



# TestCases for the SCA Policy Framework Version 1.1 Specification

## Committee Specification Draft 02 / Public Review Draft 02

**29 November 2010**

### Specification URIs:

#### This Version:

<http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-testcases-csprd02.html>  
<http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-testcases-csprd02.odt>  
<http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-testcases-csprd02.pdf>  
(Authoritative)

#### Previous Version:

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

#### Latest Version:

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

### Technical Committee:

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

### Chair(s):

[David Booz](#), IBM  
[Ashok Malhotra](#), Oracle

### Editor(s):

[David Booz](#), IBM

### Related Work:

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

<http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-testcases-csprd02.zip>

This document is related to

- [Service Component Architecture Policy Framework Specification Version 1.1](#)

**Declared XML Namespace(s):**

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

**Abstract:**

This document defines the TestCases for the SCA Policy specification.

The TestCases represent a series of tests that an SCA runtime must pass in order to claim conformance to the requirements of the SCA Policy specification.

**Status:**

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

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-policy/>.

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-policy/ipr.php>).

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

**Citation Format:**

When referencing this specification the following citation format should be used:

SCA-Policy-v1.1    OASIS Committee Specification Draft 02 / Public Review Draft 02, *TestCases for the SCA Policy Framework Version 1.1 Specification*, November 2010.  
<http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-testcases-csprd02.odt>

---

# 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](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 TestCase Structure.....	5
1.2 Namespaces and Java Package Names.....	7
1.3 Terminology.....	7
1.4 Normative References.....	7
1.5 Non-Normative References.....	8
2 TestCases.....	9
2.1 Section 3.....	9
2.2 Section 4.....	17
2.3 Section 5.....	29
2.4 Section 9.....	29
2.5 Section 10.....	35
2.6 Section 11.....	35
3 Cross Mapping of Test Assertions to TestCases.....	36
4 Catalog of Test Artifacts.....	41
4.1 Composite Files - lower level.....	41
4.2 Java Interfaces – General_Java contribution.....	41
4.3 Java Implementation Classes - General_Java.....	41
4.4 WSDL Interface Files - General.....	41
4.5 Policy files - General.....	42
5 Testcases Grouped by Optional Capability.....	43
6 Conformance.....	45
Appendix A: Acknowledgments.....	46
Appendix B: Revision History.....	48

---

# 1 Introduction

This document defines the TestCases for the SCA Policy Framework specification.

The tests described in this document are related to the Test Assertions described in the SCA Policy Test Assertions document [POLICY-TA].

## 1.1 TestCase Structure

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

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

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

### Test Client

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

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

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

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

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

The Test Client defaults to using Web services to communicate with the test application. The client is structured to permit Web services to be replaced by some other binding (e.g. JMS) should the SCA runtime under test not support Web services as a binding technology.

## Test Application

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

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

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

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

Which implementation language to use for test artifacts is controlled by a system variable or environment variable which is read by the test client application, with the name "OASIS\_TESTENV\_IMPL\_LANG". This variable can have one of the following values:

- "Java" - for Java implementations

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

## Test Artifacts Organization

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

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

Broadly, Contribution names are as follows:

- POL\_nnnn - a contribution that is specific for a particular testcase, where "nnnn" is the number of the testcase. Often this is required because a particular testcase involves artifacts that contain errors that are statically checkable - an SCA runtime is permitted to reject such artifacts when they are contributed and deployed and it is important to ensure that contributions containing deliberate errors for one testcase do not interfere with the operation of other testcases.
- General - a shared contribution containing implementation type independent artifacts that can be used by many testcases.
- General\_Java - a shared contribution containing implementation type dependent artifacts for Java POJOs. These artifacts can include both Java classes and also SCA composites that directly use Java classes.

Note that the names of Contributions containing implementation specific artifacts ends with a name that is specific to the implementation type - so "\_Java" is used for Java implementations. Note that the name following the underscore matches the name used in the "OASIS\_TESTENV\_IMPL\_LANG" variable used to control execution of the test client. The concept is that where there is an implementation type specific contribution, each implementation type must provide its own versions of the same basic artifacts. Typically, this means that each contribution must contain the same set of Composites, but that the implementation type dependent artifacts that these composites use will differ from implementation type to implementation type.

Basically, the setting of the variable is used to select the suffix used for implementation type dependent contributions. If the variable is set to "Java" then the contribution "General\_Java" is selected, whereas if the variable is set to "BPEL", the contribution "General\_BPEL" is selected.

## TestCase Groups

The SCA Policy FW specification [POLICY] contains some optional capabilities in the way that policySets are attached to SCA composites. Specifically, policySets can be attached directly, externally or both ways. An SCA runtime has the freedom to choose which of those three sets of capability to implement. There are three groups of testcases which match each set of capability. In order to claim compliance to the test suite, an SCA implementation has to pass all the tests in at least one of the three groups. The three groups are defined in Section "Testcases Grouped by Optional Capability".

## 1.2 Namespaces and Java Package Names

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

### SCA Artifact Namespaces

These apply to artifacts such as Composites

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

### WSDL Namespace

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

### Java Package name

For Java interface classes and for Java implementation classes

`org.oasisopen.sca.test`

## 1.3 Terminology

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

## 1.4 Normative References

- [RFC 2119] S. Bradner. "Key words for use in RFCs to Indicate Requirement Levels". IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- [POLICY-TA] OASIS Committee Draft 02, "Test Assertions for the SCA Policy Framework 1.1 Specification", September 2010. <http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-test-assertions-CD02.pdf>
- [POLICY] OASIS Committee Draft 04, "SCA Policy Framework 1.1 Specification", September 2010. <http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1-spec-cd04.pdf>

## 1.5 Non-Normative References

- [TBD] [TBD]



---

## 2 TestCases

### 2.1 Section 3

#### POL\_3001\_TestCase

Testcase ID	POL_3001_TestCase
Test Assertion	POL-TA-30001
Description	Tests that the runtime rejects a composite where the use of an intent conflicts with the configuration of the binding itself.
Artifacts	POL_3001_TestCase.java Test_POL_3001.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite Service1.wsdl Service1.java service1Impl.java
Expected output	Negative test: "exception"


#### POL\_3003\_TestCase

Testcase ID	POL_3003_TestCase
-------------	-------------------

Test Assertion	POL-TA-30003
Description	Tests that the runtime rejects a definitions file when it contains duplicate intent definitions.
Artifacts	POL_3003_TestCase.java Test_POL_3003.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3004\_TestCase

Testcase ID	POL_3004_TestCase
Test Assertion	POL-TA-30004
Description	Tests that the runtime rejects a definitions file when it contains an intent definition which has two default qualifiers.
Artifacts	POL_3004_TestCase.java Test_POL_3004.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3005\_TestCase

Testcase ID	POL_3005_TestCase
Test Assertion	POL-TA-30005
Description	Tests that the runtime rejects a definitions file when it contains an intent definition which has two duplicate qualifiers.
Artifacts	POL_3005_TestCase.java Test_POL_3005.composite

	TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3006\_TestCase

Testcase ID	POL_3006_TestCase
Test Assertion	POL-TA-30006
Description	Tests that the runtime rejects a definitions file when it contains a profile intent definition that contains a '.' in its name.
Artifacts	POL_3006_TestCase.java Test_POL_3006.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3007\_TestCase

Testcase ID	POL_3007_TestCase
Test Assertion	POL-TA-30007
Description	Tests that when a profile intent is required, that the attached policySet(s) satisfy all the intents required by the profile intent.
Artifacts	POL_3007_TestCase.java Test_POL_3007.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_3007 request no invocation"

## POL\_3008\_TestCase

Testcase ID	POL_3008_TestCase
Test Assertion	POL-TA-30008
Description	Tests that an intentMap provides an unqualified intent that the policySet provides.
Artifacts	POL_3008_TestCase.java Test_POL_3008.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_3008 request no invocation"

## POL\_3009\_TestCase

Testcase ID	POL_3009_TestCase
Test Assertion	POL-TA-30010
Description	Tests that a policySet only has one intentMap for any given qualifiable intent.
Artifacts	POL_3009_TestCase.java Test_POL_3009.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3011\_TestCase

Testcase ID	POL_3011_TestCase
-------------	-------------------

Test Assertion	POL-TA-30013
Description	Tests that a referenced policySet provides a subset of the intents provided by the referencing policySet.
Artifacts	POL_3011_TestCase.java Test_POL_3011.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_3011 request no invocation"

## POL\_3012\_TestCase

Testcase ID	POL_3012_TestCase
Test Assertion	POL-TA-30015
Description	Tests that the runtime rejects a definitions file when it contains a profile intent definition that requires undefined intents.
Artifacts	POL_3012_TestCase.java Test_POL_3012.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3013\_TestCase

Testcase ID	POL_3013_TestCase
Test Assertion	POL-TA-30016
Description	Tests that the runtime rejects a definitions file when it contains a profile intent definition that excludes an undefined intent.
Artifacts	POL_3013_TestCase.java Test_POL_3013.composite

	TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3014\_TestCase

Testcase ID	POL_3014_TestCase
Test Assertion	POL-TA-30017
Description	Tests that the runtime ensures that policySet definitions are unique in the Domain.
Artifacts	POL_3014_TestCase.java Test_POL_3014.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3015\_TestCase

Testcase ID	POL_3015_TestCase
Test Assertion	POL-TA-30018
Description	Tests that the XPath expression in policySet/@appliesTo is valid.
Artifacts	POL_3015_TestCase.java Test_POL_3015.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3016\_TestCase

Testcase ID	POL_3016_TestCase
Test Assertion	POL-TA-30019
Description	Tests that the XPath expression in policySet/@attachTo is valid.
Artifacts	POL_3016_TestCase.java Test_POL_3016.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3017\_TestCase

Testcase ID	POL_3017_TestCase
Test Assertion	POL-TA-30020
Description	Tests that an intentMap specifies all possible qualifiers for a qualifiable intent.
Artifacts	POL_3017_TestCase.java Test_POL_3017.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3018\_TestCase

Testcase ID	POL_3018_TestCase
Test Assertion	POL-TA-30021

Description	Tests that an intentMap provides an intent that is provided by the containing policySet.
Artifacts	POL_3018_TestCase.java Test_POL_3018.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_3019\_TestCase

Testcase ID	POL_3019_TestCase
Test Assertion	POL-TA-30025
Description	Tests that intents defined in the Policy spec are in the SCA Domain.
Artifacts	POL_3019_TestCase.java Test_POL_3019.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_3019 request no invocation"

## POL\_3020\_TestCase

Testcase ID	POL_3020_TestCase
Test Assertion	POL-TA-30026
Description	Tests that an intent definition with one qualifier has the qualifier marked as a default.
Artifacts	POL_3020_TestCase.java Test_POL_3020.composite TestInvocation.wsdl TestClient_0004.composite



	definitions.xml
Expected output	Positive test: "POL_3020 request no invocation"

## 2.2 Section 4

### POL\_4001\_TestCase

Testcase ID	POL_4001_TestCase
Test Assertion	POL-TA-40001
Description	Tests that directly attached policySets are ignored when there are externally attached policySets.
Artifacts	POL_4001_TestCase.java Test_POL_4001.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative: "exception"

### POL\_4002\_TestCase

Testcase ID	POL_4002_TestCase
Test Assertion	POL-TA-40002
Description	Tests that directly attached policySets are supported
Artifacts	POL_4002_TestCase.java Test_POL_4002.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4002 request no invocation"

## POL\_4003\_TestCase

Testcase ID	POL_4003_TestCase
Test Assertion	POL-TA-40003
Description	Tests that externally attached policy sets are supported.
Artifacts	POL_4003_TestCase.java Test_POL_4003.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4003 request no invocation"

## POL\_4004\_TestCase

Testcase ID	POL_4004_TestCase
Test Assertion	POL-TA-40004
Description	Tests that qualified intents override qualifiable intents in the implementation hierarchy.
Artifacts	POL_4004_TestCase.java Test_POL_4004.composite TestCompositeImplIntent.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4005\_TestCase

Testcase ID	POL_4005_TestCase
Test Assertion	POL-TA-40005
Description	Tests that qualified intents override qualifiable intents in the implementation hierarchy of service promotion.
Artifacts	POL_4005_TestCase.java Test_POL_4005.composite TestCompositeImplIntent.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

### POL\_4006\_TestCase

Testcase ID	POL_4006_TestCase
Test Assertion	POL-TA-40006
Description	Tests that intents higher in the structural hierarchy are unioned with intents lower in the structural hierarchy.
Artifacts	POL_4006_TestCase.java Test_POL_4006.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

### POL\_4007\_TestCase

Testcase ID	POL_4007_TestCase
Test Assertion	POL-TA-40007
Description	Tests that intents higher in the structural hierarchy are unioned with non-mutually exclusive intents lower in the structural hierarchy.

Artifacts	POL_4007_TestCase.java Test_POL_4007.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4007 request service1 operation1 invoked"

## POL\_4008\_TestCase

Testcase ID	POL_4008_TestCase
Test Assertion	POL-TA-40008
Description	Tests that qualified intents in the structural hierarchy override the unqualified form of the intent.
Artifacts	POL_4008_TestCase.java Test_POL_4008.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4008 request service1 operation1 invoked"

## POL\_4009\_TestCase

Testcase ID	POL_4009_TestCase
Test Assertion	POL-TA-40009
Description	Tests that componentType attached policySets are ignored when policySets are attached to the using component definition.
Artifacts	POL_4009_TestCase.java Test_POL_4009.composite TestCompositeImplPolicySet.composite TestInvocation.wsdl

	TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4010\_TestCase

Testcase ID	POL_4010_TestCase
Test Assertion	POL-TA-40015
Description	Tests that directly attached policySets are ignored when the runtime does not support directly attached policySets and there are externally attached policySet(s).
Artifacts	POL_4010_TestCase.java Test_POL_4010.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4012\_TestCase

Testcase ID	POL_4012_TestCase
Test Assertion	POL-TA-40012
Description	A testcase to verify that mutually exclusive intents cause an error.
Artifacts	POL_4012_TestCase.java Test_POL_4012.composite TestInvocation.wsdl TestClient_0004.composite
Expected output	Negative test: "exception"

## POL\_4013\_TestCase

Testcase ID	POL_4013_TestCase
Test Assertion	POL-TA-40014
Description	A testcase to verify that external attachment of policySets works correctly.
Artifacts	POL_4013_TestCase.java Test_POL_4013.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4013 request no invocation"

## POL\_4015\_TestCase

Testcase ID	POL_4015_TestCase
Test Assertion	POL-TA-40013
Description	Tests that directly attached policySets are ignored when the runtime does not support directly attached policySets.
Artifacts	POL_4015_TestCase.java Test_POL_4015.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4016\_TestCase

Testcase ID	POL_4016_TestCase
-------------	-------------------

Test Assertion	POL-TA-40016
Description	Tests that externally attached policySets are ignored when the runtime does not support them.
Artifacts	POL_4016_TestCase.java Test_POL_4016.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4018\_TestCase

Testcase ID	POL_4018_TestCase
Test Assertion	POL-TA-40018
Description	Tests that externally attached policySets are ignored when the runtime does not support them, and there are directly attached policySets that are supported.
Artifacts	POL_4018_TestCase.java Test_POL_4018.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4019\_TestCase

Testcase ID	POL_4019_TestCase
Test Assertion	POL-TA-40024
Description	Tests that policySets on interfaces apply to bindings and services.
Artifacts	POL_4019_TestCase.java Test_POL_4019.composite

	TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4019 request no invocation"

## POL\_4020\_TestCase

Testcase ID	POL_4020_TestCase
Test Assertion	POL-TA-40025
Description	Tests that policySets on interfaces apply to bindings and services.
Artifacts	POL_4020_TestCase.java Test_POL_4020.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Positive test: "POL_4020 request no invocation"

## POL\_4024\_TestCase

Testcase ID	POL_4024_TestCase
Test Assertion	POL-TA-40027, POL-TA-40028
Description	Tests that when an intent is required, that the attached policySet(s) satisfies the intent.
Artifacts	POL_4024_TestCase.java Test_POL_4024.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite Service1.java



	Service1.wsdl definitions.xml
Expected output	Positive test: "POL_4024 request service1 operation1 invoked"

## POL\_4027\_TestCase

Testcase ID	POL_4027_TestCase
Test Assertion	POL-TA-40039
Description	BindingTypes are unique in the Domain.
Artifacts	POL_4027_TestCase.java Test_POL_4027.composite TestInvocation.wsdl TestClient_0004.composite Service1.wsdl definitions.xml
Expected output	Positive test: "exception"

## POL\_4028\_TestCase

Testcase ID	POL_4028_TestCase
Test Assertion	POL-TA-40049
Description	A testcase to verify that an error is raised if policySet/@attachTo points to a property.
Artifacts	POL_4028_TestCase.java Test_POL_4028.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_4029\_TestCase

Testcase ID	POL_4029_TestCase
Test Assertion	POL-TA-40020
Description	Tests that where a <component/> has an <implementation/> element which has a service with an attached intent, that the intent applies to the binding of the <component/> <service/> and must be satisfied by that binding.
Artifacts	POL_4029_TestCase.java Test_POL_4029.composite TestInvocation.wsdl Service1.wsdl TestClient_0004.composite TestCompositeSOAP.composite
Expected output	Positive test: "POL_4029 request service1 operation1 invoked"

## POL\_4030\_TestCase

Testcase ID	POL_4030_TestCase
Test Assertion	POL-TA-40021
Description	Tests that where a <component/> has an <implementation/> element which has a service with an attached unqualified form of an intent and the qualified form of the intent is attached to <service/> element of the component, that the qualified form of the intent applies to the <component/> <service/> and must be satisfied by its binding.
Artifacts	POL_4030_TestCase.java Test_POL_4030.composite TestInvocation.wsdl Service1.wsdl TestClient_0004.composite TestCompositeSOAP.composite
Expected output	Positive test: "POL_4030 request service1 operation1 invoked"

## POL\_4031\_TestCase

Testcase ID	POL_4031_TestCase
Test Assertion	POL-TA-40022 POL-TA-40037 POL-TA-40050
Description	Tests that where a <component/> has an <service/> element which has a <interface/> subelement with an intent attached to it, that the intent applies to the <component/> <service/> and must be satisfied by its binding.
Artifacts	POL_4031_TestCase.java Test_POL_4031.composite TestInvocation.wsdl TestClient_0004.composite Service1.wsdl Service1withSOAPIntent.wsdl TestComposite1.composite
Expected output	Positive test: "POL_4031 request service1 operation1 invoked"

## POL\_4032\_TestCase

Testcase ID	POL_4032_TestCase
Test Assertion	POL-TA-40023 POL-TA-40038 POL_TA_40050
Description	Tests that where a <component/> has an <reference/> element which has a <interface/> subelement with an intent attached to it, that the intent applies to the <component/> <reference/> and must be satisfied by its binding.
Artifacts	POL_4032_TestCase.java Test_POL_4032.composite TestInvocation.wsdl

	TestClient_0004.composite Service1.wsdl Service1withSOAPIntent.wsdl TestComposite1.composite
Expected output	Positive test: "POL_4032 request service1 operation1 invoked"

## POL\_4033\_TestCase

Testcase ID	POL_4033_TestCase
Test Assertion	POL-TA-40058
Description	Tests that the @attachTo attribute of an <externalAttachment/> element is a valid XPath 1.0 production expression
Artifacts	POL_4033_TestCase.java Test_POL_4033.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml in POL_4033 contribution
Expected output	Negative test: "exception"

## 2.3 Section 5

### POL\_5001\_TestCase

Testcase ID	POL_5001_TestCase
Test Assertion	POL-TA-50001
Description	A testcase to verify that the runtime can detect unknown implementation types.
Artifacts	POL_5001_TestCase.java Test_POL_5001.composite TestInvocation.wsdl TestClient_0004.composite

	definitions.xml
Expected output	Negative test: "exception"

## 2.4 Section 9

### POL\_9006\_TestCase

Testcase ID	POL_9006_TestCase
Test Assertion	POL-TA-90009, POL-TA-90036
Description	A testcase to verify that a reference with transactedOneWay on a non-global tran component, results in an error from the runtime.
Artifacts	POL_9006_TestCase.java Test_POL_9006.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

### POL\_9009\_TestCase

Testcase ID	POL_9009_TestCase
Test Assertion	POL-TA-90012, POL-TA-90030
Description	A testcase to verify that a service with transactedOneWay on a non-global tran component, results in an error from the runtime.
Artifacts	POL_9009_TestCase.java Test_POL_9009.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite

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

## POL\_9015\_TestCase

Testcase ID	POL_9015_TestCase
Test Assertion	POL-TA-90021
Description	A testcase to verify that a component with managedTransaction.local and a service with propagatesTransaction, results in an error from the runtime.
Artifacts	POL_9015_TestCase.java Test_POL_9015.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite Service1.wsdl
Expected output	Negative test: "exception"

## POL\_9016\_TestCase

Testcase ID	POL_9016_TestCase
Test Assertion	POL-TA-90022
Description	A testcase to verify that a component with noManagedTransaction and a service with propagatesTransaction, results in an error from the runtime.
Artifacts	POL_9016_TestCase.java Test_POL_9016.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite Service1.wsdl
Expected	Negative test:

output	"exception"
--------	-------------

## POL\_9017\_TestCase

Testcase ID	POL_9017_TestCase
Test Assertion	POL-TA-90025
Description	A testcase to verify that a reference with propagatesTransaction on a local tran component, results in an error from the runtime.
Artifacts	POL_9017_TestCase.java Test_POL_9017.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test: "exception"

## POL\_9018\_TestCase

Testcase ID	POL_9018_TestCase
Test Assertion	POL-TA-90026
Description	A testcase to verify that a reference with propagatesTransaction on a noManagedTransaction component, results in an error from the runtime.
Artifacts	POL_9018_TestCase.java Test_POL_9018.composite TestInvocation.wsdl TestClient_0004.composite TestComposite1.composite TestComposite4.composite Service1.wsdl
Expected output	Negative test:

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

## POL\_9019\_TestCase

Testcase ID	POL_9019_TestCase
Test Assertion	POL-TA-90031
Description	Tests that where a <component/> <implementation/> is marked with the intent managedTransaction.local that a reference of the component is not marked with the transactedOneWay intent
Artifacts	POL_9019_TestCase.java Test_POL_9019.composite TestInvocation.wsdl TestClient_0004.composite Service1.wsdl TestCompositeOneWayClient.composite TestCompositeOneWay.composite definitions.xml in POL_9019 contribution
Expected output	Negative test: "exception"

## POL\_9020\_TestCase

Testcase ID	POL_9020_TestCase
Test Assertion	POL-TA-90032
Description	Tests that where the <interface/> of a reference involves request/response operations, that the <reference/> using that interface is not marked with the transactedOneWay intent
Artifacts	POL_9020_TestCase.java Test_POL_9020.composite TestInvocation.wsdl TestClient_0004.composite Service1.wsdl TestComposite1.composite



	TestComposite4.composite definitions.xml in POL_9020 contribution
Expected output	Negative test: "exception"

## POL\_9021\_TestCase

Testcase ID	POL_9021_TestCase
Test Assertion	POL-TA-90033
Description	Tests that where the <interface/> of a reference involves request/response operations, that the <reference/> using that interface is not marked with the immediateOneWay intent
Artifacts	POL_9021_TestCase.java Test_POL_9021.composite TestInvocation.wsdl TestClient_0004.composite Service1.wsdl TestComposite1.composite TestComposite4.composite definitions.xml in POL_9021 contribution
Expected output	Negative test: "exception"

## POL\_9022\_TestCase

Testcase ID	POL_9022_TestCase
Test Assertion	POL-TA-90034
Description	Tests that where a <component/> <reference/> is marked with the asyncInvocation intent it is not also marked with the propagatesTransaction intent
Artifacts	POL_9022_TestCase.java Test_POL_9022.composite TestInvocation.wsdl

	TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## POL\_9023\_TestCase

Testcase ID	POL_9023_TestCase
Test Assertion	POL-TA-90035
Description	Tests that where a <component/> <service/> is marked with the asyncInvocation intent it is not also marked with the propagatesTransaction intent
Artifacts	POL_9023_TestCase.java Test_POL_9023.composite TestInvocation.wsdl TestClient_0004.composite definitions.xml
Expected output	Negative test: "exception"

## 2.5 Section 10

### POL\_10001\_TestCase

Testcase ID	POL_10001_TestCase
Test Assertion	POL-TA-100005
Description	A testcase to verify that the noListener intent is not used on a service.
Artifacts	POL_10001_TestCase.java Test_POL_10001.composite TestInvocation.wsdl TestClient_0004.composite
Expected output	Negative test: "exception"

## 2.6 Section 11

### POL\_11001\_TestCase

Testcase ID	POL_11001_TestCase
Test Assertion	POL-TA-110001
Description	A testcase to verify that the runtime rejects a composite that does not conform to the Policy FW schema.
Artifacts	POL_11001_TestCase.java Test_POL_11001.composite TestInvocation.wsdl TestClient_0004.composite
Expected output	Negative test: "exception"

### 3 Cross Mapping of Test Assertions to TestCases

Test Assertion	Test Cases
POL-TA-30001	POL_3001_TestCase
POL-TA-30002	Untestable - requires policySets written using a specific policy language - none is mandatory
POL-TA-30003	POL_3003_TestCase
POL-TA-30004	POL_3004_TestCase
POL-TA-30005	POL_3005_TestCase
POL-TA-30006	POL_3006_TestCase
POL-TA-30007	POL_3007_TestCase
POL-TA-30008	POL_3008_TestCase
POL-TA-30010	POL_3009_TestCase
POL-TA-30011	Untestable - WS-Policy not required to be supported
POL-TA-30013	POL_3011_TestCase
POL-TA-30015	POL_3007_TestCase POL_3012_TestCase
POL-TA-30016	POL_3013_TestCase
POL-TA-30017	POL_3011_TestCase POL_3014_TestCase
POL-TA-30018	POL_3011_TestCase POL_3015_TestCase
POL-TA-30019	POL_3016_TestCase
POL-TA-30020	POL_3017_TestCase POL_4002_TestCase
POL-TA-30021	POL_3018_TestCase POL_4002_TestCase
POL-TA-30025	POL_3019_TestCase
POL-TA-30026	POL_3020_TestCase

Test Assertion	Test Cases
POL-TA-40001	POL_4001_TestCase
POL-TA-40002	POL_4002_TestCase

POL-TA-40003	POL_4003_TestCase
POL-TA-40004	POL_4004_TestCase
POL-TA-40005	POL_4005_TestCase
POL-TA-40006	POL_4006_TestCase
POL-TA-40007	POL_4007_TestCase
POL-TA-40008	POL_4008_TestCase
POL-TA-40009	POL_4009_TestCase
POL-TA-40010	Untestable - WS-Policy not required to be supported
POL-TA-40011	Untestable - WS-Policy not required to be supported
POL-TA-40012	POL_4012_TestCase
POL-TA-40013	POL_4015_TestCase
POL-TA-40014	POL_4013_TestCase
POL-TA-40015	POL_4010_TestCase
POL-TA-40016	POL_4016_TestCase
POL-TA-40017	POL_3007_TestCase
POL-TA-40018	POL_4018_TestCase
POL-TA-40020	POL_4029_TestCase
POL-TA-40021	POL_4030_TestCase
POL-TA-40022	POL_4031_TestCase
POL-TA-40023	POL_4032_TestCase
POL-TA-40024	POL_4019_TestCase
POL-TA-40025	POL_4020_TestCase
POL-TA-40026	POL_4012_TestCase
POL-TA-40027	POL_4024_TestCase – delete? TA is preferred
POL-TA-40028	POL_4024_TestCase – delete? TA is preferred
POL-TA-40036	POL_3007_TestCase – delete? TA is preferred
POL-TA-40037	POL_4031_TestCase
POL-TA-40038	POL_4032_TestCase
POL-TA-40039	POL_4027_TestCase
POL-TA-40040	Untestable – no required binding available that meets the prerequisites of the test assertion.
POL-TA-40041	Untestable - requires policySets written using a specific policy language - none is mandatory.
POL-TA-40042	Untestable - requires policySets written using 2 specific policy language - none is mandatory.

POL-TA-40043	Untestable - requires support for WS-Policy which is not mandatory
POL-TA-40049	POL_4028_TestCase
POL-TA-40050	POL_4031_TestCase POL_4032_TestCase
POL-TA-40051	Untestable - requires a redeployment API
POL-TA-40052	Untestable - requires a redeployment API
POL-TA-40053	Untestable - requires a redeployment API
POL-TA-40054	Untestable - requires a redeployment API
POL-TA-40055	Untestable - requires a redeployment API
POL-TA-40056	Untestable - requires a redeployment API
POL-TA-40057	Untestable - requires a measurable required concrete policy
POL-TA-40058	POL_4033_TestCase
POL-TA-40059	Untestable - requires a redeployment API
POL-TA-40060	Untestable - requires a redeployment API
POL-TA-40061	Untestable - requires a redeployment API

Test Assertion	Test Cases
POL-TA-50001	POL_5001_TestCase

Test Assertion	Test Cases
POL-TA-70001	Untestable
POL-TA-70002	Untestable
POL-TA-70003	Untestable
POL-TA-70004	Untestable
POL-TA-70005	Untestable
POL-TA-70006	Untestable
POL-TA-70007	Untestable

Test Assertion	Test Cases
POL-TA-80001	Untestable
POL-TA-80002	Untestable
POL-TA-80003	Untestable

POL-TA-80004	Untestable
--------------	------------

Test Assertion	Test Cases
POL-TA-90003	Untestable
POL-TA-90004	Untestable
POL-TA-90006	Untestable
POL-TA-90007	Untestable
POL-TA-90008	Untestable
POL-TA-90009	POL_9006_TestCase
POL-TA-90010	Untestable
POL-TA-90011	Untestable
POL-TA-90012	POL_9009_TestCase
POL-TA-90013	Untestable - No required binding available
POL-TA-90016	Untestable
POL-TA-90017	Untestable
POL-TA-90018	Untestable
POL-TA-90020	Untestable
POL-TA-90021	POL_9015_TestCase
POL-TA-90022	POL_9016_TestCase
POL-TA-90025	POL_9017_TestCase
POL-TA-90026	POL_9018_TestCase
POL-TA-90027	Untestable
POL-TA-90028	Untestable
POL-TA-90030	POL_9009_TestCase
POL-TA-90031	POL_9006_TestCase POL_9019_TestCase
POL-TA-90032	POL_9020_TestCase
POL-TA-90033	POL_9021_TestCase
POL-TA-90034	POL_9022_TestCase
POL-TA-90035	POL_9023_TestCase
POL-TA-90036	POL_9006_TestCase

Test Assertion	Test Cases
----------------	------------

POL-TA-100001	POL_3019_TestCase
POL-TA-100002	POL_4030_TestCase
POL-TA-100003	Untestable
POL-TA-100004	Untestable - JMS binding not a required binding
POL-TA-100005	POL_10001_TestCase
POL-TA-100006	Untestable
POL-TA-100007	Untestable - EJB binding not a required binding
POL-TA-100008	Untestable - EJB binding not a required binding

<b>Test Assertion</b>	<b>Test Cases</b>
POL-TA-110001	POL_11001_TestCase



## 4 Catalog of Test Artifacts

### 4.1 Composite Files - lower level

Name	Valid	Description
TestClient_0004.composite	Y	Contains a service, property and optional reference that receives control from the test client.
TestComposite1.composite	Y	Contains a component that implements Service1.
TestComposite4.composite	Y	Contains a component that implements Service1 and has a reference to another component.
TestCompositeSOAP.composite	Y	Contains the use of the SCA defined SOAP intent.

### 4.2 Java Interfaces – General\_Java contribution

Name	Description
Service1.java	Interface with one request/response operation called operation1

### 4.3 Java Implementation Classes - General\_Java

Name	Description
ASM_0002_Client.java	Basic class that receives operation call from test client, and has one optional reference
Service1Impl.java	Implements Service1 by returning the value of the serviceName property on the component
Service1Impl2.java	Implements Service1 by passing the invocation on to it's reference and then returning the value of the serviceName property on the component

### 4.4 WSDL Interface Files - General

Name	Description
------	-------------

Service1.wsdl	PortType with one request/response operation called operation1
TestInvocation.wsdl	PortType used to invoke a test application

## 4.5 Policy files - General

Name	Description
General - definitions.xml	Contains a base set of policySets and intents that are used to test syntactic conformance statements

NOTE: Some testcase contributions also have a definitions.xml file. The existence of such a file usually indicates that it contains a statically checkable error.

## 5 Testcases Grouped by Optional Capability

These three testcase groups can be found in the Test\_Client contribution within the SCA Policy Testcases.

Testcase Group	Testcases
AllDirectAttachTests.java	POL_3001_TestCase, POL_3003_TestCase, POL_3004_TestCase, POL_3005_TestCase, POL_3006_TestCase, POL_3007_TestCase, POL_3008_TestCase, POL_3009_TestCase, POL_3011_TestCase, POL_3012_TestCase, POL_3013_TestCase, POL_3014_TestCase, POL_3015_TestCase, POL_3017_TestCase, POL_3018_TestCase, POL_3019_TestCase, POL_3020_TestCase, POL_4001_TestCase, POL_4002_TestCase, POL_4003_TestCase, POL_4004_TestCase, POL_4005_TestCase, POL_4006_TestCase, POL_4007_TestCase, POL_4008_TestCase, POL_4009_TestCase, POL_4012_TestCase, POL_4016_TestCase, POL_4018_TestCase, POL_4019_TestCase, POL_4020_TestCase, POL_4024_TestCase, POL_4027_TestCase, POL_4029_TestCase, POL_4030_TestCase, POL_4031_TestCase, POL_4032_TestCase, POL_4033_TestCase, POL_5001_TestCase, POL_9006_TestCase, POL_9009_TestCase, POL_9015_TestCase, POL_9016_TestCase, POL_9017_TestCase, POL_9018_TestCase, POL_9019_TestCase, POL_9020_TestCase, POL_9021_TestCase, POL_9022_TestCase, POL_9023_TestCase, POL_10001_TestCase, POL_11001_TestCase
AllExtAttachTests.java	POL_3001_TestCase, POL_3003_TestCase, POL_3004_TestCase, POL_3005_TestCase, POL_3006_TestCase, POL_3007_TestCase, POL_3008_TestCase, POL_3009_TestCase, POL_3011_TestCase, POL_3012_TestCase, POL_3013_TestCase, POL_3014_TestCase, POL_3015_TestCase, POL_3016_TestCase, POL_3017_TestCase, POL_3018_TestCase, POL_3019_TestCase, POL_3020_TestCase, POL_4001_TestCase, POL_4002_TestCase, POL_4003_TestCase, POL_4004_TestCase, POL_4005_TestCase, POL_4006_TestCase, POL_4007_TestCase, POL_4008_TestCase, POL_4009_TestCase, POL_4010_TestCase, POL_4012_TestCase, POL_4013_TestCase, POL_4015_TestCase, POL_4019_TestCase, POL_4020_TestCase, POL_4024_TestCase, POL_4027_TestCase, POL_4028_TestCase, POL_4029_TestCase, POL_4030_TestCase, POL_4031_TestCase, POL_4032_TestCase, POL_4033_TestCase, POL_5001_TestCase, POL_9006_TestCase, POL_9009_TestCase, POL_9015_TestCase, POL_9016_TestCase, POL_9017_TestCase, POL_9018_TestCase, POL_9019_TestCase, POL_9020_TestCase, POL_9021_TestCase, POL_9022_TestCase, POL_9023_TestCase, POL_10001_TestCase, POL_11001_TestCase
AllTests.java	POL_3001_TestCase, POL_3003_TestCase, POL_3004_TestCase, POL_3005_TestCase, POL_3006_TestCase, POL_3007_TestCase, POL_3008_TestCase, POL_3009_TestCase, POL_3011_TestCase, POL_3012_TestCase, POL_3013_TestCase, POL_3014_TestCase, POL_3015_TestCase, POL_3016_TestCase, POL_3017_TestCase, POL_3018_TestCase, POL_3019_TestCase, POL_3020_TestCase, POL_4001_TestCase, POL_4002_TestCase, POL_4003_TestCase,

POL\_4004\_TestCase, POL\_4005\_TestCase, POL\_4006\_TestCase,  
POL\_4007\_TestCase, POL\_4008\_TestCase, POL\_4009\_TestCase,  
POL\_4010\_TestCase, POL\_4012\_TestCase, POL\_4013\_TestCase,  
POL\_4018\_TestCase, POL\_4019\_TestCase, POL\_4020\_TestCase,  
POL\_4024\_TestCase, POL\_4027\_TestCase, POL\_4028\_TestCase,  
POL\_4029\_TestCase, POL\_4030\_TestCase, POL\_4031\_TestCase,  
POL\_4032\_TestCase, POL\_4033\_TestCase, POL\_5001\_TestCase,  
POL\_9006\_TestCase, POL\_9009\_TestCase, POL\_9015\_TestCase,  
POL\_9016\_TestCase, POL\_9017\_TestCase, POL\_9018\_TestCase,  
POL\_9019\_TestCase, POL\_9020\_TestCase, POL\_9021\_TestCase,  
POL\_9022\_TestCase, POL\_9023\_TestCase, POL\_10001\_TestCase,  
POL\_11001\_TestCase

---

## 6 Conformance

There are no conformance statements relating to the TestCases.

---

## Appendix A: Acknowledgments

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

### Participants:

Participant Name	Affiliation
Jeff Anderson	Deloitte Consulting LLP
Bryan Aupperle	IBM
Ron Barack	SAP AG*
Michael Beisiegel	IBM
Vladislav Bezrukov	SAP AG*
Henning Blohm	SAP AG*
David Booz	IBM
Fred Carter	AmberPoint
Tai-Hsing Cha	TIBCO Software Inc.
Martin Chapman	Oracle Corporation
Mike Edwards	IBM
Raymond Feng	IBM
Billy Feng	Primeton Technologies, Inc.
Robert Freund	Hitachi, Ltd.
Murty Gurajada	TIBCO Software Inc.
Simon Holdsworth	IBM
Michael Kanaley	TIBCO Software Inc.
Anish Karmarkar	Oracle Corporation
Nickolaos Kavantzias	Oracle Corporation
Rainer Kerth	SAP AG*
Pundalik Kudapkar	TIBCO Software Inc.
Meeraj Kunnumpurath	Individual
Rich Levinson	Oracle Corporation
Mark Little	Red Hat
Ashok Malhotra	Oracle Corporation
Jim Marino	Individual
Jeff Mischkinsky	Oracle Corporation
Dale Moberg	Axway Software*
Simon Nash	Individual
Bob Natale	Mitre Corporation*
Eisaku Nishiyama	Hitachi, Ltd.
Sanjay Patil	SAP AG*
Plamen Pavlov	SAP AG*
Martin Raepple	SAP AG*
Fabian Ritzmann	Sun Microsystems
Ian Robinson	IBM
Scott Vorthmann	TIBCO Software Inc.
Eric Wells	Hitachi, Ltd.

Prasad Yendluri  
Alexander Zubev

Software AG, Inc.\*  
SAP AG\*

---

## Appendix B: Revision History

Revision	Date	Editor	Changes Made
1	10/14/09	David Booz	Initial version sections 3 and 4
2	11/11/09	David Booz	Completed first pass of all test cases
3	05/11/10	Mike Edwards	Sync up with TestAssertions CD01
4	05/28/10	David Booz	Misc editorial updates
5	06/04/10	David Booz	More updates based on review comments
6	06/14/10	David Booz	Misc updates from review with TC – applied Issue 120, 121.
CD01	06/21/10	David Booz	Editorial updates for CD01.
CD01 rev1	09/15/10	David Booz	Apply 123,124,125,126
CSD01 rev2	11/10/10	David Booz	Apply 128 and prep for next CSD