Service Component Architecture Client and Implementation Model for C++ Test Cases Version 1.1

Committee Draft 02

14 October 2010

Specification URIs:
This Version:
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd02.html
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd02.doc
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd02.pdf (Authoritative)

Previous Version:
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd01.html
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd01.doc
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd01.pdf (Authoritative)

Latest Version:
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases.html
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases.doc
http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases.pdf (Authoritative)

Technical Committee:
OASIS Service Component Architecture / C and C++ (SCA-CPP-C++) TC

Chair:
Bryan Aupperle, IBM

Editors:
Bryan Aupperle, IBM
David Haney
Pete Robbins, IBM

Related work:

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

- http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-testcases-cd02.zip

This document is related to:

- SCA Client and Implementation Model for C++ Specification Version 1.1

Declared XML Namespaces:
http://docs.oasis-open.org/ns/opencsa/scatests/200903
http://docs.oasis-open.org/ns/opencsa/scatests/2009032
http://test.sca.oasisopen.org/

Abstract:

This document defines the Test Cases for the SCA C++ Client and Implementation Model specification.

The Test Cases represent a series of tests that an SCA implementation must pass in order to

claim conformance to the requirements of the SCA C++ Client and Implementation Model specification.

Status:

This document was last revised or approved by the Service Component Architecture / C and C++ TC on the above date. The level of approval is also listed above. Check the “Latest Version” or “Latest Approved Version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/sca-c-cpp/.

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-c-cpp/ipr.php).

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

Copyright © OASIS® 2010. All Rights Reserved.

Notices

Copyright © OASIS® 2010. All Rights Reserved.

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

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

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

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

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

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

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

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

Copyright © OASIS® 2010. All Rights Reserved.

1 Introduction

This document defines the Test Cases for the SCA Client and Implementation Model for C++ specification **[SCA-CPP]**.

The tests described in this document are related to the Test Assertions described in SCA Client and Implementation Model for C Test Assertions **[SCA-CPP-TA]**.

The test cases are structured in the same manner as the test cases for the SCA Assembly specification as described in the SCA Assembly test cases document **[SCA-TC]**.

1.1 Namespaces

The SCA C++ test case suite makes use of a set of XML namespaces.

1.1.1 SCA Artifact Namespaces

These apply to artifacts such as Composites

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

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

1.1.2 WSDL Namespace

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

1.2 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

1.3 Normative References

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[SCA-CPP] OASIS Committee Draft 06, *Service Component Architecture Client and Implementation Model for C++ Specification Version 1.1*, October 2010. http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-spec-cd06.pdf

[SCA-CPP-TA] OASIS Committee Draft 02, *Service Component Architecture Client and Implementation Model for C++ Test Assertions Version 1.1*, October 2010. http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions-cd02.pdf

[SCA-ASM-TA] OASIS Committee Draft 03, *Test Assertions for the SCA Assembly Model Version 1.1 Specification*, August 2010. http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-test-assertions-cd03.pdf

1.4 Non-Normative References

[SCA-TC] OASIS Committee Draft 03, *Testcases for the SCA Assembly Specification Version 1.1*, June 2010. http://docs.oasis-open.org/opencsa/sca-assembly/sca-assembly-1.1-testcases-cd03.pdf

Copyright © OASIS® 2010. All Rights Reserved.

2 Test Cases for Base Functionality

2.1 Assembly

Test Case ID	CPP_1001_TestCase
Test Assertion	ASM-TA-8001
Description	Tests that a C++ service which is marked remotable does not use method overloading.
Artifacts	CPP_1001_TestCase.java Test_CPP_1001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service3.h Service3Impl.h Service3Impl.cpp Service3Impl.componentType
Expected output	Negative test: “exception”

2.2 Section 2

Test Case ID	CPP_2001_TestCase
Test Assertion	CPP-TA-2001
Description	Tests that all functions of a service interface of a component are implemented by the component implementation.
Artifacts	CPP_2001_TestCase.java Test_CPP_2001.composite TestInvocation.h TestException.h TestException.cpp CPP_0002_Client.h CPP_0002_Client.cpp CPP_0002_Client.componentType Service1Superset.h Service1SupersetImpl2.h

Copyright © OASIS® 2010. All Rights Reserved.

	Service1SupersetImpl2.cpp Service1SupersetImpl2.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_2002_TestCase
Test Assertion	CPP-TA-2002
Description	Tests that a runtime supports composite scoped implementations.
Artifacts	CPP_2002_TestCase.java Test_CPP_2002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl11.h Service1Impl11.cpp Service1Impl11componentType Service1Impl12.h Service1Impl12.cpp Service1Impl12componentType
Expected output	Positive test: “CPP_2002 request service1 operation1 invoked service2 operation 1 invoked a service2 operation 1 invoked b”

Test Case ID	CPP_2003_TestCase
Test Assertion	CPP-TA-2002, CPP-TA-2004
Description	Tests that a runtime supports composite scoped implementations.
Artifacts	CPP_2002_TestCase.java Test_CPP_2002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h

	Service1Impl11.h Service1Impl11.cpp Service1Impl11componentType Service1Impl13.h Service1Impl13.cpp Service1Impl13componentType
Expected output	Positive test: “CPP_2003 request service1 operation1 invoked service2 operation 1 invoked a service2 operation 1 invoked a”

42

Test Case ID	CPP_2004_TestCase
Test Assertion	CPP-TA-2006
Description	Tests that a class named by an interface is present in the header file.
Artifacts	CPP_2004_TestCase.java Test_CPP_2004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Negative test: “exception”

43

Test Case ID	CPP_2005_TestCase
Test Assertion	CPP-TA-2007
Description	Tests that a callbackClass named by an interface is present in the header file.
Artifacts	CPP_2005_TestCase.java Test_CPP_2005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType

	Service1.h Service5.h Service5Callback.h Service1Callback5Impl.h Service1Callback5Impl.cpp Service1Callback5Impl.componentType Service5Impl.h Service5Impl.cpp Service5.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_2006_TestCase
Test Assertion	CPP-TA-2015
Description	Tests that functions excluded from a service interface cannot be invoked.
Artifacts	CPP_2006_TestCase.java Test_CPP_2006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11.h Service1Impl14.h Service1Impl14.cpp Service1Impl14.componentType Service11Impl.h Service11Impl.cpp Service11Impl.componentType
Expected output	Positive test: “CPP_2006 request Test service got an error during execution”

Test Case ID	CPP_2007_TestCase
Test Assertion	CPP-TA-2016
Description	Tests that functions excluded from a callback interface cannot be invoked.
Artifacts	CPP_2007_TestCase.java Test_CPP_2007.composite

	TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service12.h Service12Callback.h Service1Callback12Impl.h Service1Callback12Impl.cpp Service1Callback12Impl.componentType Service12Impl.h Service12Impl.cpp Service12.componentType
Expected output	Positive test: “CPP_2007 request service1 operation1 invoked Test service got an error during execution”

Test Case ID	CPP_2008_TestCase
Test Assertion	CPP-TA-2008
Description	Tests each function explicitly identified in a c++ interface is uniquely named.
Artifacts	CPP_2008_TestCase.java Test_CPP_2008.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11.h Service1Impl14.h Service1Impl14.cpp Service1Impl14.componentType Service11Impl.h Service11Impl.cpp Service11Impl.componentType
Expected output	Negative test: “exception”

47

Test Case ID	CPP_2009_TestCase
Test Assertion	CPP-TA-2009
Description	Tests that each function explicitly identified in a c++ callback interface is uniquely named.
Artifacts	CPP_2009_TestCase.java Test_CPP_2009.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service12.h Service12Callback.h Service1Callback12Impl.h Service1Callback12Impl.cpp Service1Callback12Impl.componentType Service12Impl.h Service12Impl.cpp Service12Impl.componentType
Expected output	Negative test: “exception”

48

Test Case ID	CPP_2010_TestCase
Test Assertion	CPP-TA-2011
Description	Tests each function explicitly identified in a c++ implementation is uniquely named.
Artifacts	CPP_2010_TestCase.java Test_CPP_2010.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11.h Service1Impl14.h Service1Impl14.cpp

	Service1Impl14.componentType Service11Impl.h Service11Impl.cpp Service11Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_2011_TestCase
Test Assertion	CPP-TA-2010
Description	Tests that a componentType file exists for every implementation class.
Artifacts	CPP_2011_TestCase.java Test_CPP_2011.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h
Expected output	Negative test: “exception”

Test Case ID	CPP_2012_TestCase
Test Assertion	CPP-TA-2012
Description	Tests that that every implementation class has a default constructor.
Artifacts	CPP_2012_TestCase.java Test_CPP_2006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl15.h Service1Impl15.cpp Service1Impl15.componentType
Expected output	Negative test:

	“exception”

Test Case ID	CPP_2013_TestCase
Test Assertion	CPP-TA-2014
Description	Tests that where one component is a client of a service provided by a second component, both with C++ implementations and which both run in the same address space, but the service implementation methods are not marked "allows pass by reference" that invocations of the service use "pass by value" semantics.
Artifacts	CPP_2013_TestCase.java Test_CPP_2013.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service4.h Service1Impl7.h Service1Impl7.cpp Service1Impl7.componentType Service4Impl.h Service4Impl.cpp Service4Impl.componentType
Expected output	Positive test: “CPP_2013 request service1 operation1 invoked service2 operation1 invoked request+1”

Test Case ID	CPP_2014_TestCase
Test Assertion	CPP-TA-2014
Description	Tests that where one component is a client of a service provided by a second component, both with C++ implementations and which both run in the same address space, with the service implementation methods are marked "allows pass by reference" but the client implementation is not marked "allows pass by reference" that invocations of the service use "pass by value" semantics.
Artifacts	CPP_2014_TestCase.java Test_CPP_2014.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h

	CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service4.h Service1Impl7.h Service1Impl7.cpp Service1Impl7.componentType Service4Impl.h Service4Impl.cpp Service4Impl.componentType
Expected output	Positive test: “CPP_2014 request service1 operation1 invoked service2 operation1 invoked request+1”

Test Case ID	CPP_2015_TestCase
Test Assertion	CPP-TA-2017
Description	Tests that a runtime does no synchronization of component invocations.
Artifacts	CPP_2015_TestCase.java Test_CPP_2015.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl56.h Service1Impl56.cpp Service1Impl56.componentType Service1Impl57.h Service1Impl57.cpp Service1Impl57.componentType
Expected output	Positive test: “CPP_2015 request service1 operation1 invoked service2 operation1 invoked b”

Test Case ID	CPP_2016_TestCase
Test Assertion	CPP-TA-2003
Description	Tests that a runtime creates a new instance of a stateless component for each invocation.

Artifacts	CPP_2016_TestCase.java Test_CPP_2016.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl56.h Service1Impl56.cpp Service1Impl56.componentType Service1Impl58.h Service1Impl58.cpp Service1Impl58.componentType
Expected output	Positive test: “CPP_2016 request service1 operation1 invoked service2 operation1 invoked 1”

2.3 Section 3

Test Case ID	CPP_3001_TestCase
Test Assertion	CPP-TA-3002
Description	Tests that all functions of a service interface are available via a proxy.
Artifacts	CPP_3001_TestCase.java Test_CPP_3001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1Superset.h Service1SupersetImpl2.h Service1SupersetImpl2.cpp Service1SupersetImpl2.componentType
Expected output	Positive test: “CPP_3001 request service1 operation2 invoked”

Test Ccase ID	CPP_3002_TestCase
Test Assertion	CPP-TA-3003, CPP-TA-3004

Copyright © OASIS® 2010. All Rights Reserved.

Description	Tests that a correct proxy is generated for an interface marked requires="asyncInvocation".
Artifacts	CPP_3002_TestCase.java Test_CPP_3002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service46.h Service1Impl53.h Service1Impl53.cpp Service1Impl53.componentType Service46Impl.h Service46Impl.cpp Service46Impl.componentType
Expected output	Positive test: “CPP_3002 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_3003_TestCase
Test Assertion	N/A – API Testing
Description	Tests that polling can be used for an interface marked requires="asyncInvocation".
Artifacts	CPP_3003_TestCase.java Test_CPP_3003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service46.h Service1Impl54.h Service1Impl54.cpp Service1Impl54.componentType Service46Impl.h Service46Impl.cpp Service46Impl.componentType

Expected output	Positive test: “CPP_3003 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_3004_TestCase
Test Assertion	N/A – API Testing
Description	Tests that an operation can be canceled for an interface marked requires="asyncInvocation".
Artifacts	CPP_3004_TestCase.java Test_CPP_3004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service46.h Service1Impl55.h Service1Impl55.cpp Service1Impl55.componentType Service46Impl.h Service46Impl.cpp Service46Impl.componentType
Expected output	Positive test: “CPP_3004 request service1 operation1 invoked”

2.4 Section 4

Test Case ID	CPP_4001_TestCase
Test Assertion	CPP-TA-4001
Description	Tests that the OneWay MEP is not rejected.
Artifacts	CPP_4001_TestCase.java Test_CPP_4001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h

	Service24OW.h Service24OW.wsdl Service1Impl29OW.h Service1Impl29OW.cpp Service1Impl29OW.componentType Service24OWImpl.h Service24OWImpl.cpp Service24OWImpl.componentType
Expected output	Positive test: “CPP_4001 request service1 operation1 invoked service2 operation1 invoked”

60

Test Case ID	CPP_4002_TestCase
Test Assertion	CPP-TA-4002
Description	Tests that all functions of a callback interface are available via a proxy.
Artifacts	CPP_4002_TestCase.java Test_CPP_4002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service12.h Service12Callback.h Service1Callback12Impl.h Service1Callback12Impl.cpp Service1Callback12Impl.componentType Service12Impl2.h Service12Impl2.cpp Service12Impl2.componentType
Expected output	Positive test: “CPP_4002 request service1 operation1 invoked service2 operation1 invoked service1 callback2 invoked”

61

Test Case ID	CPP_4003_TestCase
Test Assertion	CPP-TA-4003
Description	Tests that a callback proxy includes a member functions for invoking operations asynchronously.

Artifacts	CPP_4003_TestCase.java Test_CPP_4003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service47.h Service47Callback.h Service1Callback47Impl.h Service1Callback47Impl.cpp Service1Callback47Impl.componentType Service47Impl.h Service47Impl.cpp Service47Impl.componentType
Expected output	Positive test: "CPP_4003 request service1 operation1 invoked service2 operation1 invoked service1 callback1 invoked"

62
2.5 Section 6

Test Case ID	CPP_6001_TestCase
Test Assertion	N/A – API Testing
Description	Tests that all instances of a multiplicity 1..n reference are called.
Artifacts	CPP_6001_TestCase.java Test_CPP_6001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType Service1Impl3.h Service1Impl3.cpp Service1Impl3.componentType

Expected output	Positive test: “CPP_6001 request service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked service4 operation1 invoked”

Test Case ID	CPP_6002_TestCase
Test Assertion	N/A – API Testing
Description	Tests that a ServiceReference can be obtained and used to invoke a service operation.
Artifacts	CPP_6002_TestCase.java Test_CPP_6002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType Service1Impl18.h Service1Impl18.cpp Service1Impl18.componentType
Expected output	Positive test: “CPP_6002 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_6003_TestCase
Test Assertion	N/A – API Testing
Description	Tests that ServiceReferences for all instances of a multiplicity 1..n reference can be obtained and used to invoke a service operation.
Artifacts	CPP_6003_TestCase.java Test_CPP_6003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h

	Service1Impl.h Service1Impl.cpp Service1Impl.componentType Service1Impl19.h Service1Impl19.cpp Service1Impl19.componentType
Expected output	Positive test: “CPP_6003 request service1 operation1 invoked service2 operation1 invoked service3 operation1 invoked service4 operation1 invoked”

Test Case ID	CPP_6004_TestCase
Test Assertion	N/A – API Testing
Description	Tests that component can invoke its own services using getSelfRefrence().
Artifacts	CPP_6004_TestCase.java Test_CPP_6004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service16.h Service1_16Impl.h Service1_16Impl.cpp Service1_16Impl.componentType
Expected output	Positive test: “CPP_6004 request service1 operation1 invoked operation2 invoked”

Test Case ID	CPP_6005_TestCase
Test Assertion	N/A – API Testing
Description	Tests RefCountingPointer operations and casting.
Artifacts	CPP_6005_TestCase.java Test_CPP_6005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp

	CPP_0001_Client.componentType Service1.h Service1Impl20.h Service1Impl20.cpp Service1Impl20.componentType
Expected output	Positive test: “CPP_6005 request service1 operation1 invoked abc”

Test Case ID	CPP_6006_TestCase
Test Assertion	N/A – API Testing
Description	Tests getDataFactory.
Artifacts	CPP_6006_TestCase.java Test_CPP_6006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service40.h Service1Impl47.h Service1Impl47.cpp Service1Impl47.componentType Service40Impl.h Service40Impl.cpp Service40Impl.componentType
Expected output	Positive test: “CPP_6006 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_6007_TestCase
Test Assertion	N/A – API Testing
Description	Tests getURI.
Artifacts	CPP_6007_TestCase.java Test_CPP_6007.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h

Copyright © OASIS® 2010. All Rights Reserved.

	CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl22.h Service1Impl22.cpp Service1Impl22.componentType
Expected output	Positive test: “CPP_6007 request service1 operation1 invoked getURI getURI returned URI = TestComponent1”

69
2.6 Section 7

Test Case ID	CPP_7001_TestCase
Test Assertion	CPP-TA-7001
Description	Tests <export.cpp/> elements have unique names within a domain.
Artifacts	CPP_7001_TestCase.java Test_CPP_7001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP _0001_Client.cpp CPP _0001_Client.componentType Service1.h Service4.h Service1Impl7.h Service1Impl7.cpp Service1Impl7.componentType Service4Impl.h Service4Impl.cpp Service4Impl.componentType
Expected output	Negative test: “exception”

70

Test Case ID	CPP _7002_TestCase
Test Assertion	CPP -TA-7002
Description	Tests <import.cpp/> elements have unique names within a contribution.
Artifacts	CPP _7002_TestCase.java Test_CPP _7002.composite TestInvocation.h

	TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP _0001_Client.componentType Service1.h Service4.h Service1Impl7.h Service1Impl7.cpp Service1Impl7.componentType Service4Impl.h Service4Impl.cpp Service4Impl.componentType
Expected output	Negative test: “exception”

2.7 Section 8

Test Case ID	CPP_8001_TestCase
Test Assertion	CPP-TA-8001
Description	Tests that macros and typedefs are processed when testing interface compatibility.
Artifacts	CPP_8001_TestCase.java Test_CPP_8001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service15.h Service15Exp.h Service1Impl16.h Service1Impl16.cpp Service1Impl16.componentType Service15Impl.h Service15Impl.cpp Service15Impl.componentType
Expected output	Positive test: “CPP_8001 request service1 operation1 invoked service2 operation 1 invoked”

Copyright © OASIS® 2010. All Rights Reserved.

Test Case ID	CPP_8002_TestCase
Test Assertion	CPP-TA-8002
Description	Tests that UDTs are not allowed in remotable interfaces.
Artifacts	CPP_8002_TestCase.java Test_CPP_8002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service15.h Service1Impl17.h Service1Impl17.cpp Service1Impl17.componentType Service16Impl.h Service16Impl.cpp Service16Impl.componentType
Expected output	Negative test: “exception”

73

Test Case ID	CPP_8003_TestCase
Test Assertion	CPP-TA-8003
Description	Tests that a header file used to define a service interface contains a class declaration with public member functions.
Artifacts	CPP_8003_TestCase.java Test_CPP_8003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service13.h Service1Impl15.h Service1Impl15.cpp Service1Impl15.componentType Service13Impl.componentType

Expected output	Negative test: “exception”

Test Case ID	CPP_8004_TestCase
Test Assertion	CPP-TA-8004
Description	Tests that a class used to define a service interface contains no public member functions that are not pure virtual.
Artifacts	CPP_8004_TestCase.java Test_CPP_8004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service14.h Service1Impl21.h Service1Impl21.cpp Service1Impl21.componentType Service14Impl.h Service14Impl.cpp Service14Impl.componentType
Expected output	Negative test: “exception”

2.8 Section 9

Test Case ID	CPP_10001_TestCase
Test Assertion	CPP-TA-10007
Description	Tests the default mapping of C++ parameter passing styles to WSDL.
Artifacts	CPP_10001_TestCase.java Test_CPP_10001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType

	Service1.h Service19.h Service19.wsdl Service1Impl24.h Service1Impl24.cpp Service1Impl24.componentType Service19Impl.h Service19Impl.cpp Service19Impl.componentType
Expected output	Positive test: “CPP_10001 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_10002_TestCase
Test Assertion	CPP-TA-10010
Description	Tests the default mapping of C++ types to WSDL.
Artifacts	CPP_10002_TestCase.java Test_CPP_10002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service21.h Service21.wsdl Service1Impl26.h Service1Impl26.cpp Service1Impl26.componentType Service21Impl.h Service21Impl.cpp Service21Impl.componentType
Expected output	Positive test: “CPP_10002 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_10003_TestCase
Test Assertion	CPP-TA-10012
Description	Tests the core default mapping of an unannotated header to WSDL is correct.
Artifacts	CPP_10003_TestCase.java

	Test_CPP_10003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_ClientNA.h CPP_0001_ClientNA.cpp CPP_0001_ClientNA.componentType Service1.h Service1NA.h Service1.wsdl Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: “CPP_10003 request service1 operation1 invoked”

79

Test Case ID	CPP_10004_TestCase
Test Assertion	CPP-TA-10004
Description	Tests the mapping of unwrapped WSDL parameters to C++.
Artifacts	CPP_10004_TestCase.java Test_CPP_10004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated – Perform CPP_F006_Testcase prior to this one) Service19U.h (system under test generated) Service19U.wsdl Service1Impl24U.h Service1Impl24U.cpp Service1Impl24U.componentType Service19ImplU.h Service19ImplU.cpp Service19ImplU.componentType
Expected output	Positive test: “CPP_10004 request service1 operation1 invoked service2 operation 1 invoked”

80

Test Case ID	CPP_10005_TestCase
Test Assertion	CPP-TA-10005
Description	Tests the mapping of wrapped WSDL parameters to C++.
Artifacts	CPP_10005_TestCase.java Test_CPP_10005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated) Service19.h (system under test generated) Service19.wsdl Service1Impl24.h Service1Impl24.cpp Service1Impl24.componentType Service19Impl.h Service19Impl.cpp Service19Impl.componentType
Expected output	Positive test: “CPP_10005 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_10006_TestCase
Test Assertion	CPP-TA-10009
Description	Tests the mapping of WSDL types to C++.
Artifacts	CPP_10006_TestCase.java Test_CPP_10006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated) Service32.h (system under test generated) Service32.wsdl Service1Impl40.h Service1Impl40.cpp

	Service1Impl40.componentType Service32Impl.h Service32Impl.cpp Service32Impl.componentType
Expected output	Positive test: “CPP_10006 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_10007_TestCase
Test Assertion	CPP-TA-10003
Description	Tests the mapping of equivalent WSDL faults to C++.
Artifacts	CPP_10007_TestCase.java Test_CPP_10007.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated) Service32F.h (system under test generated) Service32F.wsdl Service1Impl40F.h Service1Impl40F.cpp Service1Impl40F.componentType Service32ImplF.h Service32ImplF.cpp Service32ImplF.componentType
Expected output	Positive test: “CPP_10007 request service1 operation1 invoked service2 operation 1 invoked”

2.9 Section 10

Test Case ID	CPP_11001_TestCase
Test Assertion	CPP-TA-11001
Description	Tests that a composite containing incorrect interface.cpp element is rejected.
Artifacts	CPP_11001_TestCase.java Test_CPP_11001.composite TestInvocation.h TestException.h

	TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11.h Service1Impl14.h Service1Impl14.cpp Service1Impl14.componentType Service11Impl.h Service11Impl.cpp Service11Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_11002_TestCase
Test Assertion	CPP-TA-11002
Description	Tests that a componentType containing incorrect interface.cpp elements is rejected.
Artifacts	CPP_11002_TestCase.java Test_CPP_11002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11.h Service1Impl14.h Service1Impl14.cpp Service1Impl14.componentType Service11Impl.h Service11Impl.cpp Service11Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_11004_TestCase
Test Assertion	CPP-TA-11001

Description	Tests that a composite containing incorrect implementation.cpp attribute values is rejected.
Artifacts	CPP_11004_TestCase.java Test_CPP_11004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl11.h Service1Impl11.cpp Service1Impl11.componentType Service1Impl12.h Service1Impl12.cpp Service1Impl12.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_11005_TestCase
Test Assertion	CPP-TA-11004
Description	Tests that a contribution containing incorrect import.cpp attributes is rejected.
Artifacts	CPP_11005_TestCase.java
Expected output	Negative test: “exception”

Test Case ID	CPP_11006_TestCase
Test Assertion	CPP-TA-11005
Description	Tests that a WSDL containing incorrect wsdl extension elements is rejected.
Artifacts	CPP_11006_TestCase.java Test_CPP_11006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType

	Service1.h Service18.h Service18.wsdl Service1Impl23.h Service1Impl23.cpp Service1Impl23.componentType Service18Impl.h Service18Impl.cpp Service18Impl.componentType
Expected output	Negative test: “exception”

2.10 Appendix F

Test Case ID	CPP_F001_TestCase
Test Assertion	CPP-TA-C001, CPP-TA-F026, CPP-TA-F030, CPP-TA-F033, CPP-TA-F038, CPP-TA-F039, CPP-TA-F044, CPP-TA-F045, CPP-TA-F046,
Description	Tests that the core default mapping of C++ to WSDL is correct.
Artifacts	CPP_F001_TestCase.java Test_CPP_F001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.c CPP_0001_Client.componentType Service1.wsdl Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: “CPP_F001 request service1 operation1 invoked”

Test Case ID	CPP_F002_TestCase
Test Assertion	CPP-TA-F037
Description	Tests the mapping of C++ unnamed parameters to WSDL.
Artifacts	CPP_F002_TestCase.java Test_CPP_F002.composite TestInvocation.h

Copyright © OASIS® 2010. All Rights Reserved.

	TestException.h TestException.cpp CPP_0001_Client.c CPP_0001_Client.componentType Service1.h Service18.h Service18.wsdl Service1Impl23.h Service1Impl23.cpp Service1Impl23.componentType Service18Impl.h Service18Impl.cpp Service18Impl.componentType
Expected output	Positive test: “CPP_F002 request service1 operation1 invoked service2 operation 1 invoked”

91

Test Case ID	CPP_F003_TestCase
Test Assertion	CPP-TA-F027
Description	Tests the WSDL mapping of C names containing "_x" and starting with "xml".
Artifacts	CPP_F003_TestCase.java Test_CPP_F003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.c CPP_0001_Client.componentType Service1.h Service23.h Service23.wsdl Service1Impl28.h Service1Impl28.cpp Service1Impl28.componentType Service23Impl.h Service23Impl.cpp Service23Impl.componentType
Expected output	Positive test: “CPP_F003 request service1 operation1 invoked service2 operation 1 invoked”

92

Test Case ID	CPP_F004_TestCase

Test Assertion	CPP-TA-F004
Description	Tests the WSDL mapping of C++ member functions with only in parameters and a void return type.
Artifacts	CPP_F004_TestCase.java Test_CPP_F004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service24.h Service24.wsdl Service1Impl29.h Service1Impl29.cpp Service1Impl29.componentType Service24Impl.h Service24Impl.cpp Service24Impl.componentType
Expected output	Positive test: “CPP_F004 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_F005_TestCase
Test Assertion	CPP-TA-F005
Description	Tests the mapping of inherited C++ classes to WSDL.
Artifacts	CPP_F005_TestCase.java Test_CPP_F005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service22.h Service22Super.h Service22.wsdl Service1Impl27.h Service1Impl27.cpp

Copyright © OASIS® 2010. All Rights Reserved.

	Service1Impl27.componentType Service22Impl.h Service22Impl.cpp Service22Impl.componentType
Expected output	Positive test: "CPP_F005 request service1 operation1 invoked service2 operation1 invoked"

94

Test Case ID	CPP_F006_TestCase
Test Assertion	CPP-TA-10001, CPP-TA-10011, CPP-TA-F001, CPP-TA-F002, CPP-TA-F005, CPP-TA-F006, CPP-TA-F007, CPP-TA-F008, CPP-TA-F009, CPP-TA-F012, CPP-TA-F013, CPP-TA-F017, CPP-TA-F022
Description	Tests the core default mapping of WSDL to C++.
Artifacts	CPP_F006_TestCase.java Test_CPP_F006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service1.h (system under test generated) Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: "CPP_F006 request service1 operation1 invoked"

95

Test Case ID	CPP_F007_TestCase
Test Assertion	CPP-TA-F014, CPP-TA-F015
Description	Tests the mapping of unwrapped WSDL parameters to C++.
Artifacts	CPP_F007_TestCase.java Test_CPP_F007.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType

	Service1U.wsdl Service1U.h (system under test generated) Service1ImplU.h Service1ImplU.cpp Service1ImplU.componentType
Expected output	Positive test: “CPP_F007 request service1 operation1 invoked”

96

Test Case ID	CPP_F008_TestCase
Test Assertion	CPP-TA-F019, CPP-TA-F020
Description	Tests the mapping of WSDL faults to C++.
Artifacts	CPP_F008_TestCase.java Test_CPP_F008.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated) Service31.h (system under test generated) OperationFault.h (system under test generated) OperationFault.cpp Service31.wsdl Service1Impl39.h Service1Impl39.cpp Service1Impl39.componentType Service31Impl.h Service31Impl.cpp Service31Impl.componentType
Expected output	Positive test: “CPP_F008 request service1 operation1 invoked service2 operation1 invoked”

97

Test Case ID	CPP_F009_TestCase
Test Assertion	CPP-TA-F036
Description	Tests the WSDL mapping of C names containing "_x" and starting with "xml".
Artifacts	CPP_F009_TestCase.java Test_CPP_F009.composite TestInvocation.h

	TestException.h TestException.cpp CPP_0001_Client.c CPP_0001_Client.componentType Service1.h Service23.h Service23.wsdl Service1Impl28.h Service1Impl28.cpp Service1Impl28.componentType Service23ImplOW.h Service23ImplOW.cpp Service23ImplOW.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_F010_TestCase
Test Assertion	CPP-TA-F018
Description	Tests that parameter name clashes are detected.
Artifacts	CPP_F010_TestCase.java Test_CPP_F010.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.c CPP_0001_Client.componentType Service1.h Service20.h (system under test generated – should fail) Service20.wsdl Service1Impl25.h Service1Impl25.cpp Service1Impl25.componentType Service20Impl.h Service20Impl.cpp Service20Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_F012_TestCase

Test Assertion	CPP-TA-F011, CPP-TA-F023
Description	Tests the mapping of a SOAP binding to C++.
Artifacts	CPP_F012_TestCase.java Test_CPP_F012.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated) Service36.h (system under test generated) Service36.wsdl Service1Impl43.h Service1Impl43.cpp Service1Impl43.componentType Service36Impl.h Service36Impl.cpp Service36Impl.componentType
Expected output	Positive test: “CPP_F012 request service1 operation1 invoked service2 operation1 invoked”

100

Test Case ID	CPP_F013_TestCase
Test Assertion	CPP-TA-F010
Description	Tests the mapping or OneWay WSDL operations to C++.
Artifacts	CPP_F013_TestCase.java Test_CPP_F013.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h (previously system under test generated) Service24OW.h (system under test generated) Service24OW.wsdl Service1Impl29OW.h Service1Impl29OW.cpp Service1Impl29OW.componentType Service24OWImpl.h

	Service24OWImpl.cpp Service24OWImpl.componentType
Expected output	Positive test: “CPP_F013 request service1 operation1 invoked service2 operation1 invoked”

3 Test Cases for Annotation Support

The test cases in this section apply to the optional C++ SCA annotation support and are only required if an implementation supports these annotations.

3.1 Appendix A

Test Case ID	CPP_A001_TestCase
Test Assertion	CPP-TA-A001
Description	Tests that @Interface and @Property generate correct SCDL.
Artifacts	CPP_A001_TestCase.java Test_CPP_A001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service15A.h Service15Exp.h Service1Impl16.h Service1Impl16.cpp Service1Impl16.componentType Service15ImplA.h Service15ImplA.cpp Service15ImplA.componentType (system under test generated)
Expected output	Positive test: "CPP_A001 request service1 operation1 invoked service2 operation 1 invoked"

Test Case ID	CPP_A002_TestCase
Test Assertion	CPP-TA-A001
Description	Tests that @Remotable generates correct SCDL.
Artifacts	CPP_A002_TestCase.java Test_CPP_A002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h

	CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl2.h Service1Impl2.cpp Service1Impl2.componentType (system under test generated) Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: “CPP_A002 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_A003_TestCase
Test Assertion	CPP-TA-A001, CPP-TA-A003
Description	Tests that that @Callback generates correct SCDL.
Artifacts	CPP_A003_TestCase.java Test_CPP_A003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service5.h Service5Callback.h Service1Callback5Impl.h Service1Callback5Impl.cpp Service1Callback5Impl.componentType Service5Impl.h Service5Impl.cpp Service5.componentType (system under test generated)
Expected output	Positive test: “CPP_A003 request service1 operation1 invoked service2 operation1 invoked service1 callback2 invoked”

Test Case ID	CPP_A004_TestCase
Test Assertion	CPP-TA-A001. CPP-TA-C001
Description	Tests that that @OneWay generates correct SCDL.

Artifacts	CPP_A004_TestCase.java Test_CPP_A004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service24OW.h Service24OW.wsdl Service1Impl29OW.h Service1Impl29OW.cpp Service1Impl29OW.componentType Service24OWImpl.h Service24OWImpl.cpp Service24OWImpl.componentType (system under test generated)
Expected output	Positive test: “CPP_A004 request service1 operation1 invoked service2 operation1 invoked”

109

Test Case ID	CPP_A005_TestCase
Test Assertion	CPP-TA-A001, CPP-TA-C010
Description	Tests that that @Function generates correct SCDL.
Artifacts	CPP_A005_TestCase.java Test_CPP_A005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11A.h Service11A.wsdl Service1Impl14A.h Service1Impl14A.cpp Service1Impl14A.componentType (system under test generated) Service11ImplA.h Service11ImplA.cpp Service11ImplA.componentType (system under test generated)

Expected output	Positive test: “CPP_A005 request Test service got an error during execution”

110

Test Case ID	CPP_A006_TestCase
Test Assertion	CPP-TA-A001
Description	Tests that @Reference generates correct SCDL.
Artifacts	CPP_A006_TestCase.java Test_CPP_A006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl30.h Service1Impl30.cpp Service1Impl30.componentType (system under test generated) Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: “CPP_A006 request service1 operation1 invoked service2 operation 1 invoked”

111

Test Case ID	CPP_A007_TestCase
Test Assertion	CPP-TA-2002
Description	Tests that a runtime supports composite scoped implementations.
Artifacts	CPP_A007_TestCase.java Test_CPP_A007.composite TestCompositeC1.composite (system under test generated) • Contains 1 composite scoped component implemented by Service1Impl12 • Provides 1 service (Service1) which promotes the Service1 of the component • Has one property (ServiceName, type string) that sets ServiceName of the component TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp

	CPP_0001_Client.componentType Service1.h Service1Impl11.h Service1Impl11.cpp Service1Impl11componentType Service1Impl12.h Service1Impl12.cpp Service1Impl12componentType (system under test generated)
Expected output	Positive test: “CPP_A007 request service1 operation1 invoked service2 operation 1 invoked a service2 operation 1 invoked b”

Test Case ID	CPP_A008_TestCase
Test Assertion	CPP-TA-A002
Description	Tests that annotations are ignored at runtime.
Artifacts	CPP_A008_TestCase.java Test_CPP_A008.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Impl2.h Service1Impl2.cpp Service1Impl2.componentType Service5ImplA.h Service5ImplA.cpp Service5ImplA.componentType
Expected output	Positive test: “CPP_A008 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_A009_TestCase
Test Assertion	CPP-TA-A002
Description	Tests that @WebService generates same SCDL as equivalent, but missing @Remotable.
Artifacts	CPP_A009_TestCase.java Test_CPP_A009.composite

	TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service45.h Service1Impl52.h Service1Impl52.cpp Service1Impl52.componentType Service45Impl.h Service45Impl.cpp Service45Impl.componentType
Expected output	Positive test: “CPP_A009 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_A010_TestCase
Test Assertion	CPP-TA-A005, CPP-TA-C001, CPP-TA-F034
Description	Tests that @WebFunction generates same SCDL as equivalent, but missing @Function.
Artifacts	CPP_A010_TestCase.java Test_CPP_A010.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service11A.wsdl Service11W.h Service1Impl14W.h Service1Impl14W.cpp Service1Impl14W.componentType (system under test generated) Service11ImplW.h Service11ImplW.cpp Service11ImplW.componentType (system under test generated)
Expected output	Positive test: “CPP_A010 request Test service got an error during execution”

3.2 Appendix B

Test Case ID	CPP_B001_TestCase
Test Assertion	CPP-TA-A001
Description	Tests that @Requires generates correct SCDL.
Artifacts	CPP_B001_TestCase.java Test_CPP_B001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Requires.h Service1Impl2.h Service1Impl2.cpp Service1Impl2.componentType Service1ImplRequires.h Service1ImplRequires.cpp Service1ImplReqires.componentType (system under test generated)
Expected output	Positive test: “CPP_B001 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_B002_TestCase
Test Assertion	CPP-TA-A001
Description	Tests that @Requires generates correct SCDL.
Artifacts	CPP_B002_TestCase.java Test_CPP_B002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service1Intent.h Service1Impl2.h

Copyright © OASIS® 2010. All Rights Reserved.

	Service1Impl2.cpp Service1Impl2.componentType Service1ImplIntent.h Service1ImplIntent.cpp Service1ImplIntent.componentType (system under test generated)
Expected output	Positive test: “CPP_B002 request service1 operation1 invoked service2 operation 1 invoked”

3.3 Appendix C

Test Case ID	CPP_C001_TestCase
Test Assertion	CPP-TA-C001, CPP-TA-F047
Description	Tests that @WebService generates correct WSDL.
Artifacts	CPP_C001_TestCase.java Test_CPP_C001.composite TestInvocation.h TestInvocation.wsdl TestException.h TestException.cpp CPP_0005_Client.h CPP_0005_Client.cpp CPP_0005_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: “CPP_C001 request service1 operation1 invoked”

Test Case ID	CPP_C002_TestCase
Test Assertion	CPP-TA-C001
Description	Tests that @WebParam generates correct WSDL.
Artifacts	CPP_C002_TestCase.java Test_CPP_C002.composite TestInvocation.h TestInvocation.wsdl TestException.h TestException.cpp CPP_0005_Client.h CPP_0005_Client.cpp

	CPP_0005_Client.componentType Service1.h Service25.h Service25.wsdl Service1Impl33.h Service1Impl33.cpp Service1Impl33.componentType Service25Impl.h Service25Impl.cpp Service25Impl.componentType
Expected output	Positive test: “CPP_C002 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_C003_TestCase
Test Assertion	CPP-TA-C001
Description	Tests that @WebResult generates correct WSDL.
Artifacts	CPP_C003_TestCase.java Test_CPP_C003.composite TestInvocation.h TestInvocation.wsdl TestException.h TestException.cpp CPP_0005_Client.h CPP_0005_Client.cpp CPP_0005_Client.componentType Service1.h Service26.h Service26.wsdl Service1Impl34.h Service1Impl34.cpp Service1Impl34.componentType Service26Impl.h Service26Impl.cpp Service26Impl.componentType
Expected output	Positive test: “CPP_C003 request service1 operation1 invoked service2 operation 1 invoked”

Test Case ID	CPP_C004_TestCase
Test Assertion	CPP-TA-C002

Description	Tests that @WebService results in a remotable SCDL interface definition.
Artifacts	CPP_C004_TestCase.java Test_CPP_C004.composite TestInvocation.h TestInvocation.wsdl TestException.h TestException.cpp CPP_0006_Client.h CPP_0006_Client.cpp CPP_0006_Client.componentType (system under test generated) Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Positive test: “CPP_C004 request service1 operation1 invoked”

122

Test Case ID	CPP_C005_TestCase
Test Assertion	CPP-TA-C004
Description	Tests that @WebParam generates correct WSDL.
Artifacts	CPP_C005_TestCase.java Test_CPP_C005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service30.h Service1Impl38.h Service1Impl38.cpp Service1Impl38.componentType Service30Impl.h Service30Impl.cpp Service30Impl.componentType
Expected output	Negative test: “exception”

123

Copyright © OASIS® 2010. All Rights Reserved.

Test Case ID	CPP_C006_TestCase
Test Assertion	CPP-TA-C005
Description	Tests that @WebParam type errors are detected.
Artifacts	CPP_C006_TestCase.java Test_CPP_C006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service27.h Service1Impl35.h Service1Impl35.cpp Service1Impl35.componentType Service27Impl.h Service27Impl.cpp Service27Impl.componentType
Expected output	Negative test: “exception”

124

Test Case ID	CPP_C007_TestCase
Test Assertion	CPP-TA-C006
Description	Tests that @WebParam type errors are detected.
Artifacts	CPP_C007_TestCase.java Test_CPP_C007.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service29.h Service1Impl37.h Service1Impl37.cpp Service1Impl37.componentType Service29Impl.h

	Service29Impl.cpp Service29Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_C008_TestCase
Test Assertion	CPP-TA-C009
Description	Tests that a class that is listed in a @WebThrows annotation has a @WebFault annotation.
Artifacts	CPP_C008_TestCase.java Test_CPP_C008.composite TestInvocation1.h TestException1.h TestException1.cpp CPP_0007_Client.h CPP_0007_Client.cpp CPP_0007_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_C009_TestCase
Test Assertion	CPP-TA-C007
Description	Tests that a class with a @WebFault annotation has the correct constructor.
Artifacts	CPP_C009_TestCase.java Test_CPP_C009.composite TestInvocation2.h TestException2.h TestException2.cpp CPP_0008_Client.h CPP_0008_Client.cpp CPP_0008_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType

Copyright © OASIS® 2010. All Rights Reserved.

Expected output	Negative test: “exception”

Test Case ID	CPP_C010_TestCase
Test Assertion	CPP-TA-C008
Description	Tests that a class with a @WebFault annotation has a getFaultInfo member function.
Artifacts	CPP_C010_TestCase.java Test_CPP_C010.composite TestInvocation3.h TestException3.h TestException3.cpp CPP_0009_Client.h CPP_0009_Client.cpp CPP_0009_Client.componentType Service1.h Service1Impl.h Service1Impl.cpp Service1Impl.componentType
Expected output	Negative test: “exception”

3.4 Appendix F

Test Case ID	CPP_F014_TestCase
Test Assertion	CPP-TA-F028
Description	Tests that overloaded functions can be disambiguated using @WebFunction.
Artifacts	CPP_F014_TestCase.java Test_CPP_F014.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.h Service28.h Service28.wsdl Service1Impl36.h Service1Impl36.cpp

	Service1Impl36.componentType Service28Impl.h Service28Impl.cpp Service28Impl.componentType
Expected output	Positive test: “CPP_F014 request service1 operation1 invoked service2 operation1 invoked”

3.5 Assembly

Test Case ID	CPP_1002_TestCase
Test Assertion	ASM-TA-8005
Description	Tests that the callback interface of a @Remotable interface is also @Remotable.
Artifacts	CPP_1002_TestCase.java Test_CPP_1002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service3.h Service1.h Service6.h Service6Callback.h Service1Callback6Impl.h Service1Callback6Impl.cpp Service1Callback6Impl.componentType (system under test generated) Service6Impl.h Service6Impl.cpp Service6Impl.componentType (system under test generated)
Expected output	Negative test: “exception”

Test Case ID	CPP_1003_TestCase
Test Assertion	ASM-TA-8006
Description	Tests that the callback interface of a local interface is not @Remotable.
Artifacts	CPP_1003_TestCase.java Test_CPP_1003.composite TestInvocation.h TestException.h

	TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service3.h Service1.h Service7.h Service7Callback.h Service1Callback7Impl.h Service1Callback7Impl.cpp Service1Callback7Impl.componentType (system under test generated) Service7Impl.h Service7Impl.cpp Service7Impl.componentType (system under test generated)
Expected output	Negative test: “exception”

4 Test Cases for WSDL Extension Support

The test cases in this section apply to the optional C++ WSDL annotation support and are only required if an implementation supports these annotations.

4.1 Appendix D

Test Case ID	CPP_D001_TestCase
Test Assertion	CPP-TA-D001, CPP-TA-F003
Description	Tests the mapping of a memberFunction WSDL extension.
Artifacts	CPP_D001_TestCase.java Test_CPP_D001.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service37.h (system under test generated) Service37.wsdl Service1Impl44.h Service1Impl44.cpp Service1Impl45.componentType Service37Impl.h Service37Impl.cpp Service37Impl.componentType
Expected output	Positive test: "CPP_D001 request service1 operation1 invoked service2 operation1 invoked"

Test Case ID	CPP_D002_TestCase
Test Assertion	CPP-TA-D001, CPP-TA-F003
Description	Tests the mapping of a parameter WSDL extension.
Artifacts	CPP_D002_TestCase.java Test_CPP_D002.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h

	CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service38.h (system under test generated) Service38.wsdl Service1Impl45.h Service1Impl45.cpp Service1Impl45.componentType Service38Impl.h Service38Impl.cpp Service38Impl.componentType
Expected output	Positive test: “CPP_D002 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_D003_TestCase
Test Assertion	CPP-TA-D001, CPP-TA-F003
Description	Tests the mapping of a class WSDL extension.
Artifacts	CPP_D003_TestCase.java Test_CPP_D003.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service39.h (system under test generated) Service39.wsdl Service1Impl46.h Service1Impl46.cpp Service1Impl46.componentType Service39Impl.h Service39Impl.cpp Service39Impl.componentType
Expected output	Positive test: “CPP_D003 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_D004_TestCase
Test Assertion	CPP-TA-D001, CPP-TA-F003, CPP-TA-F016

Description	Tests the mapping of an enableWrapperStyle WSDL extension.
Artifacts	CPP_D004_TestCase.java Test_CPP_D004.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service40.h (system under test generated) Service40.wsdl Service1Impl47.h Service1Impl47.cpp Service1Impl47.componentType Service40Impl.h Service40Impl.cpp Service40Impl.componentType
Expected output	Positive test: “CPP_D004 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_D005_TestCase
Test Assertion	CPP-TA-D001, CPP-TA-F003
Description	Tests the mapping of a namespace WSDL extension.
Artifacts	CPP_D005_TestCase.java Test_CPP_D005.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service35.h (system under test generated) Service35.wsdl Service1Impl42.h Service1Impl42.cpp Service1Impl42.componentType Service35Impl.h Service35Impl.cpp

Copyright © OASIS® 2010. All Rights Reserved.

	Service35Impl.componentType
Expected output	Positive test: “CPP_D005 request service1 operation1 invoked service2 operation1 invoked”

Test Case ID	CPP_D006_TestCase
Test Assertion	CPP-TA-D002
Description	Tests that a second class WSDL extension in one binding is detected.
Artifacts	CPP_D006_TestCase.java Test_CPP_D006.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service39.h (previously system under test generated) Service39Bad.wsdl Service1Impl46.h Service1Impl46.cpp Service1Impl46.componentType Service39Impl.h Service39Impl.cpp Service39Impl.componentType
Expected output	Negative test: “exception”

Test Case ID	CPP_D007_TestCase
Test Assertion	CPP-TA-D003
Description	Tests that a second enableWrapperStyle WSDL extension in one binding is detected.
Artifacts	CPP_D007_TestCase.java Test_CPP_D007.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType

	Service1.wsdl Service40.h (previously system under test generated) Service40Bad.wsdl Service1Impl47.h Service1Impl47.cpp Service1Impl47.componentType Service40Impl.h Service40Impl.cpp Service40Impl.componentType
Expected output	Negative test: “exception”

142

Test Case ID	CPP_D008_TestCase
Test Assertion	CPP-TA-D004
Description	Tests that a second namespace WSDL extension in one binding is detected.
Artifacts	CPP_D008_TestCase.java Test_CPP_D008.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service35.h (previously system under test generated) Service35Bad.wsdl Service1Impl42.h Service1Impl42.cpp Service1Impl42.componentType Service35Impl.h Service35Impl.cpp Service35Impl.componentType
Expected output	Negative test: “exception”

143

Test Case ID	CPP_D009_TestCase
Test Assertion	CPP-TA-D005
Description	Tests that a second memberFunction WSDL extension in one binding is detected.
Artifacts	CPP_D009_TestCase.java

	Test_CPP_D009.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service37.h (previously system under test generated) Service37Bad.wsdl Service1Impl44.h Service1Impl44.cpp Service1Impl45.componentType Service37Impl.h Service37Impl.cpp Service37Impl.componentType
Expected output	Negative test: “exception”

144

Test Case ID	CPP_D010_TestCase
Test Assertion	CPP-TA-D006
Description	Tests the type mapping of a parameter WSDL extension.
Artifacts	CPP_D010_TestCase.java Test_CPP_D010.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service42.h (system under test generated) Service42.wsdl Service1Impl49.h Service1Impl49.cpp Service1Impl49.componentType Service42Impl.h Service42Impl.cpp Service42Impl.componentType
Expected output	Positive test:

	“CPP_D010 request service1 operation1 invoked service2 operation1 invoked”

145

Test Case ID	CPP_D011_TestCase
Test Assertion	CPP-TA-D007
Description	Tests the mapping of a JAX-WS WSDL extension.
Artifacts	CPP_D011_TestCase.java Test_CPP_D011.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service37.h (system under test generated) Service37J.wsdl Service1Impl44.h Service1Impl44.cpp Service1Impl45.componentType Service37Impl.h Service37Impl.cpp Service37Impl.componentType
Expected output	Positive test: “CPP_D011 request service1 operation1 invoked service2 operation1 invoked”

146

Test Case ID	CPP_D012_TestCase
Test Assertion	CPP-TA-D008
Description	Tests that a C++ WSDL extension overrides a JAX-WS WSDL extension.
Artifacts	CPP_D012_TestCase.java Test_CPP_D012.composite TestInvocation.h TestException.h TestException.cpp CPP_0001_Client.h CPP_0001_Client.cpp CPP_0001_Client.componentType Service1.wsdl Service37.h (system under test generated) Service37JC.wsdl

	Service1Impl44.h Service1Impl44.cpp Service1Impl45.componentType Service37Impl.h Service37Impl.cpp Service37Impl.componentType
Expected output	Positive test: “CPP_D012 request service1 operation1 invoked service2 operation1 invoked”

147

5 Cross Mapping of Test Assertions to Test Cases

Test Assertion	Test Cases
ASM-TA-8001	CPP_1001_TestCase
ASM-TA-8005	CPP_1002_TestCase
ASM-TA-8006	CPP_1003_TestCase

148

Test Assertion	Test Cases
CPP-TA-2001	CPP_2001_TestCase
CPP-TA-2002	CPP_2002_TestCase CPP_2003_TestCase
CPP-TA-2003	CPP_2016_TestCase
CPP-TA-2004	CPP_2003_TestCase
CPP-TA-2005	Optional – Not Tested
CPP-TA-2006	CPP_2004_TestCase
CPP-TA-2007	CPP_2005_TestCase
CPP-TA-2008	CPP_2008_TestCase
CPP-TA-2009	CPP_2009_TestCase
CPP-TA-2010	CPP_2011_TestCase
CPP-TA-2011	CPP_2010_TestCase
CPP-TA-2012	CPP_2012_TestCase
CPP-TA-2013	Optional – Not Tested
CPP-TA-2014	CPP_2013_TestCase CPP_2014_TestCase
CPP-TA-2015	CPP_2006_TestCase
CPP-TA-2016	CPP_2007_TestCase
CPP-TA-2017	CPP_2015_TestCase

149

Test Assertion	Test Cases
CPP-TA-3001	Not runtime testable
CPP-TA-3002	CPP_3001_TestCase
CPP-TA-3003	CPP_3002_TestCase

CPP-TA-3004	CPP_3002_TestCase

Test Assertion	Test Cases
CPP-TA-4001	CPP_4001_TestCase
CPP-TA-4002	CPP_4002_TestCase
CPP-TA-4003	CPP_4003_TestCase

Test Assertion	Test Cases
CPP-TA-7001	CPP_7001_TestCase
CPP-TA-7002	CPP_7002_TestCase

Test Assertion	Test Cases
CPP-TA-8001	CPP_8001_TestCase
CPP-TA-8002	CPP_8002_TestCase
CPP-TA-8003	CPP_8003_TestCase
CPP-TA-8004	CPP_8004_TestCase

Test Assertion	Test Cases
CPP-TA-10001	CPP_F006_TestCase
CPP-TA-10002	Optional – Not tested
CPP-TA-10003	CPP_10007_TestCase
CPP-TA-10004	CPP_10004_TestCase
CPP-TA-10005	CPP_10005_TestCase
CPP-TA-10006	Untestable – Not a runtime behavior
CPP-TA-10007	CPP_10001_TestCase
CPP-TA-10009	CPP_10006_TestCase
CPP-TA-10010	CPP_10002_TestCase
CPP-TA-10011	CPP_F006_TestCase
CPP-TA-10012	CPP_10003_TestCase

Test Assertion	Test Cases
CPP-TA-11001	CPP_11001_TestCase CPP_11004_TestCase

CPP-TA-11002	CPP_11002_TestCase
CPP-TA-11004	CPP_11005_TestCase
CPP-TA-11005	CPP_11006_TestCase

Test Assertion	Test Cases
CPP-TA-A001	CPP_A001_TestCase CPP_A002_TestCase CPP_A003_TestCase CPP_A004_TestCase CPP_A005_TestCase CPP_A006_TestCase CPP_A007_TestCase CPP_B001_TestCase CPP_B002_TestCase
CPP-TA-A002	CPP_A008_TestCase
CPP-TA-A003	CPP_A003_TestCase
CPP-TA-A004	CPP_A009_TestCase
CPP-TA-A005	CPP_A010_TestCase

Test Assertion	Test Cases
CPP-TA-C001	CPP_F001_TestCase CPP_A004_TestCase CPP_A010_TestCase CPP_C001_TestCase CPP_C002_TestCase CPP_C003_TestCase
CPP-TA-C002	CPP_C004_TestCase
CPP-TA-C004	CPP_C005_TestCase
CPP-TA-C005	CPP_C006_TestCase
CPP-TA-C006	CPP_C007_TestCase
CPP-TA-C007	CPP_C009_TestCase
CPP-TA-C008	CPP_C010_TestCase
CPP-TA-C009	CPP_C008_TestCase
CPP-TA-C010	CPP_A005_TestCase

Test Assertion	Test Cases
CPP-TA-D001	CPP_D001_TestCase CPP_D002_TestCase CPP_D003_TestCase CPP_D004_TestCase CPP_D005_TestCase
CPP-TA-D002	CPP_D006_TestCase
CPP-TA-D003	CPP_D007_TestCase
CPP-TA-D004	CPP_D008_TestCase
CPP-TA-D005	CPP_D009_TestCase
CPP-TA-D006	CPP_D010_TestCase
CPP-TA-D007	CPP_D011_TestCase
CPP-TA-D008	CPP_D012_TestCase

158

Test Assertion	Test Cases
CPP-TA-F001	CPP_F006_TestCase
CPP-TA-F002	CPP_F006_TestCase
CPP-TA-F003	CPP_D001_TestCase CPP_D002_TestCase CPP_D003_TestCase CPP_D004_TestCase CPP_D005_TestCase
CPP-TA-F004	Not tested – this is base WSDL handling, not SCA specific.
CPP-TA-F005	CPP_F006_TestCase
CPP-TA-F006	CPP_F006_TestCase
CPP-TA-F007	CPP_F006_TestCase
CPP-TA-F008	CPP_F006_TestCase
CPP-TA-F009	CPP_F006_TestCase
CPP-TA-F010	CPP_F013_TestCase
CPP-TA-F011	CPP_F012_TestCase
CPP-TA-F012	CPP_F006_TestCase
CPP-TA-F013	CPP_F006_TestCase
CPP-TA-F014	CPP_F007_TestCase
CPP-TA-F015	CPP_F007_TestCase

CPP-TA-F016	CPP_D004_TestCase
CPP-TA-F017	CPP_F006_TestCase
CPP-TA-F018	CPP_F010_TestCase
CPP-TA-F019	CPP_F008_TestCase
CPP-TA-F020	CPP_F008_TestCase
CPP-TA-F022	CPP_F006_TestCase
CPP-TA-F023	CPP_F012_TestCase
CPP-TA-F024	Not tested – This is an assertion about the web services binding behavior
CPP-TA-F025	Not tested – not all platforms allow application access to header content
CPP-TA-F026	CPP_F001_TestCase
CPP-TA-F027	CPP_F003_TestCase
CPP-TA-F028	CPP_F014_TestCase
CPP-TA-F029	Not Tested
CPP-TA-F030	CPP_F001_TestCase
CPP-TA-F031	CPP_F005_TestCase
CPP-TA-F032	Optional – Not tested
CPP-TA-F033	CPP_F001_TestCase
CPP-TA-F034	CPP_A010_TestCase
CPP-TA-F035	CPP_F004_TestCase
CPP-TA-F036	CPP_F009_TestCase
CPP-TA-F037	CPP_F002_TestCase
CPP-TA-F038	CPP_F001_TestCase
CPP-TA-F039	CPP_F001_TestCase
CPP-TA-F040	Not tested – not all platforms allow application access to header content
CPP-TA-F041	Not tested – not all platforms allow application access to header content
CPP-TA-F043	Not tested – This is an assertion about the web services binding behavior
CPP-TA-F044	CPP_F001_TestCase
CPP-TA-F045	CPP_F001_TestCase
CPP-TA-F046	CPP_F001_TestCase
CPP-TA-F047	CPP_C001_TestCase

CPP-TA-F048	Not tested – This is an assertion about the web services binding behavior

160 # 6 Catalog of Test Artifacts

161 ## 6.1 C++ Interfaces

Name	Description
Service1.h	1 operation with 1 input, 1 output parameter
Service1Superset.h	Remotable service interface which is a superset of Service1 interface 2 operations with 1 input,1 output parameter
Service3.h	Remotable service with Operation1 is overloaded
Service4.h	Remotable service interface 1 operation with 1 input, 1 output parameter - Input parameter is not const
Service5.h	Remotable service interface with a Callback interface (Service5Callback) 1 operation with 1 input, 1 output parameter
Service5Callback.h	Remotable callback interface for Service5 1 operation with 1 input, 1 output parameter
Service6.h	Remotable service interface with a local Callback interface (Service6Callback) 1 operation with 1 input, 1 output parameter
Service6Callback.h	Local callback interface for Service6 1 operation with 1 input, 1 output parameter
Service7.h	Local service interface with a remotable Callback interface (Service7Callback) 1 operation with 1 input, 1 output parameter
Service7Callback.h	Remotable callback interface for Service7 1 operation with 1 input, 1 output parameter
Service11.h	Remotable interface with a function in the file that is not part of the interface - it is also a superset of Service1
Service12.h	Remotable service interface with a Callback interface (Service12Callback) 1 operation with 1 input, 1 output parameter
Service12Callback.h	Remotable callback interface for Service12 with a function in the file that is not part of the callback interface, 1 operation with 1 input, 1 output parameter
Service13.h	Remotable interface containing no public member

	functions
Service14.h	Remotable interface containing no public non virtual member functions
Service15.h Service15Exp.h	Local service interface with a macro and a typedef used in the declaration. Same interface with macro and typedef expanded
Service16.h	Remotable service interface with a UDT parameter
Service18.h	Remotable service interface with a unnamed parameter
Service19.h	Remotable service interface with a member function with multiple parameter passing styles
Service20.h (system under test generated from Service20.wsdl – should fail)	
Service21.h	Remotable service interface with member functions with multiple parameter and return types
Service22.h Service22Super.h	Remotable service interface that inherits from another service interface.
Service23.h	Remotable service interface with a member function with parameters names a_x and xml
Service24.h Service24OW.h	Remotable service interface with a member function with only an in parameter and a void return type Remotable service interface with a member function with only an in parameter and a void return type designated oneWay
Service25.h	Remotable service interface with function annotated with @WebParam
Service26.h	Remotable service interface with function annotated with @WebResult
Service27.h	Remotable service interface with function annotated with @WebParam but incorrect simple type value
Service28.h	Remotable service interface with overloaded function annotated with @WebFunction to disambiguate
Service29.h	Remotable service interface with function annotated with @WebResult but incorrect simple type value
Service30.h	Remotable service interface with function annotated with @WebParam parameter is unnamed
Service31.h (system under test generated from Service31.wsdl)	Remotable service interface with a member function that can throw a fault.

OperationFault.h (system under test generated from Service31.wsdl)	Exception Class for the fault
Service32.h (system under test generated from Service32.wsdl) Service32F.h (system under test generated from Service32F.wsdl)	Remotable service interface with member functions with multiple parameter and return types Same but multiple uses of OperationFault
Service35.h (system under test generated from Service35.wsdl)	Remotable service interface based on WSDL with a namespace extension
Service36.h (system under test generated from Service36.wsdl)	Remotable service interface
Service37.h (system under test generated from Service37.wsdl)	Remotable service interface based on WSDL with a function extension
Service38.h (system under test generated from Service38.wsdl)	Remotable service interface based on WSDL with a parameter extension
Service39.h (system under test generated from Service39.wsdl)	Remotable service interface based on WSDL with a class extension
Service40.h (system under test generated from Service40.wsdl)	Remotable service interface based on WSDL with an enableWrapperStyle extension
Service42.h (system under test generated from Service42.wsdl)	Remotable service interface based on WSDL with a parameter extension with a specified type
Service45.h	Service interface annotated with @WebService
Service46.h	Remotable service with operation1 that sleeps before returning
Service47.h	Remotable service interface with a Callback interface (Service47Callback)
Service47Callback.h	Remotable callback interface for Service47 with callback1 that sleeps before returning

6.2 C++ Implementations

Name	Description
	Services and References use interface Service1 unless described otherwise
Service1Impl.h Service1Impl.cpp Service1Impl.componentType	1 service 0 references
Service1Impl2.h Service1Impl2.cpp Service1Impl2.componentType	1 service 1 reference (1..1)
Service1Impl3.h	1 service

Service1Impl3.cpp Service1Impl3.componentType	1 reference (1..n)
Service1Impl7.h Service1Impl7.cpp Service1Impl7.componentType	1 service 1 reference (1..1) with interface Service 4
Service1Impl11.h Service1Impl11.cpp Service1Impl11.componentType	1 service 1 reference (1..1) - Operation invoked twice
Service1Impl12.h Service1Impl12.cpp Service1Impl12.componentType	1 service 0 references Composite scope with data member
Service1Impl13.h Service1Impl13.cpp Service1Impl13.componentType	1 service 0 references Stateless scope with data member
Service1Impl14.h Service1Impl14.cpp Service1Impl14.componentType	1 service 1 reference (1..1) with interface Service11 Incorrect operation invoked
Service1Impl15.h Service1Impl15.cpp Service1Impl15.componentType	1 service 1 reference (1..1) with interface Service11 No default constructor
Service1Impl16.h Service1Impl16.cpp Service1Impl16.componentType	1 service 1 reference (1..1) with interface Service15Exp
Service1Impl17.h Service1Impl17.cpp Service1Impl17.componentType	1 service 1 reference (1..1) with interface Service16
Service1Impl18.h Service1Impl18.cpp Service1Impl18.componentType	1 service 1 reference (1..1)
Service1Impl19.h Service1Impl19.cpp Service1Impl19.componentType	1 service 1 reference (1..n)
Service1Impl20.h Service1Impl20.cpp Service1Impl20.componentType	1 service 0 references
Service1Impl21.h Service1Impl21.cpp Service1Impl21.componentType	1 service 1 reference (1..1) with Service14

Copyright © OASIS® 2010. All Rights Reserved.

Service1Impl22.h Service1Impl22.cpp Service1Impl22.componentType	1 service 0 references
Service1Impl23.h Service1Impl23.cpp Service1Impl23.componentType Service1Impl23OW.h Service1Impl23OW.cpp Service1Impl23OW.componentType	1 service 1 reference (1..1) with Service18 One function marked oneWay
Service1Impl24.h Service1Impl24.cpp Service1Impl24.componentType	1 service 1 reference (1..1) with Service19
Service1Impl25.h Service1Impl25.cpp Service1Impl25.componentType	1 service 1 reference (1..1) with Service20
Service1Impl26.h Service1Impl26.cpp Service1Impl26.componentType	1 service 1 reference (1..1) with Service21
Service1Impl27.h Service1Impl27.cpp Service1Impl27.componentType	1 service 1 reference (1..1) with Service22
Service1Impl28.h Service1Impl28.cpp Service1Impl28.componentType	1 service 1 reference (1..1) with Service23
Service1Impl29.h Service1Impl29.cpp Service1Impl29.componentType Service1Impl29OW.h Service1Impl29OW.cpp Service1Impl29OW.componentType	1 service 1 reference (1..1) with interface Service24(OW)
Service1Impl30.h Service1Impl30.cpp Service1Impl30.componentType	1 service 1 reference (1..1)
Service1Impl33.h Service1Impl33.cpp Service1Impl33.componentType	1 service 1 reference (1..1) with interface Service25
Service1Impl34.h Service1Impl34.cpp Service1Impl34.componentType	1 service 1 reference (1..1) with interface Service26

Service1Impl35.h Service1Impl35.cpp Service1Impl35.componentType	1 service 1 reference (1..1) with interface Service27
Service1Impl36.h Service1Impl36.cpp Service1Impl36.componentType	1 service 1 reference (1..1) with interface Service28
Service1Impl37.h Service1Impl37.cpp Service1Impl37.componentType	1 service 1 reference (1..1) with interface Service29
Service1Impl38.h Service1Impl38.cpp Service1Impl38.componentType	1 service 1 reference (1..1) with interface Service30
Service1Impl39.h Service1Impl39.cpp Service1Impl39.componentType	1 service 1 reference (1..1) with interface Service31
Service1Impl40.h Service1Impl40.cpp Service1Impl40.componentType Service1Impl40F.h Service1Impl40F.cpp Service1Impl40F.componentType	1 service 1 reference (1..1) with interface Service32 1 service 1 reference (1..1) with interface Service32F
Service1Impl42.h Service1Impl42.cpp Service1Impl42.componentType	1 service 1 reference (1..1) with interface Service35
Service1Impl43.h Service1Impl43.cpp Service1Impl43.componentType	1 service 1 reference (1..1) with interface Service36
Service1Impl44.h Service1Impl44.cpp Service1Impl44.componentType	1 service 1 reference (1..1) with interface Service37
Service1Impl45.h Service1Impl45.cpp Service1Impl45.componentType	1 service 1 reference (1..1) with interface Service38
Service1Impl46.h Service1Impl46.cpp Service1Impl46.componentType	1 service 1 reference (1..1) with interface Service39
Service1Impl47.h Service1Impl47.cpp Service1Impl47.componentType	1 service 1 reference (1..1) with interface Service40

Service1Impl49.h Service1Impl49.cpp Service1Impl49.componentType	1 service 1 reference (1..1) with interface Service42
Service1Impl52.h Service1Impl52.cpp Service1Impl52.componentType	1 service 1 reference (1..1) with interface Service45
Service1Impl53.h Service1Impl53.cpp Service1Impl53.componentType	1 service 1 reference (1..1) with interface Service46 Asynch invocation with callback
Service1Impl54.h Service1Impl54.cpp Service1Impl54.componentType	1 service 1 reference (1..1) with interface Service46 Asynch invocation with polling
Service1Impl55.h Service1Impl55.cpp Service1Impl55.componentType	1 service 1 reference (1..1) with interface Service46 Asynch invocation that is cancelled
Service1Impl56.h Service1Impl56.cpp Service1Impl56.componentType	1 service 1 reference (1..1) Two threads created each invokes operation on reference
Service1Impl57.h Service1Impl57.cpp Service1Impl57.componentType	1 service 0 references Uses a global variable
Service1Impl58.h Service1Impl58.cpp Service1Impl58.componentType	1 service 0 references Uses a data member
Service1Callback5Impl.h Service1Callback5Impl.cpp Service1Callback5Impl.componentType	1 service 1 reference with Service5/Service5Callback interfaces
Service1Callback6Impl.h Service1Callback6Impl.cpp Service1Callback6Impl.componentType	1 service 1 reference with Service6/Service6Callback interfaces
Service1Callback7Impl.h Service1Callback7Impl.cpp Service1Callback7Impl.componentType	1 service 1 reference with Service7/Service7Callback interfaces
Service1Callback12Impl.h Service1Callback12Impl.cpp Service1Callback12Impl.componentType	1 service 1 reference with Service12/Service12Callback interfaces
Service1Callback47Impl.h Service1Callback47Impl.cpp	1 service 1 reference with Service47/Service47Callback

Service1Callback47Impl.componentType	interfaces
Service1_16Impl.h Service1_16Impl.cpp Service1_16Impl.componentType	2 services, one with interface Service1 and one with interface Service16 0 references - Operation of service1 invokes operation of service 16
Service1SupersetImpl2.h Service1SupersetImpl2.cpp Service1SupersetImpl2.componentType	1 service with interface Service1Superset
Service3Impl.h Service3Impl.cpp Service3Impl.componentType	1 service with interface Service3
Service4Impl.h Service4Impl.cpp Service4Impl.componentType	1 service with interface Service4
Service5Impl.h Service5Impl.cpp Service5Impl.componentType	1 service with interface Service5
Service6Impl.h Service6Impl.cpp Service6Impl.componentType	1 service with interface Service6
Service7Impl.h Service7Impl.cpp Service7Impl.componentType	1 service with interface Service7
Service12Impl.h Service12Impl.cpp Service12Impl.componentType	1 service with interface Service12
Service13Impl.h Service13Impl.cpp Service13Impl.componentType	1 service with interface Service13
Service14Impl.h Service14Impl.cpp Service14Impl.componentType	1 service with interface Service14
Service15Impl.h Service15Impl.cpp Service15Impl.componentType	1 service with interface Service15
Service16Impl.h Service16Impl.cpp Service16Impl.componentType	1 service with interface Service16

Service18Impl.h Service18Impl.cpp Service18Impl.componentType	1 service with interface Service18
Service19Impl.h Service19Impl.cpp Service19Impl.componentType	1 service with interface Service19
Service20Impl.h Service20Impl.cpp Service20Impl.componentType	1 service with interface Service20
Service21Impl.h Service21Impl.cpp Service21Impl.componentType	1 service with interface Service21
Service22Impl.h Service22Impl.cpp Service22Impl.componentType	1 service with interface Service22
Service23Impl.h Service23Impl.cpp Service23Impl.componentType	1 service with interface Service23
Service24Impl.h Service24Impl.cpp Service24Impl.componentType Service24IOWmpl.h Service24OWImpl.cpp Service24OWImpl.componentType	1 service with interface Service24 One function marked oneWay
Service25Impl.h Service25Impl.cpp Service25Impl.componentType	1 service with interface Service25
Service26Impl.h Service26Impl.cpp Service26Impl.componentType	1 service with interface Service26
Service27Impl.h Service27Impl.cpp Service27Impl.componentType	1 service with interface Service27
Service28Impl.h Service28Impl.cpp Service28Impl.componentType	1 service with interface Service28
Service29Impl.h Service29Impl.cpp Service29Impl.componentType	1 service with interface Service29

Service30Impl.h Service30Impl.cpp Service30Impl.componentType	1 service with interface Service30
Service31Impl.h Service31Impl.cpp Service31Impl.componentType OperationFault.cpp	1 service with interface Service31
Service32Impl.h Service32Impl.cpp Service32Impl.componentType Service32ImplF.h Service32ImplF.cpp Service32ImplF.componentType	1 service with interface Service32 1 service with interface Service32F
Service35Impl.h Service35Impl.cpp Service35Impl.componentType	1 service with interface Service35
Service36Impl.h Service36Impl.cpp Service36Impl.componentType	1 service with interface Service36
Service37Impl.h Service37Impl.cpp Service37Impl.componentType	1 service with interface Service37
Service38Impl.h Service38Impl.cpp Service38Impl.componentType	1 service with interface Service38
Service39Impl.h Service39Impl.cpp Service39Impl.componentType	1 service with interface Service39
Service40Impl.h Service40Impl.cpp Service40Impl.componentType	1 service with interface Service40
Service42Impl.h Service42Impl.cpp Service42Impl.componentType	1 service with interface Service42
Service45Impl.h Service45Impl.cpp Service45Impl.componentType	1 service with interface Service45
Service46Impl.h Service46Impl.cpp	1 service with interface Service46

Service46Impl.componentType	
Service47Impl.h Service47Impl.cpp Service47Impl.componentType	1 service with interface Service47

7 Conformance

There are no conformance statements relating to the TestCases.

Copyright © OASIS® 2010. All Rights Reserved.

A. Acknowledgements

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

Participants:

Participant Name	Affiliation
Bryan Aupperle	IBM
Andrew Borley	IBM
Jean-Sebastien Delfino	IBM
Mike Edwards	IBM
David Haney	Individual
Mark Little	Red Hat
Jeff Mischkinsky	Oracle Corporation
Peter Robbins	IBM

B. Revision History

[optional; should not be included in OASIS Standards]

Revision	Date	Editor	Changes Made
			•