



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

Committee Draft 01 / Public Review Draft 01

29 April 2010

**Specification URIs:**

**This Version:**

<http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions-cd01.html>  
<http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions-cd01.doc>  
<http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions-cd01.pdf>  
(Authoritative)

**Previous Version:**

N/A

**Latest Version:**

<http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions.html>  
<http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions.doc>  
<http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-test-assertions.pdf> (Authoritative)

**Technical Committee:**

OASIS Service Component Architecture / C and C++ (SCA-C-C++) TC

**Chair:**

Bryan Aupperle, IBM

**Editors:**

Bryan Aupperle, IBM  
David Haney  
Pete Robbins, IBM

**Related work:**

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 Assertions for the SCA C++ Client and Implementation Model specification.

The Test Assertions represent the testable items relating to the normative statements made in the SCA C++ Client and Implementation Model specification. The Test Assertions provide a bridge between the normative statements in the specification and the conformance TestCases which are designed to check that an SCA runtime conforms to the requirements of the 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/>.

---

## Notices

Copyright © OASIS® 2009. 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](http://www.oasis-open.org), 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	Example Test Assertion.....	5
1.2	Terminology.....	5
1.3	Normative References.....	5
2	Test Assertions.....	7
2.1	Section 2.....	7
2.2	Section 3.....	11
2.3	Section 4.....	12
2.4	Section 7.....	13
2.5	Section 8.....	13
2.6	Section 9.....	14
2.7	Section 10.....	17
2.8	Appendix A.....	18
2.9	Appendix C.....	19
2.10	Appendix D.....	21
2.11	Appendix F.....	23
3	Conformance.....	35
A.	Cross Mapping of Normative Statements to Assertions.....	36
B.	Acknowledgements.....	41

---

# 1 Introduction

This document defines the Test Assertions for the SCA Assembly Specification Version 1.1 [SCA C++]. The test assertions in this document follow the format defined in the OASIS Test Assertion Guidelines specification [TAG].

## 1.1 Example Test Assertion

Test assertions are presented in a tabular format with rows corresponding to the entry types defined in [TAG].

Assertion ID	CPP-TA-xxxx
Source	[CPPx00yy]
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"

**Assertion ID:** Is a unique ID for the test assertion. Its format starts with a 3 letter string that identifies the specification to which it relates - "CPP" is for the SCA C++ Client and Implementation Model specification. This is followed by "-TA-" to indicate that this identifier is for a test assertion. This is then followed by a unique 4 digit number.

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

**Target:** Identifies the target which is addressed by this assertion. This is typically some SCA document element, or other SCA artifact but possibly could identify an SCA implementation and its behavior.

**Prerequisites:** Defines any prerequisites for this test assertion. The prerequisites may be defined in terms of one or more other test assertions that must be true.

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

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

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

## 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.

32       **[SCA C++]**       OASIS Committee Draft 05, *Service Component Architecture Client and*  
33                            *Implementation Model for C++ Specification Version 1.1*, March 2010.  
34                            <http://docs.oasis-open.org/opencsa/sca-c-cpp/sca-cppcni-1.1-spec-cd05.pdf>  
35       **[TAG]**            OASIS Committee Draft 04, *Test Assertion Guidelines*, February 2010.  
36                            [http://docs.oasis-open.org/tag/guidelines/v1.0/cd04/testassertionsguidelines-cd-](http://docs.oasis-open.org/tag/guidelines/v1.0/cd04/testassertionsguidelines-cd-04.pdf)  
37                            04.pdf

## 38 2 Test Assertions

### 39 2.1 Section 2

Assertion ID	CPP-TA-2001
Source	[CPP20001]
Target	<implementation.cpp/> of a <component/>
Prerequisites	<component/> declares at least one <service/>
Predicate	Each operation of each <service/> declared by the <component/> is implemented by the <implementation.cpp>.
Prescription Level	Mandatory
Tags	"implementation.cpp"

40

Assertion ID	CPP-TA-2002
Source	[CPP20003]
Target	@scope attribute of <implementation.cpp/> of a <component/>
Prerequisites	@scope attribute of the <implementation.cpp> is either not present or present and has a value of "stateless" or "composite".
Predicate	The <component/> is deployed and runnable.
Prescription Level	Mandatory
Tags	"implementation.cpp" "scope"

41

Assertion ID	CPP-TA-2003
Source	[CPP20012]
Target	@scope attribute of a <implementation.cpp/> of a <component/>
Prerequisites	@scope attribute of the <implementation.cpp> is either not present or present and has a value of "stateless".
Predicate	An implementation instance only executes on one execution thread.
Prescription Level	Mandatory
Tags	"implementation.cpp" "scope" "stateless scope"

42

Assertion ID	CPP-TA-2004
Source	[CPP20012]
Target	@scope attribute of a <implementation.cpp/> of a <component/>
Prerequisites	@scope attribute of the <implementation.cpp> is either not present or present and has a value of "stateless"
Predicate	Only one operation of the component is invoked in an implementation instance lifecycle.
Prescription	Mandatory

Level	
Tags	"implementation.cpp" "scope" "stateless scope"

43

Assertion ID	CPP-TA-2005
Source	[CPP20013]
Target	@scope attribute of a <implementation.cpp/> of a <component/>
Prerequisites	@scope attribute of the <implementation.cpp> has a value of "composite".
Predicate	Multiple execution threads can simultaneously use an implementation instance.
Prescription Level	Optional
Tags	"implementation.cpp" "scope" "composite scope"

44

Assertion ID	CPP-TA-2006
Source	[CPP20005]
Target	@class attribute of a <interface.cpp/> element
Prerequisites	
Predicate	@class attribute of the <interface.cpp> specifies a specific class in the file referenced by the @header attribute of the <interface.cpp>.
Prescription Level	Mandatory
Tags	"interface.cpp"

45

Assertion ID	CPP-TA-2007
Source	[CPP20006]
Target	@callbackClass attribute of a <interface.cpp/> element
Prerequisites	
Predicate	@callbackClass attribute of the <interface.cpp> specifies a specific class in the file referenced by the @callbackHeader attribute of the <interface.cpp>.
Prescription Level	Mandatory
Tags	"interface.cpp" "callbacks"

46

Assertion ID	CPP-TA-2008
Source	[CPP20007]
Target	<function/> element in a <interface.cpp/> element
Prerequisites	The <interface.cpp> element has more than one <function/> elements.
Predicate	The @name attribute [of the <function/> element] is not equal to the @name attribute of any other <function/> element in the <interface.cpp/>.
Prescription Level	Mandatory
Tags	"interface.cpp"

47

Assertion ID	CPP-TA-2009
--------------	-------------



Source	[CPP20008]
Target	<callbackFunction/> element in a <interface.cpp/> element
Prerequisites	The <interface.cpp> element has more than one <callbackFunction/> elements.
Predicate	The @name attribute [of the <callbackFunction/> element] is not equal to the @name attribute of any other <callbackFunction/> element in the <interface.cpp/>.
Prescription Level	Mandatory
Tags	"interface.cpp" "callbacks"

48

Assertion ID	CPP-TA-2010
Source	[CPP20009]
Target	@class attribute of a <implementation.cpp/> element
Prerequisites	
Predicate	A componentType file with a root name matching the value of the @class attribute, excluding any namespace definitions, of the <implementation.cpp/> is found.
Prescription Level	Mandatory
Tags	"implementation.cpp" "componentType"

49

Assertion ID	CPP-TA-2011
Source	[CPP20010]
Target	<function/> element in a <implementation.cpp/> element
Prerequisites	The <implementation.cpp> element has more than one <function/> elements.
Predicate	The @name attribute [of the <function/> element] is not equal to the @name attribute of any other <function/> element in the <implementation.cpp/>.
Prescription Level	Mandatory
Tags	"implementation.cpp"

50

Assertion ID	CPP-TA-2012
Source	[CPP20011]
Target	<implementation.cpp/> of a <component/>
Prerequisites	
Predicate	The SCA runtime can create an implementation instance using a default constructor.
Prescription Level	Mandatory
Tags	"implementation.cpp"

51

Assertion ID	CPP-TA-2013
Source	[CPP20014]
Target	@allowsPassByReference attribute of an <implementation.cpp/> element or a <<function/> element of an <implementation.cpp/> element.
Prerequisites	The interface of a service is defined with an <interface.cpp/> element, the client

	implementation and the service operation implementation of a wire are marked “allowsPassByReference” and a member function of the interface has parameters that can be passed by-reference.
Predicate	By-reference marshalling is used for the operation invocation.
Prescription Level	Permitted
Tags	“implementation.cpp” “allowsPassByReference”

52

Assertion ID	CPP-TA-2014
Source	[CPP20015]
Target	@allowsPassByReference attribute of an <implementation.cpp/> element or a <<function/> element of an <implementation.cpp/> element.
Prerequisites	Either the client implementation or the service operation implementation of a wire is not marked “allowsPassByReference” .
Predicate	By-value marshalling is used for the operation invocation.
Prescription Level	Mandatory
Tags	“implementation.cpp” “allowsPassByReference”

53

Assertion ID	CPP-TA-2015
Source	[CPP20016]
Target	<function/> element of an <interface.cpp>
Prerequisites	@class attribute of the <interface.cpp> references a class containing member functions that are excluded from the interface via <function/> child elements.
Predicate	A member function excluded from the interface cannot be invoked.
Prescription Level	Mandatory
Tags	“interface.cpp”

54

Assertion ID	CPP-TA-2016
Source	[CPP20017]
Target	<callbackFunction/> element of an <interface.cpp>
Prerequisites	@callbackClass attribute of the <interface.cpp> references a class containing member functions that are excluded from the callback interface via <callbackFunction/> child elements.
Predicate	A member function excluded from the callback interface cannot be invoked.
Prescription Level	Mandatory
Tags	“interface.cpp” “callbacks”

55

Assertion ID	CPP-TA-2017
Source	[CPP20018]
Target	<implementation.cpp/>
Prerequisites	An implementation uses static or global variables.
Predicate	Multiple execution threads can simultaneously modify a global or static variable.
Prescription Level	Mandatory
Tags	"implementation.cpp"

56 **2.2 Section 3**

Assertion ID	CPP-TA-3001
Source	[CPP30001]
Target	Service proxy generation
Prerequisites	A remotable interface is defined with a C++ file.
Predicate	The interface definition is mapped to WSDL and the service proxy is generated from the WSDL.
Prescription Level	Preferred
Tags	"Service proxy"

57

Assertion ID	CPP-TA-3002
Source	[CPP30002]
Target	<reference/> element of a <component/>
Prerequisites	A component has a reference.
Predicate	A service proxy, derived from ServiceProxy, is generated for the reference and the proxy has a member function for each operation of the interface.
Prescription Level	Mandatory
Tags	"Service proxy"

58

Assertion ID	CPP-TA-3003
Source	[CPP30003]
Target	<reference/> element of a <component/>
Prerequisites	A component has a reference with an <interface/> that includes at least one operation marked @requires="asynct invocation".
Predicate	The generated service proxy for the reference contains an asynchronous invocation member function for each operation of the <interface/> that is marked @requires="asynct invocation".
Prescription Level	Mandatory
Tags	"Service proxy" "asynchronous invocation"

59

Assertion ID	CPP-TA-3004
--------------	-------------

Source	[CPP30004]
Target	<reference/> element of a <component/>
Prerequisites	A component has a reference with an <interface/> that includes at least one operation marked @requires="asynctinvocation".
Predicate	The generated service proxy for the reference contains a response class for each response message for an operation of the <interface/> that is marked @requires="asynctinvocation".
Prescription Level	Mandatory
Tags	"Service proxy" "asynchronous invocation"

## 60 2.3 Section 4

Assertion ID	CPP-TA-4001
Source	[CPP40001]
Target	<binding/> of <interface/>
Prerequisites	Operation of the <interface/> is designated as using a oneway message exchange pattern.
Predicate	The request is queued for later delivery.
Prescription Level	Permitted
Tags	"oneway operations"

61

Assertion ID	CPP-TA-4002
Source	[CPP40002]
Target	<service/> element of a <component/>
Prerequisites	The <interface/> of a <service/> is bidirectional.
Predicate	A service proxy, derived from ServiceProxy, is generated for the service and the proxy has a member function for each operation of the callback interface.
Prescription Level	Mandatory
Tags	"Service proxy"

62

Assertion ID	CPP-TA-4003
Source	[CPP40003]
Target	<service/> element of a <component/>
Prerequisites	The <interface/> of a <service/> is bidirectional and includes at least one operation marked @requires="asynctinvocation" in the callback interface.
Predicate	The generated service proxy for the reference contains an asynchronous invocation member function for each operation of the <interface/> that is marked @requires="asynctinvocation" and a response class for each response message for an operation of the <interface/> that is marked @requires="asynctinvocation".
Prescription Level	Mandatory
Tags	"Service proxy"

63 **2.4 Section 7**

Assertion ID	CPP-TA-7001
Source	[CPP70001]
Target	<export.cpp/> element
Prerequisites	The domain has more than one <export.cpp>.
Predicate	The @name attribute of [the <export.cpp/> element] is not equal to the @name attribute of any other <export.cpp/> element in the domain.
Prescription Level	Mandatory
Tags	"export.cpp"

64

Assertion ID	CPP-TA-7002
Source	[CPP70002]
Target	<import.cpp/> element
Prerequisites	A contribution has more than one <import.cpp>.
Predicate	The @name attribute of [the <import.cpp/> element] is not equal to the @name attribute of any other <import.cpp/> element in the contribution.
Prescription Level	Mandatory
Tags	"import.cpp"

65 **2.5 Section 8**

Assertion ID	CPP-TA-8001
Source	[CPP80001]
Target	<interface.cpp/> element
Prerequisites	A class is identified by the attributes of an <interface.cpp> as defining an interface and declarations of the public member functions contain macros and typedefs.
Predicate	The interface defined by the class is compatible with the same interface defined without macros and typedefs.
Prescription Level	Mandatory
Tags	"interface.cpp" "header files"

66

Assertion ID	CPP-TA-8002
Source	[CPP80002]
Target	<interface.cpp/> element
Prerequisites	The value of the @remotable attribute of the <interface.cpp/> is true.
Predicate	The return type and the type of the parameters of every member function of the interface is either a C++ type identified in the Simple Content Binding of the specification <b>[SCA C++]</b> or a DataObjectPtr.
Prescription Level	Mandatory
Tags	"interface.cpp" "header files"

67

Assertion ID	CPP-TA-8003
Source	[CPP80003]
Target	<interface.cpp>
Prerequisites	A header file is named by either the @header or the @callbackHeader attribute of an <interface.cpp>
Predicate	The header file declares at least one class with at least one public member function.
Prescription Level	Mandatory
Tags	"interface.cpp" "header files"

68

Assertion ID	CPP-TA-8004
Source	[CPP80003]
Target	<interface.cpp>
Prerequisites	A header file is named by either the @header or the @callbackHeader attribute of an <interface.cpp> and either contains only one class that declares public member functions or the class named by the @class or the @callbackClass attribute of the <interface.cpp> declares public functions.
Predicate	Every public member function of the specified class is a pure virtual member function.
Prescription Level	Mandatory
Tags	"interface.cpp" "header files"

69

## 2.6 Section 9

Assertion ID	CPP-TA-10001
Source	[CPP100001]
Target	Namespace mapping
Prerequisites	WSDL definition does not contain an <cpp:namespace/> element
Predicate	The generated C++ class(es) are in an implementation dependent default namespace.
Prescription Level	Mandatory
Tags	"WSDL mapping"

70

Assertion ID	CPP-TA-10002
Source	[CPP100001]
Target	Namespace mapping
Prerequisites	
Predicate	The implementation provides a way to control the default namespace for generated classes.
Prescription Level	Preferred
Tags	"WSDL mapping"

71

Assertion ID	CPP-TA-10003
Source	[CPP100002]
Target	Fault mapping
Prerequisites	WSDL definition contains multiple <operation/> elements that refer to a single fault <message /> element.
Predicate	A single exception class is generated and the @WebThrows annotation for each of the generated member functions corresponding to the <operation/> elements lists the exception class.
Prescription Level	Mandatory
Tags	"WSDL mapping"

72

Assertion ID	CPP-TA-10004
Source	[CPP100003]
Target	Parameter mapping
Prerequisites	A WSDL <operation/> element referring to unwrapped input and output <message/> elements containing some subset of in message parts, in/out message parts, out message parts not named return, and an out message part named return.
Predicate	Each in message part is mapped to a parameter passed by const-reference. Each in/out message part is mapped to a parameter passed by reference. Each out message part not named return is mapped to a parameter passed by reference. A out message part named return is mapped to the member function return type passed by-value.
Prescription Level	Mandatory
Tags	"WSDL mapping"

73

Assertion ID	CPP-TA-10005
Source	[CPP100004]
Target	Parameter mapping
Prerequisites	A WSDL <operation/> element referring to wrapped input and output <message/> elements containing some subset of in wrapper children, in/out wrapper children, out wrapper children not named return, and an out wrapper child named return.
Predicate	Each in wrapper child is mapped to a parameter passed by const-reference. Each in/out wrapper child is mapped to a parameter passed by reference. Each out wrapper child not named return is mapped to a parameter passed by reference. A out wrapper child named return is mapped to the member function return type passed by-value.
Prescription Level	Mandatory
Tags	"WSDL mapping"

74

Assertion ID	CPP-TA-10006
--------------	--------------

Source	[CPP100005]
Target	Namespace mapping
Prerequisites	
Predicate	The implementation provides a way to control the default target namespace for generated WSDL definitions.
Prescription Level	Preferred
Tags	"WSDL mapping"

75

Assertion ID	CPP-TA-10007
Source	[CPP100006]
Target	Parameter mapping
Prerequisites	A member function with parameters, not annotated with @WebParam, that are some subset of by-reference, by-pointer, by-value, and by-const reference.
Predicate	The member function's return type is mapped to an out message part or wrapper child. Each by-reference and each by-pointer parameter is mapped to an in/out message part or wrapper child. All other parameters are mapped to in message parts or wrapper children.
Prescription Level	Mandatory
Tags	"WSDL mapping"

76

Assertion ID	CPP-TA-10009
Source	[CPP100008]
Target	Type mapping
Prerequisites	
Predicate	A XSD type is mapped according to Table 1 of the specification <b>[SCA C++]</b> .
Prescription Level	Mandatory
Tags	"WSDL mapping"

77

Assertion ID	CPP-TA-10010
Source	[CPP100008]
Target	Type mapping
Prerequisites	
Predicate	A C++ type is mapped according to Tables 1 and 2 of the specification <b>[SCA C++]</b> .
Prescription Level	Mandatory
Tags	"WSDL mapping"

78

Assertion ID	CPP-TA-10011
Source	[CPP100009]



Target	portType mapping
Prerequisites	
Predicate	A WSDL portType is mapped to a remotable SCA interface
Prescription Level	Mandatory
Tags	“WSDL mapping”

79

Assertion ID	CPP-TA-10012
Source	[CPP100010]
Target	C++ interface mapping
Prerequisites	A C++ class, is not annotated with @WebService.
Predicate	The class is mapped to WSDL as if it had a @WebService annotation with no parameters.
Prescription Level	Mandatory
Tags	“WSDL mapping”

80 **2.7 Section 10**

Assertion ID	CPP-TA-11001
Source	[CPP110001]
Target	Composite file that violates the sca-interface-cpp.xsd schema and / or the sca-implementation-cpp.xsd schema.
Prerequisites	
Predicate	SCA runtime rejects the composite file and raises an error
Prescription Level	Mandatory
Tags	“composite” “schema” “error”

81

Assertion ID	CPP-TA-11002
Source	[CPP110002]
Target	ComponentType file that violates the sca-interface-cpp.xsd schema.
Prerequisites	
Predicate	SCA runtime rejects the componentType file and raises an error
Prescription Level	Mandatory
Tags	“componentType” “schema” “error”

82

Assertion ID	CPP-TA-11004
Source	[CPP110003]
Target	Contribution file that violates the sca-contribution-cpp.xsd schema.
Prerequisites	
Predicate	SCA runtime rejects the contribution file and raises an error
Prescription	Mandatory

Level	
Tags	"contribution" "schema" "error"

83

Assertion ID	CPP-TA-11005
Source	[CPP110004]
Target	WSDL file that violates the sca-wsdlex-cpp.xsd schema.
Prerequisites	
Predicate	SCA runtime rejects the WSDL file and raises an error
Prescription Level	Mandatory
Tags	"WSDL" "schema" "error"

## 84 2.8 Appendix A

85 The test assertions in this section apply to the optional C++ SCA annotation support. These assertions  
86 are only applicable if an implementation supports these annotations.

Assertion ID	CPP-TA-A001
Source	[CPPA0001]
Target	SCA annotations
Prerequisites	A source file containing SCA annotations is processed.
Predicate	SCDL file(s) containing elements, attributes and values corresponding to the annotations is(are) generated.
Prescription Level	Mandatory
Tags	"SCA annotations"

87

Assertion ID	CPP-TA-A002
Source	[CPPA0001]
Target	SCA annotations
Prerequisites	A header file containing SCA annotations is referenced by an <interface.cpp>
Predicate	The SCA annotations are ignored during component execution.
Prescription Level	Mandatory
Tags	"SCA annotations"

88

Assertion ID	CPP-TA-A003
Source	[CPPA0002]
Target	SCA annotations
Prerequisites	A source file containing SCA annotations with sequential comments containing annotations related to the same program element is processed.
Predicate	The annotations are processed as if they were in one comment block.
Prescription Level	Preferred
Tags	"SCA annotations"

89

Assertion ID	CPP-TA-A004
Source	[CPPA0003]
Target	@WebService annotation
Prerequisites	A class is annotated with @WebService, but not @Remotable.
Predicate	The generated SCDL is the same as if the class was annotated with @Remotable.
Prescription Level	Mandatory
Tags	"SCA annotations" "WSDL annotations"

90

Assertion ID	CPP-TA-A005
Source	[CPPA0004]
Target	@WebFunction annotation
Prerequisites	A member function declaration with a @WebFunction annotation but not a corresponding @Function annotation.
Predicate	The generated SCDL is the same as if the member function was annotated with a @Function annotation with a name value equal to the operationName value of the @WebFunction annotation and an exclude value equal to the exclude value of the @WebFunction annotation.
Prescription Level	Mandatory
Tags	"SCA annotations" "WSDL annotations"

## 91 2.9 Appendix C

92 The test assertions in this section apply to the optional C++ WSDL annotation support. These assertions  
93 are only applicable if an implementation supports these annotations.

Assertion ID	CPP-TA-C001
Source	[CPPC0001]
Target	WSDL annotations
Prerequisites	A source file containing WSDL annotations, explicit or implied by corresponding SCA annotations, is processed.
Predicate	A valid WSDL 1.1 file containing elements, attributes and values corresponding to the annotations is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

94

Assertion ID	CPP-TA-C002
Source	[CPPC0002]
Target	@Remotable annotation
Prerequisites	A class is annotated with @Remotable, but not @WebService.
Predicate	The generated WSDL is the same as if the class was annotated with @WebService

	with no parameters.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations" "SCA annotations"

95

Assertion ID	CPP-TA-C004
Source	[CPPC0004]
Target	@WebParam
Prerequisites	A member function is annotated with @WebParam.
Predicate	The paramName value of the @WebParam annotation matches the name of a parameter of the member function.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

96

Assertion ID	CPP-TA-C005
Source	[CPPC0005]
Target	@WebParam
Prerequisites	A member function is annotated with @WebParam which has a type value.
Predicate	The type value of the @WebParam annotation is a valid simple type from the namespace: <a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> .
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

97

Assertion ID	CPP-TA-C006
Source	[CPPC0006]
Target	@WebResult
Prerequisites	A member function is annotated with @WebResult which has a type value.
Predicate	The type value of the @WebResult annotation is a valid simple type from the namespace: <a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a> .
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

98

Assertion ID	CPP-TA-C007
Source	[CPPC0007]
Target	@WebFault
Prerequisites	A class is annotated with @WebFault.
Predicate	The annotated class has a constructor with std::string parameter and a parameter with a type representing the fault information.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

Assertion ID	CPP-TA-C008
Source	[CPPC0007]
Target	@WebFault
Prerequisites	A class is annotated with @WebFault. This class satisfies CPP-TA-C007.
Predicate	The annotated class has a member function “getFaultInfo” that returns a type matching the fault information type of the constructor
Prescription Level	Mandatory
Tags	“WSDL mapping” “WSDL annotations”

100

Assertion ID	CPP-TA-C009
Source	[CPPC0008]
Target	@WebThrows
Prerequisites	A class is listed named in a @WebThrows annotation.
Predicate	The class is annotated with a @WebFault annotation.
Prescription Level	Mandatory
Tags	“WSDL mapping” “WSDL annotations”

101

Assertion ID	CPP-TA-C010
Source	[CPPC0009]
Target	@Function annotation
Prerequisites	A member function is annotated with @Function, but not @WebFunction.
Predicate	The generated WSDL is the same as if the member function was annotated with @WebFunction with a operationName value equal to the name value of the @Function annotation, an exclude value equal to the exclude value of the @Function annotation and no other parameters.
Prescription Level	Mandatory
Tags	“WSDL mapping” “WSDL annotations” “SCA annotations”

## 102 2.10 Appendix D

103 The test assertions in this section apply to the optional C++ WSDL extensions support. These assertions  
104 are only applicable if an implementation supports these extensions.

Assertion ID	CPP-TA-D001
Source	[CPPD0001]
Target	WSDL extensions
Prerequisites	A file containing valid WSDL 1.1 content and C++ mapping extensions is processed.
Predicate	A C++ header file containing declarations corresponding to the WSDL content as modified by the C++ mapping extensions is generated.

Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions"

105

Assertion ID	CPP-TA-D002
Source	[CPPD0002]
Target	<cpp:binding/>
Prerequisites	A WSDL element contains a <cpp:binding/> element.
Predicate	The <cpp:binding/> contains no more than one <cpp:class/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions"

106

Assertion ID	CPP-TA-D003
Source	[CPPD0003]
Target	<cpp:binding/>
Prerequisites	A WSDL element contains a <cpp:binding/> element.
Predicate	The <cpp:binding/> contains no more than one <cpp:enableWrapperStyle/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions"

107

Assertion ID	CPP-TA-D004
Source	[CPPD0004]
Target	<cpp:binding/>
Prerequisites	A WSDL element contains a <cpp:binding/> element.
Predicate	The <cpp:binding/> contains no more than one <cpp:namespace/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions"

108

Assertion ID	CPP-TA-D005
Source	[CPPD0005]
Target	<cpp:binding/>
Prerequisites	A WSDL element contains a <cpp:binding/> element.
Predicate	The <cpp:binding/> contains no more than one <cpp:memberFunction/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions"

109

Assertion ID	CPP-TA-D006
Source	[CPPD0006]

Target	<cpp:parameter/>
Prerequisites	A WSDL operation element contains a <cpp:parameter/> element with the @type attribute specified.
Predicate	The value of @type attribute is a valid C++ type as described in the specification <b>[SCA C++]</b> .
Prescription Level	Mandatory
Tags	“WSDL mapping” “WSDL extensions”

110

Assertion ID	CPP-TA-D007
Source	[CPPD0007]
Target	WSDL extensions
Prerequisites	A WSDL file containing JAX-WS mapping extensions is processed.
Predicate	A C++ header file containing declarations corresponding to WSDL content as modified by the JAX-WS mapping extensions is generated.
Prescription Level	Permitted
Tags	“WSDL mapping” “WSDL extensions”

111

Assertion ID	CPP-TA-D008
Source	[CPPD0007]
Target	WSDL extensions
Prerequisites	A WSDL element contains a JAX-WS WSDL extension and the corresponding C++ WSDL extension as identified in section D.7 of the specification <b>[SCA C++]</b> .
Predicate	Only the C++ extension affects the generated C++ file.
Prescription Level	Mandatory
Tags	“WSDL mapping” “WSDL extensions”

112

## 2.11 Appendix F

Assertion ID	CPP-TA-F001
Source	[CPPF0001]
Target	WSDL mapping
Prerequisites	A file containing valid WSDL 1.1 content is processed.
Predicate	A C++ header file containing declarations corresponding to the WSDL content is generated.
Prescription Level	Mandatory
Tags	“WSDL mapping”

113

Assertion ID	CPP-TA-F002
Source	[CPPF0002]
Target	Annotation generation

Prerequisites	A WSDL file is processed.
Predicate	A C++ header file containing annotated declarations corresponding to the WSDL content is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

114

Assertion ID	CPP-TA-F003
Source	[CPPF0002]
Target	Annotation generation
Prerequisites	A WSDL file containing C++ mapping extensions is processed.
Predicate	A C++ header file containing annotated declarations corresponding to the WSDL content as modified by the C++ mapping extensions is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions" "WSDL annotations"

115

Assertion ID	CPP-TA-F004
Source	[CPPF0003]
Target	WSDL import mapping
Prerequisites	A WSDL file containing WSDL and XSD import directives is processed.
Predicate	A C++ header file containing declarations corresponding to the imported elements content is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

116

Assertion ID	CPP-TA-F005
Source	[CPPF0005]
Target	portType mapping
Prerequisites	A WSDL <portType/> element does not contain a <cpp:class/> element
Predicate	The name of the generated class is the name of the <portType/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping"

117

Assertion ID	CPP-TA-F006
Source	[CPPF0006]
Target	portType mapping
Prerequisites	
Predicate	The generated class corresponding to a <portType/> element has a @WebService annotation with a portName element value equal to the name of the <port/> element of the WSDL definition.



Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

118

Assertion ID	CPP-TA-F007
Source	[CPPF0007]
Target	Operation mapping
Prerequisites	A WSDL <operation/> element does not contain a <cpp:memberFunction/> element.
Predicate	The name of the generated member function is the name of the <operation/> element with the first character converted to lower case.
Prescription Level	Mandatory
Tags	"WSDL mapping"

119

Assertion ID	CPP-TA-F008
Source	[CPPF0008]
Target	Operation mapping
Prerequisites	A WSDL <operation/> element contains a <cpp:memberFunction/> element specifying a non-default name or the <soapAction/> element of the corresponding binding operation is not an empty string.
Predicate	The generated member function corresponding to the <operation/> element has a @WebFunction annotation.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

120

Assertion ID	CPP-TA-F009
Source	[CPPF0009]
Target	Operation mapping
Prerequisites	A WSDL <operation/> element uses a request-response or one-way transmission primitive.
Predicate	A member function is generated for the <operation/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping"

121

Assertion ID	CPP-TA-F010
Source	[CPPF0010]
Target	Operation mapping
Prerequisites	A WSDL <operation/> element uses a one-way transmission primitive.
Predicate	The generated member function corresponding to the <operation/> element has a @Oneway annotation.
Prescription Level	Mandatory

Tags	"WSDL mapping" "WSDL annotations"
------	-----------------------------------

122

Assertion ID	CPP-TA-F011
Source	[CPPF0011]
Target	portType mapping
Prerequisites	A WSDL <portType/> element is bound to a SOAP binding that does not have style="Document", use="literal" and wrapped parameters.
Predicate	The generated class corresponding to a <portType/> element has a @SOAPBinding annotation.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

123

Assertion ID	CPP-TA-F012
Source	[CPPF0012]
Target	Message mapping
Prerequisites	A WSDL <operation/> element uses a <message/> element which has content that causes non-default mapping of a part or wrapper child's name, mode, type, namespace or inclusion in a header.
Predicate	The generated member function corresponding to the <operation/> element has a @WebParam annotation for the parameter.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

124

Assertion ID	CPP-TA-F013
Source	[CPPF0013]
Target	Message mapping
Prerequisites	A WSDL <operation/> element uses a <message/> element which has content that causes non-default mapping of the name, type, namespace or inclusion in a header of the part or wrapper child that maps to a member function's return type.
Predicate	The generated member function corresponding to the <operation/> element has a @WebResult annotation.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

125

Assertion ID	CPP-TA-F014
Source	[CPPF0014]
Target	Message mapping
Prerequisites	A WSDL <operation/> element that does not qualify for wrapper-style refers to a <message/> element that does not contain a <cpp:parameter/> element.
Predicate	The name of each generated member function parameter is the name of the corresponding <part/> element of the <message/> element with the first character converted to lower case.

Prescription Level	Mandatory
Tags	"WSDL mapping"

126

Assertion ID	CPP-TA-F015
Source	[CPPF0015]
Target	Message mapping
Prerequisites	A WSDL <operation/> element does not qualify for wrapper-style.
Predicate	The generated member function parameters are mapped following from the <part/> elements of the <message/> elements referenced by the <operation/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping"

127

Assertion ID	CPP-TA-F016
Source	[CPPF0016]
Target	Operation mapping
Prerequisites	A WSDL <operation/> element contains a <cpp:enableWrapperStyle/> element.
Predicate	The generated member function parameters are mapped from wrapper children if the value of the <cpp:enableWrapperStyle> is true and from <part/> elements otherwise.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL extensions"

128

Assertion ID	CPP-TA-F017
Source	[CPPF0017]
Target	Message mapping
Prerequisites	A WSDL <operation/> element that qualifies for wrapper-style refers to a <message/> element that does not contain a <cpp:parameter/> element.
Predicate	The name of each generated member function parameter is the local name of the corresponding wrapper child of the <message/> element with the first character converted to lower case.
Prescription Level	Mandatory
Tags	"WSDL mapping"

129

Assertion ID	CPP-TA-F018
Source	[CPPF0018]
Target	Message mapping
Prerequisites	A WSDL <operation/> element has more than one wrapper child or <part/> element of <message/> elements referred to by the <operation/> element that do not map to the return type of the generated member function.
Predicate	Each wrapper child or <part/> element maps to a unique generated member function parameter.

Prescription Level	Mandatory
Tags	"WSDL mapping"

130

Assertion ID	CPP-TA-F019
Source	[CPPF0019]
Target	Fault message mapping
Prerequisites	A WSDL <operation/> element contains a <fault/> element.
Predicate	The generated class corresponding to the <message/> element referred to by the <fault/> element has a @WebFault annotation and the member function generated for the <operation/> element has a @WebThrows annotation naming the generated class corresponding to the <message/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

131

Assertion ID	CPP-TA-F020
Source	[CPPF0020]
Target	Fault message mapping
Prerequisites	A <fault/> element of a WSDL <operation/> element refers to a <message/> element that does not contain a <cpp:class/> element.
Predicate	The name of the generated class is the name of the <message/> element with "_Exception" appended if necessary to avoid a name collision with a class generated from a <portType/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping"

132

Assertion ID	CPP-TA-F022
Source	[CPPF0022]
Target	Binding mapping
Prerequisites	A file containing a <soap:binding> is processed.
Predicate	The generated C++ header file contains declarations and annotations corresponding to the <soap:binding/> element content.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

133

Assertion ID	CPP-TA-F023
Source	[CPPF0023]
Target	Binding mapping
Prerequisites	A WSDL <message/> element contains <part/> elements that are not bound.
Predicate	The SCA Web Service binding ignores unbound in and in/out parameters and presents unbound out parameters as empty.

Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

134

Assertion ID	CPP-TA-F024
Source	[CPPF0024]
Target	Binding mapping
Prerequisites	A <operation/> element of a <soap:binding/> element contains more that one <soap:header/> element.
Predicate	Each SOAP header has a unique qualified name.
Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

135

Assertion ID	CPP-TA-F025
Source	[CPPF0025]
Target	Binding mapping
Prerequisites	A request message contains more that one <soap:header/> element.
Predicate	Each SOAP header mapped to a member function parameter has a unique qualified name.
Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

136

Assertion ID	CPP-TA-F026
Source	[CPPF0026]
Target	WSDL mapping
Prerequisites	A C++ file containing a remotable SCA interface description is processed.
Predicate	A WSDL 1.1 file containing definitions corresponding to the C++ declarations is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping"

137

Assertion ID	CPP-TA-F027
Source	[CPPF0027]
Target	WSDL mapping
Prerequisites	A C++ header file does not contain any WSDL mapping annotations
Predicate	All C++ identifiers are mapped to XML names as per the SOAP 1.2 algorithm.
Prescription Level	Mandatory
Tags	"WSDL mapping"

138

Assertion ID	CPP-TA-F028
Source	[CPPF0028]
Target	Member function mapping
Prerequisites	A C++ class contains overloaded member functions.
Predicate	The @WebFunction annotation is used to disambiguate the <operation/> element name mapped from the overloaded member functions.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

139

Assertion ID	CPP-TA-F029
Source	[CPPF0029]
Target	WSDL Mapping
Prerequisites	
Predicate	Any WSDL or XSD imports comply with the WS-I Basic Profile 1.0 restrictions.
Prescription Level	Mandatory
Tags	"WSDL mapping"

140

Assertion ID	CPP-TA-F030
Source	[CPPF0030]
Target	Class Mapping
Prerequisites	A class does not have an @WebService annotation
Predicate	The name of the generated <portType/> element is the name of the class.
Prescription Level	Mandatory
Tags	"WSDL mapping"

141

Assertion ID	CPP-TA-F031
Source	[CPPF0044]
Target	Class Mapping
Prerequisites	A class defining an SCA interface is a derived class.
Predicate	The generated <portType/> elements contains <operation/> elements corresponding any inherited member functions of the class.
Prescription Level	Mandatory
Tags	"WSDL mapping" "Subclasses"

142

Assertion ID	CPP-TA-F032
Source	[CPPF0045]
Target	Class Mapping
Prerequisites	A class defining an SCA interface is a derived class.

Predicate	Inherited classes are mapped to separate <portType/> elements in the WSDL definition.
Prescription Level	Permitted
Tags	"WSDL mapping" "Subclasses"

143

Assertion ID	CPP-TA-F033
Source	[CPPF0031]
Target	Member function mapping
Prerequisites	A member function does not have a @WebFunction annotation
Predicate	The name of the generated <operation/> element is the name of the member function.
Prescription Level	Mandatory
Tags	"WSDL mapping"

144

Assertion ID	CPP-TA-F034
Source	[CPPF0031]
Target	Member function mapping
Prerequisites	A member function has a @WebFunction annotation with an exclude value of true.
Predicate	There is no generated <operation/> element.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

145

Assertion ID	CPP-TA-F035
Source	[CPPF0032]
Target	Member function mapping
Prerequisites	A member function does not have a @OneWay annotation.
Predicate	The generated <operation/> element uses a request-response transmission primitive.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

146

Assertion ID	CPP-TA-F036
Source	[CPPF0033]
Target	Member function mapping
Prerequisites	A member function has a @OneWay annotation and either has out or in/out parameters or has a return type that is not void.
Predicate	No <operation/> element is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping" "WSDL annotations"

147

Assertion ID	CPP-TA-F037
Source	[CPPF0035]
Target	Parameter mapping
Prerequisites	A member function has an unnamed parameter.
Predicate	The name of the generated <part/> element of the <message/> or wrapper child is "argN" where N is the cardinal position of the parameter.
Prescription Level	Mandatory
Tags	"WSDL mapping"

148

Assertion ID	CPP-TA-F038
Source	[CPPF0035]
Target	Parameter mapping
Prerequisites	A member function has a parameter that does not have a @WebParam annotation with a partName value.
Predicate	The name of the generated <part/> element of the <message/> or wrapper child is the name of the parameter.
Prescription Level	Mandatory
Tags	"WSDL mapping"

149

Assertion ID	CPP-TA-F039
Source	[CPPF0036]
Target	Member function mapping
Prerequisites	A member function does not have a @WebResult annotation with a part name value.
Predicate	The name of the generated <part/> element of the <message/> or wrapper child is "return".
Prescription Level	Mandatory
Tags	"WSDL mapping"

150

Assertion ID	CPP-TA-F040
Source	[CPPF0037]
Target	Parameter mapping
Prerequisites	A member function has a parameter that has a @WebParam annotation with a header value of true.
Predicate	The parameter appears in a <soap:header/> element of the operation's input or output message.
Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

151

Assertion ID	CPP-TA-F041
Source	[CPPF0037]



Target	Member function mapping
Prerequisites	A member function has a @WebResult annotation with a header value of true.
Predicate	The return value appears in a <soap:header/> element of the operation's output message.
Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

152

Assertion ID	CPP-TA-F043
Source	[CPPF0039]
Target	Binding selection
Prerequisites	
Predicate	A <binding/> element is generated that reflect any policy intents applied to a interface definition.
Prescription Level	Mandatory
Tags	"WSDL mapping"

153

Assertion ID	CPP-TA-F044
Source	[CPPF0040]
Target	Binding selection
Prerequisites	A C++ interface definition has no attached policy intents that affect the binding selection.
Predicate	A SOAP HTTP <binding/> element is generated.
Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

154

Assertion ID	CPP-TA-F045
Source	[CPPF0041]
Target	Binding selection
Prerequisites	A SOAP HTTP <binding/> element is generated
Predicate	The <binding/> element contains a @style attribute
Prescription Level	Mandatory
Tags	"WSDL mapping" "SOAP binding"

155

Assertion ID	CPP-TA-F046
Source	[CPPF0042]
Target	Port mapping
Prerequisites	A class does not have a @WebService annotation with a portName value.
Predicate	The name of the generated <port/> element is the name value of the @WebService annotation, if present, with "Port" appended , otherwise it is the name of the class with

	"Port" appended.
Prescription Level	Mandatory
Tags	"WSDL mapping"

156

Assertion ID	CPP-TA-F047
Source	[CPPF0042]
Target	Port mapping
Prerequisites	A class has a @WebService annotation with a portName value.
Predicate	The name of the generated <port/> element is the portName value of the @WebService annotation.
Prescription Level	Mandatory
Tags	"WSDL mapping"

157

Assertion ID	CPP-TA-F048
Source	[CPPF0043]
Target	Port mapping
Prerequisites	A C++ interface definition has no attached policy intents that affect the binding selection.
Predicate	The <port/> element refers to a <binding/> element that is consistent with the policy intents.
Prescription Level	Mandatory
Tags	"WSDL mapping"

158

---

159 **3 Conformance**

160 There are no conformance statements relating to the Test Assertions.

161  
162

## A. Cross Mapping of Normative Statements to Assertions

Normative statement	Test Assertion
CPP20001	CPP-TA-2001
CPP20003	CPP-TA-2002
CPP20005	CPP-TA-2006
CPP20006	CPP-TA-2007
CPP20007	CPP-TA-2008
CPP20008	CPP-TA-2009
CPP20009	CPP-TA-2010
CPP20010	CPP-TA-2011
CPP20011	CPP-TA-2012
CPP20012	CPP-TA-2003 CPP-TA-2004
CPP20013	CPP-TA-2005
CPP20014	CPP-TA-2013
CPP20015	CPP-TA-2014
CPP20016	CPP-TA-2015
CPP20017	CPP-TA-2016
CPP20018	CPP-TA-2017

163

Normative statement	Test Assertion
CPP30001	CPP-TA-3001
CPP30002	CPP-TA-3002
CPP30003	CPP-TA-3003
CPP30004	CPP-TA-3004

164

Normative statement	Test Assertion
CPP40001	CPP-TA-4001
CPP40002	CPP-TA-4002
CPP40003	CPP-TA-4003

165

Normative statement	Test Assertion
---------------------	----------------

Normative statement	Test Assertion
CPP70001	CPP-TA-7001
CPP70002	CPP-TA-7002

166

Normative statement	Test Assertion
CPP80001	CPP-TA-8001
CPP80002	CPP-TA-8002
CPP80003	CPP-TA-8003
	CPP-TA-8004

167

Normative statement	Test Assertion
CPP100001	CPP-TA-10001
	CPP-TA-10002
CPP100002	CPP-TA-10003
CPP100003	CPP-TA-10004
CPP100004	CPP-TA-10005
CPP100005	CPP-TA-10006
CPP100006	CPP-TA-10007
CPP100008	CPP-TA-10009
	CPP-TA-10010
CPP-100009	CPP-TA-10011
CPP-100010	CPP-TA-10012

168

Normative statement	Test Assertion
CPP110001	CPP-TA-11001
CPP110002	CPP-TA-11002
CPP110003	CPP-TA-11004
CPP110004	CPP-TA-11005

169

Normative statement	Test Assertion
---------------------	----------------

Normative statement	Test Assertion
CPPA0001	CPP-TA-A001 CPP-TA-A002
CPPA0002	CPP-TA-A003
CPPA0003	CPP-TA-A004
CPPA0004	CPP-TA-A005

170

Normative statement	Test Assertion
CPPC0001	CPP-TA-C001
CPPC0002	CPP-TA-C002
CPPC0004	CPP-TA-C004
CPPC0005	CPP-TA-C005
CPPC0006	CPP-TA-C006
CPPC0007	CPP-TA-C007 CPP-TA-C008
CPPC0008	CPP-TA-C009
CPPC0009	CPP-TA-C010

171

Normative statement	Test Assertion
CPPD0001	CPP-TA-D001
CPPD0002	CPP-TA-D002
CPPD0003	CPP-TA-D003
CPPD0004	CPP-TA-D004
CPPD0005	CPP-TA-D005
CPPD0006	CPP-TA-D006
CPPD0007	CPP-TA-D007 CPP-TA-D008

172

Normative statement	Test Assertion
---------------------	----------------

Normative statement	Test Assertion
CPPF0001	CPP-TA-F001
CPPF0002	CPP-TA-F002 CPP-TA-F003
CPPF0003	CPP-TA-F004
CPPF0004	Not Testable
CPPF0005	CPP-TA-F005
CPPF0006	CPP-TA-F006
CPPF0007	CPP-TA-F007
CPPF0008	CPP-TA-F008
CPPF0009	CPP-TA-F009
CPPF0010	CPP-TA-F010
CPPF0011	CPP-TA-F011
CPPF0012	CPP-TA-F012
CPPF0013	CPP-TA-F013
CPPF0014	CPP-TA-F014
CPPF0015	CPP-TA-F015
CPPF0016	CPP-TA-F016
CPPF0017	CPP-TA-F017
CPPF0018	CPP-TA-F018
CPPF0019	CPP-TA-F019
CPPF0020	CPP-TA-F020
CPPF0022	CPP-TA-F022
CPPF0023	CPP-TA-F023
CPPF0024	CPP-TA-F024
CPPF0025	CPP-TA-F025
CPPF0026	CPP-TA-F026
CPPF0027	CPP-TA-F027
CPPF0028	CPP-TA-F028
CPPF0029	CPP-TA-F029
CPPF0030	CPP-TA-F030
CPPF0031	CPP-TA-F033 CPP-TA-F034
CPPF0032	CPP-TA-F035
CPPF0033	CPP-TA-F036

Normative statement	Test Assertion
CPPF0035	CPP-TA-F037 CPP-TA-F038
CPPF0036	CPP-TA-F039
CPPF0037	CPP-TA-F040 CPP-TA-F041
CPPF0039	CPP-TA-F043
CPPF0040	CPP-TA-F044
CPPF0041	CPP-TA-F045
CPPF0042	CPP-TA-F046 CPP-TA-F047
CPPF0043	CPP-TA-F048
CPPF0044	CPP-TA-F031
CPPF0045	CPP-TA-F032

173



---

174 **B. Acknowledgements**

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

177 **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

178