service1</p> <p>1 reference (1..1) with interface Service3WithCallback which has a Callback interface Service3Callback</p> <p>The callback operation's method accesses the getRequestContext() method of the ComponentContext API during the processing of a callback method and checks that returned RequestContext is not null</p>
service1CallbackImpl.java	<p>1 service with interface Service1,</p> <p>1 reference (1..1) with interface Service3WithCallback which has a Callback interface Service3Callback which is implemented by this class</p> <p>Reference gets called when service1 operation1 is invoked</p>
Service1Calls3OperationsImpl.java	<p>1 service with interface Service1,</p> <p>1 reference (1..1) with interface Service3Operations</p>
Service1GCcastImpl.java	<p>1 service with interface Service1</p> <p>1 reference "reference1" with interface Service1</p> <p>Implementation performs invocations of the cast() method of the ComponentContext API</p>
Service1GCcreateSelfReferenceAImpl.java	<p>1 service with interface Service1</p> <p>0 references-</p> <p>Implementation performs invocations of the createSelfReference(Class) method of the ComponentContext API</p>
Service1GCcreateSelfReferenceBImpl.java	<p>1 service with interface Service1</p> <p>0 references-</p> <p>Implementation performs invocations of the createSelfReference(Class, String) method of the ComponentContext API</p>
Service1GCgetPropertyImpl.java	<p>1 service with interface Service1</p> <p>0 references-</p>

	Implementation performs invocations of the <code>getProperty()</code> method of the <code>ComponentContext</code> API
<code>Service1CGgetServiceImpl.java</code>	1 service with interface <code>Service1</code> 0 references Implementation performs invocations of the <code>getService()</code> method of the <code>ComponentContext</code> API
<code>Service1CGgetServiceReferenceImpl.java</code>	Java component implementation with 1 service with interface <code>Service1</code> 3 references - <code>requiredRef (1..1)</code> with interface <code>Service1</code> - <code>singledRef (0..1)</code> with interface <code>Service1</code> (intended to be unwired) - <code>multiRef (0..n)</code> with interface <code>Service1</code> Implementation performs invocations of the <code>getServiceReference()</code> method of the <code>ComponentContext</code> API
<code>Service1CGgetServiceReferencesImpl.java</code>	Java component implementation with 1 service with interface <code>Service1</code> 4 references - <code>requiredRef (1..1)</code> with interface <code>Service1</code> - <code>singledRef (0..1)</code> with interface <code>Service1</code> (intended to be unwired) - <code>multiRef (0..n)</code> with interface <code>Service1</code> - <code>multiRef2 (0..n)</code> with interface <code>Service1</code> (intended to be unwired) Implementation performs invocations of the <code>getServiceReferences()</code> method of the <code>ComponentContext</code> API
<code>Service1CGgetServicesImpl.java</code>	Java component implementation with 1 service with interface <code>Service1</code> 4 references - <code>requiredRef (1..1)</code> with interface <code>Service1</code> - <code>singledRef (0..1)</code> with interface <code>Service1</code> (intended to be unwired) - <code>multiRef (0..n)</code> with interface <code>Service1</code> - <code>multiRef2 (0..n)</code> with interface <code>Service1</code> (intended to be unwired) Implementation performs invocations of the <code>getServices()</code> method of the <code>ComponentContext</code> API
<code>Service1CGgetURIImpl.java</code>	1 service with interface <code>Service1</code>

	<p>0 references</p> <p>Implementation performs invocations of the <code>getURI()</code> method of the <code>ComponentContext</code> API</p>
<code>service1CompositeImpl.java</code>	<p>1 service with interface <code>Service1</code></p> <p>0 references</p> <p>Implementation is <code>COMPOSITE</code> scope</p>
<code>service1ConstrImpl.java</code>	<p>1 service with interface <code>Service1</code></p> <p>0 references</p> <p>Implementation has constructor with parameter annotated with <code>@Property</code></p>
<code>service1ContextImpl1.java</code>	<p>1 service with interface <code>Service1</code></p> <p>1 reference (1..n) with interface <code>Service1</code></p> <p>all configured wires get called when <code>service1 operation1</code> is invoked</p> <p>Implementation invokes the <code>ComponentContext</code> <code>getService()</code> API</p>
<code>service1CoordinatorImpl.java</code>	<p>1 service with interface <code>Service1</code></p> <p>1 reference (1..n) with interface <code>Service1</code></p> <p>all configured wires get called when <code>service1 operation1</code> is invoked</p> <p>Implementation acts as a coordinator between each of the invocations</p> <ul style="list-style-type: none"> - it sends different data as input to each reference invocation - it records all the output data from all the invocations
<code>service1GoodIntent.java</code>	<p>1 service with interface <code>Service1</code></p> <p>3 references</p> <p>Implementation has all references annotated with <code>TestIntent2</code></p>
<code>service1Impl.java</code>	<p>1 service with interface <code>Service1</code></p> <p>0 references</p>
<code>service1Impl2.java</code>	<p>1 service with interface <code>Service1</code>;</p> <p>1 reference (1..1) with interface <code>Service1</code></p> <p>reference gets called when <code>service1 operation1</code> is invoked</p>
<code>service1Impl4.java</code>	<p>1 service with interface <code>Service1</code></p> <p>1 reference (0..n) with interface <code>Service1</code></p> <p>all configured wires get called when <code>service1 operation1</code> is invoked - reports an empty reference ar-</p>

	ray if it is unwired
service1Impl6.java	<ul style="list-style-type: none"> -1 service with interface Service1 -1 reference (0..1) with interface Service1 <p>configured wire gets called when service1 operation1 is invoked, if present otherwise, this implementation reports its absence</p>
service1Impl7.java	<ul style="list-style-type: none"> 1 service with interface Service1; 1 reference (1..1) with interface Service4 <ul style="list-style-type: none"> -reference gets called when service1 operation1 is invoked -reference is marked with @AllowsPassByReference
service1Impl7b.java	<ul style="list-style-type: none"> 1 service with interface Service1; 1 reference (1..1) with interface Service4 <ul style="list-style-type: none"> -reference gets called when service1 operation1 is invoked -reference is not marked @AllowsPassByReference and the implementation checks for alteration of input parameters
service1ImplWS.java	<p>Java implementation – NOT used as an SGA component implementation, but instead used in the process of generating a WSDL interface through the standard JDK wsgen tool, which follows JAXWS 2.1 rules</p> <ul style="list-style-type: none"> 1 service with interface Service1WS 0 references
service1InitCheckerImpl.java	<ul style="list-style-type: none"> 1 service with interface Service1 1 reference (1..1) with interface Service1 2 properties- <ul style="list-style-type: none"> -invocationCount -expectedResponse <p>The implementation invokes the reference invocationCount times and checks for the response expectedResponse</p>
service1InitImpl.java	<ul style="list-style-type: none"> 1 service with interface Service1 0 references <p>Implementation has a method marked with @Init - the response from the operation1 method of Service1 differs depending on whether the @Init method has been called before the operation1 method is called</p>
service1Intent.java	<ul style="list-style-type: none"> 1 service with interface Service1

	<p>0 references</p> <p>Implementation class is annotated with TestIntent2</p>
service1IntentQualifier.java	<p>1 service with interface Service1</p> <p>0 references</p> <p>Implementation class is annotated with TestIntent6.qual1</p>
service1JAXBCallerImpl.java	<p>1 service with interface Service1;</p> <p>1 reference (1..1) with interface ServiceJAXB</p> <p>reference gets called when service1 operation1 is invoked</p>
service1LifecycleExceptionsImpl.java	<p>1 service with interface Service1</p> <p>1 reference (1..1) with interface Service1</p> <p>Java STATELESS component implementation which tests the overall Lifecycle sequence of an SCA Java implementation class</p>
service1MultiPropImpl.java	<p>1 service with interface Service1</p> <p>0 references</p> <p>1 property which is multi-valued</p>
service1OneWayCallerImpl.java	<p>1 service with interface Service1;</p> <p>1 reference (1..1) with interface ServiceOneWay</p> <p>-reference OneWay operation gets called first when service1 operation1 is invoked</p> <p>-then reference request/response operation gets called</p>
Service1OneWayMissingCallerImpl.java	<p>1 service with interface Service1;</p> <p>1 reference (1..1) with interface ServiceOneWayMissing</p> <p>-reference OneWay operation gets called first when service1 operation1 is invoked</p> <p>-then reference request/response operation gets called</p>
service1OptMultiRef.java	<p>1 service with interface Service1</p> <p>1 reference (0..n) with interface Service1</p>
Service1RCgetServiceNameImpl.java	<p>1 service with interface Service1</p> <p>2 references</p> <p>-singleRef (1..1) with interface Service1</p> <p>-callback3Ref (1..1) with interface Service3WithCallback</p> <p>Implementation performs invocations of the Re-</p>

	questContext getServiceName() method of the API
service1RequestContextImpl1.java	<p>1 service with interface Service1</p> <p>0 references</p> <p>Implementation invokes the ComponentContext getRequestContext() API and checks that the returned value is a valid RequestContext</p>
service1RequestContextImpl2.java	<p>1 service with interface Service1</p> <p>0 references</p> <p>This implementation invokes the getServiceReference() method of the RequestContext API and checks that the returned ServiceReference represents the Service offered by this component</p>
Service1SCAGClientImpl.java	<p>1 service with interface Service1</p> <p>1 reference "reference1" with interface Service1</p> <p>Implementation performs invocations of the SCAG client API</p>
service1StatelessLifecycleImpl.java	<p>1 service with interface Service1</p> <p>1 reference (1..1) with interface Service1</p> <p>Java STATELESS component implementation which tests the overall Lifecycle sequence of an SCA Java implementation class</p>
service1SupersetImpl.java	<p>1 service with interface Service1Superset</p> <p>0 references</p>
service1UninitImpl.java	<p>1 service with interface Service1</p> <p>0 references</p> <p>Implementation refuses to initialize -- it throws an exception from its @Init method</p>
service1WSImpl.java	<p>1 service with interface Service1WS</p> <p>0 references</p>
service2Impl.java	<p>1 service with interface Service2</p> <p>0 references</p>
service3Impl1.java	<p>1 service with interface Service3WithCallback</p> <p>1 callback using interface Service3Callback</p> <p>0 references</p>
Service3OperationsWebServiceImpl.java	<p>1 service with interface Service3Operations, but declared through an @WebService annotation</p> <p>0 references</p>
Service3OperationsWSDLImpl.java	<p>1 service with interface Service3OperationsWSDL</p>

	0 references
service4Impl1.java	1 service with interface Service4, where the implementation method modifies the input parameter 0 references -service is marked with @AllowsPassByReference
service4Impl.java	1 service with interface Service4, where the implementation method modifies the input parameter 0 references
service5Impl2.java	1 service with interface Service1, 1 reference (1..1) with interface Service5Intents reference gets called when service1 operation1 is invoked
service5Impl.java	1 service with interface Service5Intents 0 references
service6Impl2.java	1 service with interface Service1, 1 reference (1..1) with interface Service6PolicySets reference gets called when service1 operation1 is invoked
service6Impl.java	1 service with interface Service6PolicySets 0 references
serviceJAXBImpl.java	1 service with interface ServiceJAXB 0 references invocation of operation1 causes the implementation to read data from the supplied JAXB_Class1 parameter and return data in all the fields of the returned JAXB_Class2 parameter
serviceOneWayImpl.java	1 service with interface ServiceOneWay 0 references The implementation is COMPOSITE scope and is stateful -it remembers whether its OneWay method inonly1 was called -it reports this when its request/Response method is called
TestException.java	Exception thrown by SCA Test services
JAXB.JAXB_Class1.java	JAXB annotated complex class that is used in a service interface
JAXB.JAXB_Class2.java	JAXB annotated complex class that is used in a service interface

625 **6.4 WSDL Interface Files**

626

Name	Description
Service1ImplWSService.wsdl	WSDL mapped from service1ImplWS class
Service3OperationsWSDL.wsdl	WSDL version of the Service3OperationsWSDL java interface, but containing only operations "operation1" and "operation2" -- "operation3" intentionally missing
TestClient.wsdl	WSDL version of the TestInvocation.java interface
TestInvocation.wsdl	WSDL version of the client TestInvocation.java interface

627

<u>Test Assertion</u>	<u>Test Cases</u>
<u>JCI-TA-10001</u>	<u>POJO_10001_TestCase</u>
<u>JCI-TA-10002</u>	<u>POJO_10002_TestCase</u>
<u>JCI-TA-10003</u>	<u>POJO_10003_TestCase</u>
<u>JCI-TA-10004</u>	<u>POJO_10004_TestCase</u>
<u>JCI-TA-10005</u>	<u>POJO_10005_TestCase</u>
<u>JCI-TA-10006</u>	<u>POJO_10006_TestCase</u>
<u>JCI-TA-10007</u>	<u>POJO_10007_TestCase</u>
<u>JCI-TA-10008</u>	<u>POJO_10008_TestCase</u>
<u>JCI-TA-10009</u>	<u>POJO_10009_TestCase</u>

628 | 7 Conformance

629 | ~~There are no conformance statements relating to the TestCases.~~

630 |

631 **Appendix D Acknowledgments**

632 The following individuals have participated in the creation of this specification and are gratefully acknow-
633 ledged

Participants:

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

634

635 |

636 |

Appendix E Revision History

Revision	Date	Editor	Changes Made
<u>1</u>	<u>09/25/09</u>	<u>Mike Edwards</u>	<u>Initial version</u>
<u>4</u>	<u>10/02/09</u>	<u>Dave Booz</u>	<u>Section 3 testcases</u>
<u>5</u>	<u>10/06/09</u>	<u>Dave Booz</u>	<u>Section 8, 8001-8005</u>
<u>7</u>	<u>10/07/09</u>	<u>Dave Booz</u>	<u>Complete Section 8</u>
<u>WD01</u>	<u>11/12/09</u>	<u>Mike Edwards</u>	<u>All sections complete</u>
<u>wd02</u>	<u>11/18/09</u>	<u>Mike Edwards</u>	<u>Added testcases in 8xxx range</u>
<u>wd03</u>	<u>07/26/10</u>	<u>Mike Edwards</u>	<u>Corrected cross-mapping tables</u> <u>Added testcases POJO_8015, POJO_8016</u>
<u>wd04</u>			
<u>wd05</u>	<u>07/29/10</u>	<u>Mike Edwards</u>	<u>Added testcases: POJO_8027_TestCase</u> <u>POJO_8028_TestCase</u> <u>POJO_8029_TestCase</u> <u>POJO_8030_TestCase</u> <u>POJO_8031_TestCase</u> <u>POJO_8032_TestCase</u>
<u>wd06</u>			
<u>Wd07</u>	<u>13/10/10</u>	<u>Bryan Aupperle</u>	<u>Fix cut/paste errors in test cases 8033-8036</u> <u>Update artifact tables</u>
<u>wd08</u>	<u>10/19/10</u>	<u>Mike Edwards</u>	<u>Apply Issue 214:</u> <u>- new testcase POJO_8037_TestCase</u>
<u>csdpr01</u>	<u>11/08/10</u>	<u>TC Admin</u>	<u>CSD01 & Public Review 01</u>
<u>wd011</u>	<u>06/27/11</u>	<u>Mike Edwards</u>	<u>Issue 222 - Section 1 & Frontmatter (Abstract)</u>
<u>wd12</u>	<u>06/27/11</u>	<u>Mike Edwards</u>	<u>Issue 236 - merge TestAssertions document</u> <u>into the Testcases document - major changes</u> <u>across the document.</u>
<u>wd13</u>	<u>08/15/11</u>	<u>Mike Edwards</u>	<u>All changes accepted</u> <u>Frontmatter and references updated for cds02</u> <u>draft</u>
Revision	Date	Editor	Changes Made
<u>1</u>	<u>09/25/09</u>	<u>Mike Edwards</u>	<u>Initial version</u>
<u>4</u>	<u>10/02/09</u>	<u>Dave Booz</u>	<u>Section 3 testcases</u>
<u>5</u>	<u>10/06/09</u>	<u>Dave Booz</u>	<u>Section 8, 8001-8005</u>
<u>7</u>	<u>10/07/09</u>	<u>Dave Booz</u>	<u>Complete Section 8</u>
<u>11</u>	<u>10/09/09</u>	<u>Mike Edwards</u>	<u>All sections complete</u>
<u>12</u>	<u>06/21/10</u>	<u>Mike Edwards</u>	<u>Testcases added for CD04 of Java CAA specification:</u> <u>JCA-4008</u>

			JCA_7001--JCA_7003 JCA_9006--JCA_9016
13	06/28/10	Mike Edwards	Further Testcases added for CD04: JCA_3013--JCA_3014 JCA_7004--JCA_7006 JCA11009--JCA_11017
ed01	07/19/10	Mike Edwards	All changes accepted Frontmatter adjusted to OASIS requirements

639