



---

# WS-Reliability/WS-Security Interoperability Test Specification

Working Draft 01, April 11 2005

**Location:**

[http://www.standards-oss.org/documents/ws-r\\_wss\\_interop.pdf](http://www.standards-oss.org/documents/ws-r_wss_interop.pdf)

**Editors:**

Hamid Ben Malek ([hmalek@us.fujitsu.com](mailto:hmalek@us.fujitsu.com))

Jacques Durand ([jdurand@us.fujitsu.com](mailto:jdurand@us.fujitsu.com))

**Contributors:**

**Abstract:**

This document describes the test cases as well as the tools (applications and APIs) used for interoperability tests about the composition of WS-Reliability and WSS profiles. This document does not prescribe a conformance test suite.

---

# 1 Introduction

This document provides a documentation about the various test cases involved, as well as the web service applications, its APIs, and its tools, and on how to deploy the service application and hook up a client application with the client API of the test suite. The test suite can be downloaded from <http://www.standards-oss.org>

Basic security features that translate into specific message processing fall into four main areas identified in "Handbook of Applied Cryptography" (by A. Menezes, P. van Oorschot, and S. Vanstone, CRC Press, 1996: [www.cacr.math.uwaterloo.ca/hac](http://www.cacr.math.uwaterloo.ca/hac)), from which other features can be derived:

- Authentication
- Data integrity
- Confidentiality
- Non-repudiation

The objectives of this test plan is to verify composability of the Reliability function (as defined in WS-Reliability) with the most popular ways the above security functions are implemented in an WS-Security compliant manner.

The test suite designed here will require that the implementations of WS-Security and WS-Reliability are composed in a particular way. This architecture should not need be modified from one test case to the other. Each candidate implementation (supporting both security and reliability) must be such that it can execute all test cases. These test cases are not symmetric: in order to demonstrate that end-points A and B have equivalent capability regarding composition of reliability and security, the same test suite must be executed twice, once driven from A, once driven from B.

## 1.1 Authentication

Authentication may apply to an entity (e.g. a person) or to data. In our context, it can be verified by:

- username / password (e.g. involving wsse:UsernameToken)
- digital (XML) signature, involving a private key on sender side (e.g. involving SAML:Assertion token, and/or x509 token)

Composability of these authentication use cases with reliability should be verified. We distinguish two scopes: payload (SOAP body) and entire message (SOAP headers + body, including Reliability headers)

The test cases will restrict to the practice recommended by WS-Security and WS-I BSP 1.0.

## 1.2 Data Integrity

At minimum, this involves a signed digest of the data (e.g. HMAC). As the XML Signature (SignatureMethod) used will include computation of such a digest and its signing (e.g.

65 <http://www.w3.org/2000/09/xmlsig#hmac-sha1>), composability of this use case with reliability will require  
66 verifying composability of the signing method involved. We distinguish the same scopes as for  
67 authentication.

68 The test cases will restrict to the practice recommended by WS-Security and WS-I BSP 1.0.

69

### 70 **1.3 Confidentiality**

71

72 Involves encryption of the private sections, using XML Encryption. We distinguish the same scopes as for  
73 authentication.

74 The test cases will restrict to the practice recommended by WS-Security and WS-I BSP 1.0 (with the  
75 exception of enveloped signatures in the latter).

76

### 77 **1.4 Non-repudiation**

78

79 Two cases are usually considered: non-repudiation of origin and non-repudiation of receipt.

80 Non-repudiation of origin typically involves the same techniques as authentication of sent data above, and  
81 therefore no new test case will be provided.

82 Non-repudiation of receipt usually gives to the “receipt” some application-level semantics, such as schema  
83 validation, etc. In this case, a signed receipt message is sent back to the initial sender. The security  
84 pattern involved here is no different from the authentication use case above (besides the fact it applies to  
85 an application-level receipt). No test case need be provided here, that would verify composability aspects  
86 not verified by the authentication test cases.

87 There is another option for Non-repudiation of receipt: since in WS-Reliability the acknowledgement is  
88 sent “on delivery”, we could consider that in some cases a delivery semantics (from the reliability module)  
89 is sufficient for a receipt. In this case a signed RM Reply would be sufficient. However, since it is not  
90 required from WS-Reliability that the reception of RM Replies (signed or not) be notified to the application  
91 (Producer) layer, no test case will be proposed for this option.

92 In summary, the composability of non-repudiation use cases with reliability will either involve similar  
93 patterns as verified by previous test cases, or will involve a specific case (signed RM Replies) that would  
94 require a security-aware implementation of the Reliability module – which will not be required in this test  
95 suite.

96

---

## 97 2 Test Suite

98 This document provides a documentation about the various test cases involved, as well as the web  
99 service applications, its APIs, and its tools, and on how to deploy the service application and hook up a  
100 client application with the client API of the test suite.

### 101 2.1 Components

102 The components of the test suite, described by this specification, are the following:

103  
104 **A web service application:** this application is represented by a sample Purchase Order e-  
105 commerce application.

106  
107 **Database Server:** this is the database server used by the purchase order application to persists  
108 the request messages.

109  
110 **Audit Application:** this is a web-based application that provide a view of the contents of the  
111 database. Through this application, a user can find out whether a certain purchase order request  
112 has been received by the web service application or not.

113  
114 **Payload Files:** these are XML files representing various SOAP messages to be used by the test  
115 cases.

116  
117 **Configuration File:** this is an XML file that will reside on the client side, and prescribes WS-  
118 Reliability agreement to be used in each test case.

### 120 2.2 The web service application

121  
122 This is a sample purchase order application. The binaries of this application as well as a documentation  
123 on how to deploy it will be provided with this specification.

### 125 2.3 Audit Application

126  
127 This is a web-based application that will be deployed on the server side with the purchase order  
128 application. Its purpose is to provide a view to remote users of what messages have been received by the  
129 purchase order.

### 131 2.4 Database Server

132  
133 This will be a MySQL Database Server, used by the purchase order application to persists all the request  
134 messages it receives. The installation of this database server and its database will be provided.

135

## 136 **2.5 Payloads and Configuration files**

137

138 The payload will be represented by a set of XML files containing the SOAP messages that will be used by  
139 the test cases when calling the “Document-Based” purchase order application. These XML files are not  
140 used if the client applications are calling the “RPC-based” purchase order.

## 141 **2.6 TestSuite Client API**

142

143 Utility classes together with a documentation on their API will be provided with this specification. The  
144 purpose of these utility classes is to ease the integration of client applications with the test suite. These  
145 utility classes perform the following tasks:

146

147 Provide a proxy for the web service. The client application won't have to compile the WSDL file to  
148 generate a proxy. All what the client application will have to do is simply use the utility classes to send the  
149 SOAP messages.

150

151 Load the payload data: the utility classes can pre-load the payload data from the xml files, and provide the  
152 client application with a SOAP message ready to send. All what the client application will have to do is only  
153 append the WS-Reliability headers to the SOAP message and then send it to the endpoint web service.

154

155 Load the WS-Reliability Agreement: the utility classes can load a WS-Reliability agreement, represented  
156 in the form of an XML file. The client application can ask the utility classes for the different parameters to  
157 use, and accordingly will populate the SOAP message with the WS-Reliability headers and then send it. In  
158 other words, the client application won't have to be able to parse the WS-Reliability agreement file.

---

## 3 Test Cases

159

160

161 There will be a minimum of ten test cases for combining WS-Reliability and WSS. These test cases are  
162 not a substitute for WS-Reliability interoperability tests. It is assumed that the SOAP processors involved  
163 in this interoperability test have already passed the WS-Reliability interoperability test.

164

165 In all the ten test cases described below, from the point of view of reliability, all the SOAP message  
166 requests are to have guaranteed delivery with duplicate elimination and a "Callback" reply pattern. This  
167 combination is enough, because the goal of this test suite is not about WS-Reliability interoperability, but  
168 about the composition of security and reliability. Furthermore, the reliability interoperability tests are  
169 assumed to have been passed prior to this composition test suite.

170

171 All signatures and encryptions in this composition test suite, are accomplished using an X.509 certificate  
172 store, and the signatures are always detached.

173

### 3.1 Username Security Token Test (T1)

174

175  
176 This the first test case (Test #1) and it consists in sending a "login" SOAP message request with  
177 username/password in the WSS headers. The web service would response by sending a boolean value  
178 indicating whether the login request succeeded or not.

179

### 3.2 SAML Security Token Test (T2)

180

181  
182 This the second test case (Test #2) and it consists in sending a "login" SOAP message request with a  
183 SAML security token for authentication. The web service would response by sending a boolean value  
184 indicating whether the login request succeeded or not.

185

### 3.3 X509 Security Token Test (T3)

186

187  
188 This the third test case (Test # 3) and it consists in sending a "login" SOAP message request with an  
189 X.509 security token for authentication. The web service would response by sending a boolean value  
190 indicating whether the login request succeeded or not.

191

### 3.4 Sign Body Test (T4)

192

193  
194 This is the fourth test case (Test # 4) and it consists in sending a purchase order SOAP message request  
195 with the SOAP body being signed. The signature is a detached signature using an X.509 certificate store.

196

197

198

199 **3.5 Sign Body and Reliability headers Test (T5)**

200

201 This the fifth test case (Test # 5) and it consists in sending a purchase order SOAP message request with  
202 the SOAP body and reliability headers being signed. The signature would be a detached signature using  
203 X.509.

204

205 **3.6 Body Encryption Test (T6)**

206

207 This is test case # 6 and it consists in sending a purchase order SOAP message request with the SOAP  
208 body being encrypted with X.509.

209

210 **3.7 Encryption of body and reliability headers Test (T7)**

211

212 This is test case # 7 and it consists in sending a purchase order SOAP message request where both the  
213 SOAP body and reliability headers being encrypted with X.509.

214

215 **3.8 First Test and Fourth Test combined (T8)**

216

217 This is test case # 8 and it consists in sending a purchase order SOAP message request with the SOAP  
218 body being signed, and the WSS header containing a username security token. The signature is a  
219 detached one using X.509.

220

221 **3.9 Third Test and Seventh Test combined (T9)**

222

223 This is test case # 9 and it consists in sending a purchase order SOAP message request with the SOAP  
224 body and reliability headers being encrypted with X.509, and the WSS header containing an X.509 for  
225 authentication.

226

227 **3.10 First, Fifth, and Sixth Test combined (T10)**

228

229 This is test case # 10 and it consists in sending a purchase order SOAP message request with the SOAP  
230 body and reliability headers being signed, the SOAP body being encrypted, and the WSS header  
231 containing a security username token for authentication. The signature is detached, and both the  
232 signature and encryption are using X.509.

---

## A. Notices

234 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that  
235 might be claimed to pertain to the implementation or use of the technology described in this document or  
236 the extent to which any license under such rights might or might not be available; neither does it represent  
237 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to  
238 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made  
239 available for publication and any assurances of licenses to be made available, or the result of an attempt  
240 made to obtain a general license or permission for the use of such proprietary rights by implementors or  
241 users of this specification, can be obtained from the OASIS Executive Director.

242 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or  
243 other proprietary rights which may cover technology that may be required to implement this specification.  
244 Please address the information to the OASIS Executive Director.

245 **Copyright © OASIS Open 2004. All Rights Reserved.**

246 This document and translations of it may be copied and furnished to others, and derivative works that  
247 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
248 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
249 this paragraph are included on all such copies and derivative works. However, this document itself does  
250 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as  
251 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights  
252 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it  
253 into languages other than English.

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

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