



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Conformance Program Specification for the OASIS Security Assertion Markup Language (SAML)

Document identifier: draft-sstc-conform-spec-11

Location: <http://www.oasis-open.org/committees/security/docs>

Publication date: 19 February 2002

Maturity Level: Committee Working Draft

Send comments to: security-services-comment@lists.oasis-open.org unless you are subscribed to the security-services list for committee members -- send comments there if so. Note: Before sending messages to the security-services-comment list, you must first subscribe. To subscribe, send an email message to security-services-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Contributors:

- Marc Chanliau, Netegrity
- Robert Griffin, Entrust (editor)
- Hal Lockhart, Entegrity
- Eve Maler, Sun Microsystems
- Prateek Mishra, Netegrity
- Mike Myers
- Charles Norwood, SAIC
- Mark O'Neill, Vordel
- Tony Palmer, Vordel
- Darren Platt, RSA
- Irving Reid, Baltimore
- Lynne Rosenthal, NIST
- Krishna Sankar, Cisco Systems
- Mark Skall, NIST

Rev	What
001	Initial version
002	Strawman profiles, test cases and process
003	Revisions from 1-June-2001 review; added example of test case
004	Revisions from 18-June-2001 review; modified to reflect conformance clause
005	Additions to test cases
006	Additions to test cases; HTTP profile mandatory
007	Includes conformance clause; SOAP binding mandatory
007a	Draft using assertions rather partitions as basis of conformance

007b	Draft using bindings rather than partitions as basis of conformance
007c	Stylistic edits and added OASIS notices to 007a
08	Revised using bindings approach; corrected references; included issue
09	Removed SOAP Profile tests
10	Incorporated restriction for unbounded elements
11	Revised bounds for nested elements; mandatory/optional

31

32 **CONFORMANCE PROGRAM SPECIFICATION FOR THE OASIS SECURITY ASSERTION MARKUP**
33 **LANGUAGE (SAML) 1**

34 **1 INTRODUCTION 4**

35 1.1 SCOPE OF THE CONFORMANCE PROGRAM 4

36 1.2 NOTATION 4

37 **2 CONFORMANCE CLAUSE 5**

38 2.1 SPECIFICATION OF THE SAML STANDARD 5

39 2.2 DECLARATION OF SAML CONFORMANCE 5

40 2.3 MANDATORY/OPTIONAL ELEMENTS IN SAML CONFORMANCE 6

41 2.4 IMPACT OF EXTENSIONS ON SAML CONFORMANCE 7

42 2.5 MAXIMUM VALUES OF UNBOUNDED ELEMENTS 7

43 **3 CONFORMANCE PROCESS 9**

44 3.1 IMPLEMENTATION AND APPLICATION CONFORMANCE 9

45 3.2 PROCESS FOR DECLARING CONFORMANCE 10

46 **4 TECHNICAL REQUIREMENTS FOR SAML CONFORMANCE 11**

47 4.1 TEST GROUP 1 – SOAP OVER HTTP PROTOCOL BINDING 11

48 4.1.1 *Test Case 1-1: SOAP Protocol Binding: Valid Authentication Assertion Received in Valid Response to*
49 *Valid Authentication Query..... 11*

50 4.1.2 *Test Case 1-2: SOAP Protocol Binding: Valid Authentication Assertion Artifact Returned in Valid*
51 *Response to Valid Authentication Query..... 11*

52 4.1.3 *Test Case 1-3: SOAP Protocol Binding: Valid Authentication Assertion Returned in Valid Response to*
53 *Valid Authentication Query with artifact. 12*

54 4.1.4 *Test Case 1-4: SOAP Protocol Binding: Valid Authentication Assertion Query Received 12*

55 4.1.5 *Test Case 1-5: SOAP Protocol Binding: Valid Attribute Assertion Received in Valid Response to Valid*
56 *Attribute Query..... 12*

57 4.1.6 *Test Case 1-6: SOAP Protocol Binding: Valid Attribute Assertion Artifact Returned in Valid Response*
58 *to Valid Attribute Query. 13*

59 4.1.7 *Test Case 1-7: SOAP Protocol Binding: Valid Attribute Assertion Returned in Valid Response to*
60 *Valid Attribute Query..... 13*

61 4.1.8 *Test Case 1-8: SOAP Protocol Binding: Valid Attribute Query Received..... 13*

62	4.1.9	<i>Test Case 1-9: SOAP Protocol Binding: Valid Authorization Decision Assertion Received in Valid Response to Valid Authorization Decision Query.</i>	14
63			
64	4.1.10	<i>Test Case 1-10: SOAP Protocol Binding: Valid Authorization Decision Assertion Artifact Returned in Valid Response to Valid Authorization Decision Query.</i>	14
65			
66	4.1.11	<i>Test Case 1-11: SOAP Protocol Binding: Valid Authorization Decision Assertion Returned in Valid Response to Valid Query.</i>	15
67			
68	4.1.12	<i>Test Case 1-12: SOAP Protocol Binding: Valid Authorization Decision Assertion Query Received</i>	
69		15	
70	4.2	TEST GROUP 2 – WEB BROWSER PROFILES	15
71	4.2.1	<i>Test Case 2-1: HTTP Web Browser/Artifact Profile: Valid Authentication Assertion Artifact Produced in Response to Valid Authentication Query.</i>	15
72			
73	4.2.2	<i>Test Case 2-2: HTTP Web Browser/Artifact Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query with Artifact.</i>	16
74			
75	4.2.3	<i>Test Case 2-3: Web Browser/Post Profile: Valid Authentication Assertion Produced in Response to Valid Authentication Query.</i>	16
76			
77	5	TEST SUITE	17
78	6	CONFORMANCE SERVICES	18
79	7	REFERENCES	19
80		APPENDIX A. NOTICES	20
81		APPENDIX B. ISSUES	21
82		ISSUE: SHOULD ANY OF THE BINDINGS OR PROFILES BE MANDATORY FOR ALL IMPLEMENTATIONS OR APPLICATIONS CLAIMING CONFORMANCE TO THE SAML STANDARD?	21
83			
84		ISSUE: SHOULD THE SOAP BINDING BE MANDATORY?	21
85		ISSUE: IF THE SOAP BINDING IS MANDATORY, IS IT ALLOWABLE TO IMPLEMENT A SUBSET OF THE ASSERTIONS FOR THAT BINDING?	21
86			
87			

88 **1 Introduction**

89 This document describes the program and technical requirements for the SAML conformance system.

90 **1.1 Scope of the Conformance Program**

91 SAML deals with a rich set of functionalities ranging from authentication assertions to assertions for policy
92 enforcement. Not all software might choose to implement all the SAML specifications. In order to achieve
93 compatibility and interoperability, applications and software need to be certified for conformance in a
94 uniform manner. The SAML conformance effort aims at fulfilling this need.

95 The deliverables of the SAML conformance effort include:

- 96 ▪ Conformance Clause, defining at a high-level what conformance means for the SAML standard
- 97 ▪ Conformance Program specification, defining how an implementation or application establishes
98 conformance
- 99 ▪ Conformance Test Suite. This is a set of test programs, result files and report generation tools that
100 can be used by vendors of SAML-compliant software, buyers interested in confirming SAML
101 compliance of software, and testing labs running conformance tests on behalf of vendors or
102 buyers.

103 Section 2 of this document provides the SAML Conformance Clause. Section 3 deals with defining and
104 specifying the process by which conformance to the SAML specification can be demonstrated and certified.
105 Section 4 elaborates the technical requirements which constitute conformance; this includes both the levels
106 of conformance that may be demonstrated and the requirements for each of those levels of conformance.
107 Section 5 describes the test suite for SAML, including the processes for using the test suite to establish
108 conformance, and the policies and procedures relating to those processes. Section 6 defines the services
109 which are available to assist in establishing conformance.

110 **1.2 Notation**

111 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
112 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be
113 interpreted as described in IETF RFC 2119 [NIST/ITL] "What is this thing
114 called conformance" [Rosenthal, Brady; NIST/ITL Bulletin, January 2001]
115 <http://www.itl.nist.gov/div897/ctg/conformance/bulletin-conformance.htm>.

116 [RFC2119].

117 **2 Conformance Clause**

118 The objectives of the SAML Conformance Clause are to:

- 119 1. Ensure a common understanding of conformance and what is required to claim conformance
- 120 2. Promote interoperability in the exchange of authentication and authorization information
- 121 3. Promote uniformity in the development of conformance tests

122 The SAML Conformance Clause specifies explicitly all the requirements that have to be satisfied to claim
123 conformance to the SAML standard.

124 **2.1 Specification of the SAML Standard**

125 The following four specifications, in addition to this SAML conformance program specification, comprise the
126 proposed Version 1.0 specification for the SAML standard:

- 127 • Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) [**SAMLCore**]
- 128 • Security Considerations for the OASIS Security Assertion Markup Language (SAML) [**SAMLSec**]
- 129 • Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) [**SAMLBind**]
- 130 • Glossary for the OASIS Security Assertion Markup Language (SAML) [**SAMLGloss**]

131 Although additional documents might use or reference the SAML standard (such as white papers,
132 descriptions of custom profiles, and position papers referencing particular issues), they do not constitute
133 part of the standard.

134 **2.2 Declaration of SAML Conformance**

135 Conformance to the SAML standard may be declared for the entire standard or for a subset of the
136 standard, based on the requirements that a given implementation or application claims to meet. That is,
137 requirements can be applied at varying levels, so that a given implementation or application of the SAML
138 standard can achieve clearly defined conformance with all or part of the entire set of specifications.

139 SAML conformance must be expressed in terms of which SAML bindings and profiles are supported by a
140 given application or implementation. The application or implementation claiming conformance to the SAML
141 standard must support the SOAP protocol binding for at least one assertion. An application or
142 implementation may also support the web browser profiles.

143 For any binding for which an application or implementation claims conformance, the level of conformance
144 must then be specified in each of these dimensions:

- 145 • Whether the application or implementation acts as requester or responder or both requester and
146 responder of the SAML messages in the supported bindings and profiles.
- 147 • Which assertions the application or implementation supports for each supported binding.

148 Table 1 shows the protocols, protocol bindings, and profiles applicable to each SAML assertion. For each
149 SAML binding or profile to which an application or implementation claims conformance, the claim must
150 stipulate whether the requester and/or responder roles are supported and for which assertions for those
151 roles.

152 For example, an implementation consisting solely of an Authentication Authority responsible for generating
153 Authentication Assertions and returning those assertions in response to a SOAP-over-HTTP request would
154 correspond to the cell in the third column of the second row (including the column title row). If the
155 implementation also supported the use of that Authentication Authority Browser/Artifact profile, then the
156 third column in the fifth row would also be supported.

157

Table 1: Protocol Bindings and Profiles for SAML Assertions

Binding or Profile	Requester Role	Responder Role
SOAP over HTTP protocol binding	Send an Authentication Query to request an Authentication Assertion from a responder	Return an AuthenticationResponse containing an Authentication Assertion to the requester
	Send an AttributeQuery to request an Attribute Assertion from a responder	Return an AttributeResponse to containing an Attribute Assertion to the requester
	Send an AuthorizationDecisionQuery to request an Authorization Decision Assertion from a responder	Return an AuthorizationDecisionResponse containing an Authorization Decision Assertion to the requester
Browser/Artifact Profile	Request an Authentication Assertion corresponding to an artifact	Return the corresponding Authentication Assertion to the requester
Browser/POST Profile	Send an Authentication Assertion in a form POST	Process the received Authentication Assertion

158

159 An application or implementation should express its level of conformance in terminology such as the
160 following:

161 *[Application or implementation] as both requester and responder supports all SAML protocol bindings and*
162 *profiles, for all assertions. No optional elements for the assertions, bindings and profiles are supported.*

163 *[Application or implementation] as both requester and responder supports the SOAP protocol binding for all*
164 *assertions. It also supports the Conditions optional elements for all assertions in the SOAP protocol*
165 *binding. It does not support the Web Browser Profile and the SOAP profile for any assertion.*

166 *[Application or implementation] as both requester and responder supports the SOAP protocol binding for all*
167 *assertions, for all assertions.. It also support the Web Browser Profile for Authentication Assertion and all*
168 *required elements. No optional elements for the assertions, bindings and profiles are supported.*

169 An application or implementation that claims conformance for a particular binding or profile must support all
170 required elements of that binding or profile and of the assertions supported with that binding or profile. It
171 must also state which assertions are supported and which, if any optional elements for that binding or
172 profile and corresponding assertions are supported.

173 **2.3 Mandatory/Optional Elements in SAML Conformance**

174 The SOAP protocol binding must be implemented by all implementations or applications claiming SAML
175 conformance, for each assertion claimed as supported through a binding or profile. (see Appendix B:
176 Issues)

177 The SAML schema and binding specifications include both mandatory and optional elements. A conforming
178 application or implementation must be able to handle all valid SAML elements, including those that are
179 optional. However, it does not have to produce those optional elements.

180 For example:

- 181 ▪ An application or implementation that consumes assertions must be able to handle assertions that
 - 182 include the optional “condition” element, such as by rejecting any conditions that it does not
 - 183 recognize.
 - 184 ▪ An application or implementation that produces assertions can, but is not required to, include the
 - 185 optional “condition” element in those assertions.
 - 186 ▪ An application or implementation claiming support for an assertion must support the SOAP over
 - 187 HTTP protocol binding. It can also, optionally, implement the protocol by means of another binding.
- 188 The test cases for SAML conformance are intended to check for support of all valid SAML elements. They
- 189 also check whether an implementation or application accepts and properly handles optional assertion
- 190 elements (such as CONDITION) who value the implementation or application does not recognize. The test
- 191 suite does not check for handling of implementation- or application-specific values for optional elements.

192 **2.4 Impact of Extensions on SAML Conformance**

193 SAML supports extensions to assertions, protocols, protocol bindings and profiles. An application or

194 implementation may claim conformance to SAML only if its extensions (if any) meet the following

195 requirements:

- 196 • Extensions shall not re-define semantics for existing functions.
- 197 • Extensions shall not alter the specified behavior of interfaces defined in this standard.
- 198 • Extensions may add additional behaviors.
- 199 • Extensions shall not cause standard-conforming functions (i.e., functions that do not use the
- 200 extensions) to execute incorrectly.

201 SAML bindings and profiles can be extended so long as the above conditions are met. It is requested that,

202 if a system is extending the SAML assertions:

- 203 • The mechanism for determining application conformance and the extensions shall be clearly
- 204 described in the documentation, and the extensions shall be marked as such;
- 205 • Extensions shall follow the spirit, principles and guidelines of the SAML specification, that is, the
- 206 specifications must be extended in a standard manner as defined in the extension fields.
- 207 • In the case where an implementation has added additional behaviors, the implementation shall
- 208 provide a mechanism whereby a conforming application shall be recognized as such, and be
- 209 executed in an environment that supports the functional behavior defined in this standard

210 Extensions are outside the scope of conformance. There are no mechanisms specified to validate and

211 verify the extensions. This section contains the recommended guidelines for extensions.

212 **2.5 Maximum Values of Unbounded Elements**

213 The SAML schema supports a number of elements that can be specified multiple times in an assertion,

214 request or response. An application or implementation claiming conformance must support at least the

215 values listed in Table 2 below for each of the elements defined as “unbounded” in the SAML schema. In

216 those cases where the maximum value is greater than the listed values, the application or implementation

217 should state what that maximum supported value is.

218 However. Some of the elements in the table can be nested, such that repeated elements have a

219 multiplicative effect on the number of elements. For example, trees of nested unbounded elements include

220 the following:

- 221 Response > Assertion > Signature
- 222 Response > Assertion > Advice
- 223 Response > Assertion > Condition > Target

- 224 Response > Assertion > Condition > Audience
- 225 Response > Assertion > Statement > SubjectConfirmationMethod
- 226 Response > Assertion > Statement > AuthorityBinding
- 227 Response > Assertion > Statement > Action
- 228 Response > Assertion > Statement > Attribute > AttributeValue
- 229 In a response containing 10 assertions, each with 10 AttributeStatements, each with 10 Attributes, each
- 230 with 10 AttributeValues, this tree alone comprises 10,000 elements.
- 231 Therefore, In order to minimize the potential impact of nested unbounded elements, an application or
- 232 implementation can limit the total number of elements supported in a given request, response or (when
- 233 this is used in the POST profile) assertion to no more than 1000 total elements and still claim conformance
- 234 to the SAML V1.0 specification.

235

Table 2: Unbounded Elements

Element	Parent Element	Maximum Value	Section in sstc-core
Statement	Assertion	1000	2.3.3
Signature	Assertion	1000	2.3.3
Condition	Assertion	1000	2.3.3
Audience	Condition	1000	2.3.3.1.3
Target	Condition	1000	2.3.3.1.4
Advice	Assertion	1000	2.3.3.2
ConfirmationMethod	SubjectConfirmation	1000	2.4.2.3
AuthorityBinding	AuthenticationStatement	1000	2.4.3.2
Evidence	AuthorizationDecisionStatement	1000	2.4.4
Actions	Action	1000	2.4.4.1
Attribute	AttributeStatement	1000	2.4.5
AttributeValue	Attribute	1000	2.4.5.1.1
RespondWith	Request	1000	3.2.1
AssertionArtifact	Request	1000	3.2.2
AttributeDesignator	AttributeQuery	1000	3.3.4
Evidence	AuthorizationDecisionQuery	1000	3.3.5
Assertion	Response	1000	3.4.2
StatusMessage	Status	1000	3.4.3
StatusDetail	Status	1000	3.4.3

236

237

3 Conformance Process

238
239

As discussed in the article “What is this thing called conformance” [NIST/ITL], conformance can comprise any of several levels of formal process:

240
241
242
243
244
245
246

- **Conformance testing** (also called conformity assessment) is the execution of automated or non-automated scripts, processes or other mechanisms to determine whether an application or implementation of a specification deviates from that specification. For SAML, conformance testing means the running of (some or all) tests within the SAML Conformance Test Suite. Conformance testing performed by implementers early on in the development process can find and correct their errors before the software reaches the marketplace, without necessarily being part of either a validation or certification process.

247
248
249

- **Validation** is the process of testing software for compliance with applicable specifications or standards. The validation process consists of the steps necessary to perform the conformance testing by using an official test suite in a prescribed manner.

250
251
252
253
254

- **Certification** is the acknowledgment that a validation has been completed and the criteria established by the certifying organization for issuing a certificate have been met. Successful completion of certification results in the issuance of a certificate (or brand) indicating that the implementation conforms to the appropriate specification. It is important to note that certification cannot exist without validation, but validation can exist without certification.

255
256
257
258
259
260

The conformance process for SAML is based on validation rather than certification. That is, no certifying organization has been established with the responsibility for issuing a statement of conformance with regard to an application or implementation. Therefore, an implementer who has validated SAML conformance by means of conformance testing may not legitimately use the term “certified for SAML conformance”. Until and if a certification process is in place, vendor declaration of validation will be the only means of asserting that conformance testing has been performed.

261
262
263
264
265
266
267

The conformance process does not stipulate whether validation is performed by the implementer, by a third-party, or by the customer of an application or implementation. Rather, the conformance process describes the way in which conformance testing should be done in order to demonstrate that an application or implementation correctly performs the functionality specified in the standard. Validation achieved through the SAML conformance process provides software developers and users assurance and confidence that the product behaves as expected, performs functions in a known manner, and possesses the prescribed interface or format.

268
269

The SAML Technical Committee is responsible for generating the materials that allow vendors, customers, and third parties to evaluate software for SAML conformance. These materials include:

270
271
272
273

- Documentation describing test cases, linked to use cases and requirements
- Test suite, based on those test cases, that can be run against an implementation to demonstrate any of the several levels/profiles of conformance defined in the conformance clause of the SAML specification

274
275

- Documentation describing how to run the test suite, interpret the results, and resolve disputes regarding the results of the tests

276

The SAML Technical Committee is not, however, responsible for testing of particular implementations.

277

3.1 Implementation and Application Conformance

278

SAML Conformance is applicable to:

279
280
281

- Implementations of SAML assertions, protocols and bindings. These could be in the form of toolkits, products incorporating SAML components, or reference implementations that demonstrate the use of SAML components.

- 282 • Applications that produce or consume SAML protocol bindings or that execute on SAML
283 implementations (for example, using a SAML toolkit to support multi-domain single-signon)

284 A conforming **implementation** shall meet all the following criteria:

- 285 4. The implementation shall support all the required interfaces defined within this standard for a given
286 binding or profile. It shall also specify which assertions relevant to that binding or profile are supported.
287 The implementation shall support the functional behavior described in the standard.
- 288 5. An implementation may provide additional or enhanced features or functionality not required by the
289 SAML Specification. These non-standard extensions shall not alter the specified behavior of interfaces
290 or functionality defined in the specification.
- 291 6. The implementation may provide additional or enhanced facilities not required by this standard. These
292 non-standard extensions shall not alter the specified behavior of interfaces defined in this standard.
293 They may add additional behaviors. In these circumstances, the implementation shall provide a
294 mechanism whereby a SAML conforming application shall be recognized as such, and be executed in
295 an environment that supports the functional behavior defined in this standard.

296 A conforming **application** shall meet all the following criteria:

- 297 1. The application shall be able to execute on any conforming implementation.
- 298 2. If an application requires a particular feature set that is not available on a specific implementation, then
299 the application must act within the bounds of the SAML specification even though that means that the
300 application may not perform any useful function. Specifically, the application shall do no harm, and
301 shall correctly return resources and vacate memory upon discovery that a required element is not
302 present.

303 **3.2 Process for Declaring Conformance**

304 The following process should be followed in declaring that an application or implementation conforms to the
305 SAML standard:

- 306 1. Determine which bindings and protocols will be asserted as conforming.
- 307 2. Obtain the test suite for the SAML standard from [tbs]
- 308 3. Validate the application or implementation by execute those conformance tests from the test suite
309 which are relevant to the conformance being asserted.
- 310 4. Send the statement claiming conformance to the Security Services Technical Committee at [tbs] so
311 that it can be posted on the SAML web site. A statement of any bindings and profiles which are being
312 used that are not part of the SAML standard should also be sent to the Security Services Technical
313 Committee at the same time for posting on the SAML web site.

314 4 Technical Requirements for SAML 315 Conformance

316 This section defines the technical criteria which apply to declaring conformance to the SAML standard. The
317 requirements are specified as test cases.

318 Each test case includes:

- 319 ▪ A description of the test purpose (that is, what is being tested – the conditions, requirements, or
320 capabilities which are to be addressed by a particular test)
- 321 ▪ The pass/fail criteria
- 322 ▪ A reference to the requirement in the requirements document [**SAMLReqs**] relevant to the test case
- 323 ▪ A reference to the section in the standard from which the test case is derived (that is, traceability back
324 to the specification)

325 For each assertion, both required tests for producing and consuming the assertion, as well as tests related
326 to protocols, bindings and profiles are specified.

327 4.1 Test Group 1 – SOAP over HTTP Protocol Binding

328 The test cases in this test group check for conformance to SOAP Protocol Binding for the SAML standard.
329 Any implementation or application claiming conformance to SAML must be able to execute these test
330 cases successfully, even if that support is incidental to the primary purposes of the application or
331 implementation.

332 4.1.1 Test Case 1-1: SOAP Protocol Binding: Valid Authentication Assertion 333 Received in Valid Response to Valid Authentication Query.

334 *Description:* This test case requests and receives an authentication assertion created by an
335 implementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms
336 that the authentication assertion returned by the implementation-under-test is valid for all required
337 functionality.

338 *Pass/Fail Criteria:* Authentication assertion contains all required elements in the right format and sequence,
339 AuthenticationQuery is accepted by implementation-under-test, and AuthenticationResponse contains all
340 required elements in correct sequence.

341 *Requirements Reference:* **R-AUTHN**, and **R-MULTIDOMAIN**

342 *Specification Reference:* *SAML Core, sections 2.4.3 and 3*

343 *SAML Bind, section 3.1.*

344 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP binding of the
345 AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It
346 establishes successful execution of the test case by inspection of the format of the returned assertion.

347 4.1.2 Test Case 1-2: SOAP Protocol Binding: Valid Authentication Assertion 348 Artifact Returned in Valid Response to Valid Authentication Query.

349 *Description:* This test case submits a SOAP message containing authentication credentials to an
350 implementation-under-test, requesting an authentication artifact. It checks that the implementation-under-
351 test returns a valid authentication assertion artifact in a valid AuthenticationResponse. It then submit the
352 artifact to the application/implementation-under-test. Finally, it checks that the returned authentication
353 assertion is valid.

354 *Pass/Fail Criteria:* Authentication assertion artifact returned by implementation-under-test must be contain
355 all required information in the right sequence and format. Any optional information included (including
356 conditions) must not compromise the validity of the required information.

357 *Reference:* **R-AUTHN**, and **R-MULTIDOMAIN**

358 *Specification Reference:* *SAML Core, sections 2.4.3 and 3*

359 *SAML Bind, section 3.1.*

360 *Implementation notes:* Test program implementing this test case establishes successful execution of the
361 test case by inspection of the format of the returned assertion artifact.

362 **4.1.3 Test Case 1-3: SOAP Protocol Binding: Valid Authentication Assertion** 363 **Returned in Valid Response to Valid Authentication Query with artifact.**

364 *Description:* This test case requests and receives an authentication assertion artifact created by an
365 implementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms
366 that the returned authentication assertion is valid for all required functionality.

367 *Pass/Fail Criteria:* Authentication assertion contains all required elements in the right format and sequence,
368 AuthenticationQuery is accepted by implementation-under-test, and AuthenticationResponse contains all
369 required elements in correct sequence.

370 *Requirements Reference:* **R-AUTHN**, and **R-MULTIDOMAIN**

371 *Specification Reference:* *SAML Core, sections 2.4.3 and 3*

372 *SAML Bind, section 3.1*

373 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP binding of the
374 AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It
375 establishes successful execution of the test case by inspection of the format of the returned assertion.

376 **4.1.4 Test Case 1-4: SOAP Protocol Binding: Valid Authentication Assertion** 377 **Query Received**

378 *Description:* This test case receives an authentication assertion query created by an implementation-under-
379 test using the AuthenticationRequest protocol in the SOAP binding. It then confirms that the returned
380 authentication query is valid for all required functionality.

381 *Pass/Fail Criteria:* AuthenticationQuery contains all required elements in the right format and sequence.

382 *Requirements Reference:* **R-AUTHN**, and **R-MULTIDOMAIN**

383 *Specification Reference:* *SAML Core, sections 2.4.3 and 3*

384 *SAML Bind, section 3.1*

385 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP binding of the
386 AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It
387 establishes successful execution of the test case by inspection of the format of the returned assertion.

388 **4.1.5 Test Case 1-5: SOAP Protocol Binding: Valid Attribute Assertion Received in** 389 **Valid Response to Valid Attribute Query.**

390 *Description:* This test case requests and receives an attribute assertion created by an implementation-
391 under-test using the AttributeRequest protocol in the SOAP binding. It then confirms that the attribute
392 assertion returned by the implementation-under-test is valid for all required functionality.

393 *Pass/Fail Criteria:* Attribute assertion contains all required elements in the right format and sequence,
394 AttributeQuery is accepted by implementation-under-test, and AttributeResponse contains all required
395 elements in correct sequence.

396 *Requirements Reference:* **R-AUTHZ**, and **R-MULTIDOMAIN**

397 *Specification Reference:* *SAML Core, Sections 2.4.5 and 3*

398 *SAML Bind, section 3.1.*

399 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP bindings of the
400 AttributeQuery and AttributeResponse protocols to obtain the Attribute Assertion. It establishes successful
401 execution of the test case by inspection of the format of the returned assertion.

402 **4.1.6 Test Case 1-6: SOAP Protocol Binding: Valid Attribute Assertion Artifact** 403 **Returned in Valid Response to Valid Attribute Query.**

404 *Description:* This test case submits a SOAP message containing attribute credentials to an
405 implementation-under-test, requesting an attribute artifact. It checks that the implementation-under-test
406 returns a valid attribute assertion artifact in a valid AttributeResponse. It then submit the artifact to the
407 application/implementation-under-test. Finally, it checks that the returned attribute assertion is valid.

408 *Pass/Fail Criteria:* Attribute assertion artifact returned by implementation-under-test must be contain all
409 required information in the right sequence and format. Any optional information included (including
410 conditions) must not compromise the validity of the required information.

411 *Reference:* **R-AUTHZ**, and **R-MULTIDOMAIN**

412 *Specification Reference:* *SAML Core, Sections 2.4.5 and 3*

413 *SAML Bind, section 3.1.*

414 *Implementation notes:* Test program implementing this test case establishes successful execution of the
415 test case by inspection of the format of the returned assertion artifact.

416 **4.1.7 Test Case 1-7: SOAP Protocol Binding: Valid Attribute Assertion Returned** 417 **in Valid Response to Valid Attribute Query.**

418 *Description:* This test case requests and receives an attribute assertion created by an implementation-
419 under-test using the AttributeRequest protocol in the SOAP binding. It then confirms that the attribute
420 assertion is valid for all required functionality.

421 *Pass/Fail Criteria:* Attribute assertion contains all required elements in the right format and sequence,
422 AttributeQuery is accepted by implementation-under-test, and AttributeResponse contains all required
423 elements in correct sequence.

424 *Requirements Reference:* **R-AUTHZ**, and **R-MULTIDOMAIN**

425 *Specification Reference:* *SAML Core, Sections 2.4.5 and 3*

426 *SAML Bind, section 3.1*

427 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP binding of the
428 AttributeQuery and AttributeResponse protocols to obtain the Attribute Assertion. It establishes successful
429 execution of the test case by inspection of the format of the returned assertion.

430 **4.1.8 Test Case 1-8: SOAP Protocol Binding: Valid Attribute Query Received**

431 *Description:* This test case receives an attribute assertion query created by an implementation-under-test
432 using the AttributeRequest protocol in the SOAP binding. It then confirms that the returned authentication
433 query is valid for all required functionality.

434 *Pass/Fail Criteria:* AuthenticationQuery contains all required elements in the right format and sequence.
435 *Requirements Reference:* **R-AUTHZ**, and **R-MULTIDOMAIN**
436 *Specification Reference:* *SAML Core, sections 2.4.5 and 3*
437 *SAML Bind, section 3.1*
438 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP binding of the
439 AttributeQuery and Response protocols to obtain the Attribute Assertion. It establishes successful
440 execution of the test case by inspection of the format of the returned assertion.

441 **4.1.9 Test Case 1-9: SOAP Protocol Binding: Valid Authorization Decision**
442 **Assertion Received in Valid Response to Valid Authorization Decision**
443 **Query.**

444 *Description:* This test case requests and receives an authentication assertion created by an
445 implementation-under-test using the AuthenticationRequest protocol in the SOAP binding. It then confirms
446 that the authentication assertion returned by the implementation-under-test is valid for all required
447 functionality.

448 *Pass/Fail Criteria:* Authorization decision assertion contains all required elements in the right format and
449 sequence, AuthorizationQuery is accepted by implementation-under-test, and AuthorizationResponse
450 contains all required elements in correct sequence.

451 *Requirements Reference:* **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

452 *Specification Reference:* *SAML Core, Section 2.4.4 and 3*

453 *SAML Bind, section 3.1.*

454 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP bindings of the
455 AuthorizationQuery and AuthorizationResponse protocols to obtain the Authorization decision Assertion. It
456 establishes successful execution of the test case by inspection of the format of the returned assertion.

457 **4.1.10 Test Case 1-10: SOAP Protocol Binding: Valid Authorization Decision**
458 **Assertion Artifact Returned in Valid Response to Valid Authorization**
459 **Decision Query.**

460 *Description:* This test case submits a SOAP message containing an authorization decision request to an
461 implementation-under-test, requesting an authorization decision artifact. It checks that the implementation-
462 under-test returns a valid authorization decision assertion artifact in a valid AuthorizationResponse. It then
463 submit the artifact to the application/implementation-under-test. Finally, it checks that the returned
464 authorization decision assertion is valid.

465 *Pass/Fail Criteria:* Authorization decision assertion artifact returned by implementation-under-test must be
466 contain all required information in the right sequence and format. Any optional information included
467 (including conditions) must not compromise the validity of the required information.

468 *Reference:* **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

469 *Specification Reference:* *SAML Core, Sections 2.4.4 and 3*

470 *SAML Bind, section 3.1.*

471 *Implementation notes:* Test program implementing this test case establishes successful execution of the
472 test case by inspection of the format of the returned assertion artifact.

473 **4.1.11 Test Case 1-11: SOAP Protocol Binding: Valid Authorization Decision**
474 **Assertion Returned in Valid Response to Valid Query.**

475 *Description:* This test case requests and receives an authorization decision assertion created by an
476 implementation-under-test using the AuthorizationRequest protocol in the SOAP over HTTP binding. It then
477 confirms that the authorization decision assertion is valid for all required functionality.

478 *Pass/Fail Criteria:* Authorization decision assertion contains all required elements in the right format and
479 sequence, AuthorizationQuery is accepted by implementation-under-test, and AuthorizationResponse
480 contains all required elements in correct sequence.

481 *Requirements Reference:* **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

482 *Specification Reference:* *SAML Core, Sections 2.4.4 and 3*

483 *SAML Bind, section 3.1*

484 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP protocol
485 bindings of the AuthorizationQuery and AuthorizationResponse protocols to obtain the Authorization
486 decision Assertion. It establishes successful execution of the test case by inspection of the format of the
487 returned assertion.

488 **4.1.12 Test Case 1-12: SOAP Protocol Binding: Valid Authorization Decision**
489 **Assertion Query Received**

490 *Description:* This test case receives an authorization decision assertion query created by an
491 implementation-under-test using the AuthorizationRequest protocol in the SOAP binding. It then confirms
492 that the received query is valid for all required functionality.

493 *Pass/Fail Criteria:* AuthorizationQuery contains all required elements in the right format and sequence.

494 *Requirements Reference:* **R-AUTHZDECISION**, and **R-MULTIDOMAIN**

495 *Specification Reference:* *SAML Core, sections 2.4.4 and 3*

496 *SAML Bind, section 3.1*

497 *Implementation notes:* Test program implementing this test case uses the SOAP over HTTP binding of the
498 AuthenticationQuery and AuthenticationResponse protocols to obtain the Authentication Assertion. It
499 establishes successful execution of the test case by inspection of the format of the returned assertion.

500 **4.2 Test Group 2 – Web Browser Profiles**

501 The test cases in this test group check for conformance to the HTTP Web Browser Profiles for the SAML
502 standard. Both the Browser/Artifact and Browser/POST profiles are optional. Any implementation or
503 application claiming conformance to the Web Browser/Artifact Profile of SAML must be able to execute
504 Test Cases 3-1 and 3-2 successfully. Any implementation or application claiming conformance to the Web
505 Browser/Post Profile of SAML must be able to execute Test Cases 3-3 successfully.

506 **4.2.1 Test Case 2-1: HTTP Web Browser/Artifact Profile: Valid Authentication**
507 **Assertion Artifact Produced in Response to Valid Authentication Query.**

508 *Description:* This test case submits an HTTP message to an implementation-under-test containing
509 authentication credentials and checks that the implementation-under-test returns a valid authentication
510 assertion artifact. It submits the authentication artifact to the implementation-under-test and confirms that
511 the authentication assertion artifact has been properly consumed by inspecting the authentication assertion
512 returned.

513 *Pass/Fail Criteria:* Authentication assertion artifact returned by implementation-under-test must be contain
514 all required information in the right sequence and format. Any optional information included (including
515 conditions) must not compromise the validity of the required information.

516 *Reference: R-AUTHN, and R-MULTIDOMAIN*

517 *Specification Reference: SAML Core, Section 2.4.3;*

518 *SAML Bind, section 4.1.1*

519 *Implementation notes: Test program implementing this test case establishes successful execution of the*
520 *test case by inspection of the format of the returned assertion artifact.*

521 **4.2.2 Test Case 2-2: HTTP Web Browser/Artifact Profile: Valid Authentication**
522 **Assertion Produced in Response to Valid Authentication Query with**
523 **Artifact.**

524 *Description: This test case uses an artifact to request and receive an authentication assertion created by*
525 *an implementation-under-test. It then confirms that the authentication assertion is valid for all required*
526 *functionality.*

527 *Pass/Fail Criteria: Authorization decision assertion contains all required elements in the right format and*
528 *sequence, AuthorizationQuery is accepted by implementation-under-test, and AuthorizationResponse*
529 *contains all required elements in correct sequence.*

530 *Requirements Reference: R-AUTHN, and R-MULTIDOMAIN*

531 *Specification Reference: SAML Core, Section 2.4.3*

532 *SAML Bind, section 4.1.1*

533 *Implementation notes: Test program implementing this test case establishes successful execution of the*
534 *test case by inspection of the format of the returned assertion.*

535 **4.2.3 Test Case 2-3: Web Browser/Post Profile: Valid Authentication Assertion**
536 **Produced in Response to Valid Authentication Query.**

537 *Description: This test case submits an HTTP POST message to an implementation-under-test containing*
538 *authentication credentials and checks that the implementation-under-test returns a valid authentication*
539 *assertion (also called “SSO Assertion” in the bindings specification).*

540 *Pass/Fail Criteria: Authentication assertion returned by implementation-under-test must contain all required*
541 *information in the right sequence and format. Any optional information included (including conditions) must*
542 *not compromise the validity of the required information.*

543 *Reference: R-AUTHN, and R-MULTIDOMAIN*

544 *Specification Reference: SAML Core, Section 2.4.3;*

545 *SAML Bind, section 4.1.2*

546 *Implementation notes: Test program implementing this test case establishes successful execution of the*
547 *test case by inspection of the format of the returned assertion.*

548

5 Test Suite

549 A test suite, which is the combination of test cases and test documentation, is used to check whether an
550 implementation or application satisfies the requirements in the standard. The test cases, implemented by a
551 test tool or a set of files (i.e., data, programs, scripts, or instructions for manual action) checks each
552 requirement in the specification to determine whether the results produced by the implementation or
553 application match the expected results, as defined by the specification.

554 The test documentation describes how the testing is to be done and the directions for the tester to follow.
555 Additionally, the documentation should be detailed enough so that testing of a given implementation can be
556 repeated with no change in test results.

557 Conformance testing is black box testing to test the functionality of an implementation. This means that the
558 internal structure or the source code of a candidate implementation is not available to the tester. However,
559 content and format of received or returned messages can be inspected as part of the determination of
560 conformance.

561 The test suite for SAML should be platform independent, non-biased, objective tests. Generally a
562 conformance test suite is a collection of combinations of legal and illegal inputs to the implementation being
563 tested, together with a corresponding collection of expected results. Only the requirements specified in the
564 standard are testable. A test suite should not check any implementation properties that are not described
565 by the standard or set of standards. A test suite cannot require features that are optional in a standard, but
566 if such features are present, a test suite could include tests for those features. A test suite does not assess
567 the performance of an implementation unless performance requirements are specified in the specification,
568 although implementation dependencies or machine dependencies may be demonstrated through the
569 execution of the test cases.

570 The results of conformance testing apply only to the implementation and environment for which the tests
571 are run. Test suites may be provided as a web-based system executed on a remote server, downloadable
572 files for local execution, or a combination of remote and local access and execution. The method for
573 providing and delivering the test suite depends on what is being tested as well as the objective for test suite
574 use – that is, providing self-test capability or formal certification testing.

575 As a test suite for SAML becomes available, the following information will be provided:

- 576 ▪ Reference Architecture
- 577 ▪ Infrastructure
- 578 ▪ Using the test suite
- 579 ▪ Test result tabulation and reporting

580 The SAML test suite will be maintained on a best-effort basis.

581

6 Conformance Services

582
583
584
585
586

The OASIS Security Services Technical Committee does not itself provide conformance services. As the SAML test suite becomes available and experience with SAML identified appropriate conformance testing approaches, the Conformance Specification will describe the services which the organization should provide including software services, releases, self-test kit, actual computer systems, facilities, web based interfaces, and availability.

587

7 References

588

[NIST/ITL]

“What is this thing called conformance” [Rosenthal, Brady; NIST/ITL Bulletin, January 2001] <http://www.itl.nist.gov/div897/ctg/conformance/bulletin-conformance.htm>.

589

590

591

[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.

592

593

[SAMLBind]

P. Mishra et al., *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-11.pdf>, OASIS, December 2001.

594

595

596

[SAMLCore]

P. Hallam-Baker et al., *Assertions and Protocol for the OASIS Security Assertion Markup Language*, <http://www.oasis-open.org/committees/security/docs/draft-sstc-sec-core-27.pdf>, OASIS, January 2002.

597

598

599

[SAMLGloss]

J. Hodges et al., *Glossary for the OASIS Security Assertion Markup Language (SAML)*, <http://www.oasis-open.org/committees/security/docs/draft-sstc-glossary-02.pdf>, OASIS, December 2001.

600

601

602

[SAMLReqs]

D. Platt et al., *SAML Requirements and Use Cases*, OASIS, December 2001.

603

[SAMLSec]

C. McLaren et al., *Security Considerations for the OASIS Security Assertion Markup Language*, <http://www.oasis-open.org/committees/security/docs/draft-sstc-sec-consider-04.pdf>, OASIS, January 2002.

604

605

606 **Appendix A. Notices**

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

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

618 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS] 2001. All
619 Rights Reserved.

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

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

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

634 **Appendix B. Issues**

635 **Issue: Should any of the bindings or profiles be mandatory for** 636 **all implementations or applications claiming conformance to** 637 **the SAML standard?**

638 Because of the importance of interoperability among implementations or applications claiming conformance
639 to the SAML standard, one of the recommendations in this version of the SAML Conformance Specification
640 is to require all implementations or applications to implement the SOAP binding for any assertions it
641 supports (including in other profiles).. This ensures that 1) assertions created by the implementation or
642 application can be retrieved using the SOAP binding, either directly or by means of an artifact, and can be
643 inspected for validity; and 2) the ability of the implementation or application to consume assertions
644 generated by another SAML-compliant implementation or application can be verified.

645 Alternatively, no single binding or profile need be mandatory, as long as an implementation or application
646 claiming conformance is specific regarding which bindings and/or profiles it supports, with what assertions,
647 and for what roles (responder / requester). This is the approach taken in the Conformance Specification
648 prior to version 006.

649 **Issue: Should the SOAP binding be mandatory?**

650 The SOAP binding is suggested as mandatory because it provides the most fully-specified mechanism for
651 requesting and returning all three assertions.

652 **Issue: If the SOAP binding is mandatory, is it allowable to** 653 **implement a subset of the assertions for that binding?**

654 The current specification suggests that a subset of the SOAP binding (only the authentication assertion, for
655 example) is allowable as satisfying this mandatory binding.