

DRUMMOND GROUP INC. TEST PLAN FOR LIBERTY ALLIANCE SAML TEST EVENT TEST CRITERIA

SAML 2.0

VERSION 3.0 ERRATA J

Prepared & Facilitated By:
DRUMMOND GROUP INC.
www.drummondgroup.com

Contents

www.drummondgroup.com	i
Contents	iii
Contents	iii
Test Plan Amendments	1
Introduction	2
Overview of Test Plan	2
Test Plan History	2
SAML Conformance Modes	2
GSA Profile	3
Technical Requirements	4
Metadata	4
IdP Authentication	4
Trivial Processing	4
Authentication Contexts	4
Name Identifier Formats	5
XML Signatures	5
XML Encryption.....	6
Attribute Profiles	6
Test Cases	7
Overview of Test Case Description	7
Test Cases Associated with Conformance Modes	7
General Test Case Requirements	8
Test Case A – Redirect Binding	8
Test Case B – SOAP Binding	11
Test Case C – POST Binding	14
Test Case D – Extended SAML Modes	17
Test Case E – IDP Introduction	20
Test Case F – Single Session Logout	21
Test Case G – Unsolicited <Response>	23
Test Case H – Affiliations	24
Test Case I – ECP	26
Test Case J – SAML Authentication Authority	27
Test Case K – SAML Attribute Authority	29
Test Case L – SAML Authorization Decision Authority	31
Test Case M – SAML URI Binding	33
Test Case N – Error Testing	34
Test Case O – GSA Profile	36
References	42
About Drummond Group	44

1 Test Plan Amendments

Date of Change	Document Version	Reason for Change	Summary of Change
09/20/07	3.0	Approved by TEG	
10/04/07	3.0.Errata A	Clarification and minor correction	Clarified MNI-Term step in TC A; corrected misuse of HTTP Redirect binding for Response message to POST binding in several test cases
10/10/07	3.0.Errata B	Clarification and minor correction	Called out general requirements for all test cases; Made use of SOAP consistent throughout Test Case B; inserted line numbers into layout
10/16/07	3.0 Errata C	Clarified Test Case D	Clarified which SPs (A or B) and IDPs (A or B) is acting in each test step
10/18/07	3.0 Errata D.	Clarified Test Case E	Changed to using HTTP Redirect binding to be consistent across test case
10/30/07	3.0 Errata E	Clarified Test Case G	Changed G.4 to use SOAP binding to be consistent with previous step of using Artifact Resolution through SOAP
10/31/07	3.0 Errata F	Clarified Test Case F	Clarified actions of User A in different browser sessions
11/01/07	3.0 Errata G	Clarified Test Case G	Removed the must of logging out user when federation is terminated.
11/05/07	3.0 Errata H	Clarified Test Case B, D, J, K and M	Corrected IDP used in D.3 and added SLO step in D.4. Clarified use of Redirect binding in TC B. Corrected TC M to show proper steps in SAML URI binding and request assertion by ID. Removed redundant request assertion by ID steps in TC J and K
11/13/07	3.0 Errata I	Clarified TC N	For TC H, changed steps 5 & 6 to make clear they are for backing system out of affiliations; For TC H, changed POST binding to Redirect; In, N.4 and N.5, changed "message payload" to "assertion to be more clear
11/27/07	3.0 Errata J	Clarified TC A	Made the MNI binding Redirect in TS 6

2

3 Introduction

4 **Overview of Test Plan**

5 This document is the Liberty SAML 2.0 Test Criteria Test Plan which contains the scope of the
6 technical requirements for Liberty certification of SAML 2.0. This document is intended to be
7 publicly viewable through the Liberty Alliance website as well as prospective test participants.

8 The contents of this document include the test cases for Liberty SAML 2.0 certification as well as
9 additional technical information relevant to testing. The test cases include different test steps which
10 as a whole cover the requirements of the SAML profiles [SAMLProf] and SAML conformance
11 modes [SAMLConf].

12 Another document, Liberty SAML 2.0 Process Test Plan, contains the detailed testing process and
13 test administration requirements for the SAML 2.0 certification test. The Liberty SAML 2.0 Process
14 Test Plan is available only to registered test participants. While the Process Test Plan is used in
15 completing a certification event, it is not needed to understand the technical expectation for
16 completing SAML 2.0 certification.

17 **Test Plan History**

18 This test plan replaces SAML 2.0 Interoperability Testing Procedure (vs. 2.0) test plan [SAMLTP2].
19 The majority of content is copied directly from [SAMLTP2] test plan, but main changes from the
20 previous version are the reconstructing of the test steps in to specific test cases and the addition test
21 step details.

- 22 • SAML 2.0 Interoperability Testing Procedure, vs. 2.0 (07/07/2006)
- 23 • SAML 2.0 Interoperability Testing Procedure, vs. 1.0 (2005)

24 **SAML Conformance Modes**

25 This test plan document contains test cases which cover the nine operational conformance modes of
26 SAML 2.0 and the specific features that are required or optional for each mode. The details of each
27 mode are provided in [SAMLConf], and the conformance modes a listed here:

- 28 • IdP – Identity Provider
- 29 • IdP Lite – Identity Provider Lite
- 30 • SP – Service Provider
- 31 • SP Lite – Service Provider Lite
- 32 • ECP – Enhanced Client/Proxy
- 33 • IdP Extended – Identify Provider Extended
- 34 • SP Extended – Service Provider Extended
- 35 • SAML Attribute Authority

- 36 • SAML Authorization Decision Authority
- 37 • SAML Authentication Authority
- 38 • SAML Requester

39 Each conformance mode requires different test cases, but some test cases cover multiple
40 conformance modes. The required test cases for each conformance mode are noted in the Test Case
41 section of this document.

42 Certification in conformance modes IdP Extended and SP Extended can only be given if a
43 participant has met the certification requirements of one of the standard SP or IdP modes.

44 Because significant features in some of these modes are Optional the Liberty Interoperability Testing
45 Program has created an additional designation “Complete” to recognize and differentiate
46 implementations that demonstrate interoperability of all optional features for a particular mode. The
47 list of “Complete” interoperability designations is:

- 48 • SP Complete
- 49 • SAML Requester Complete

50 **GSA Profile**

51 The GSA Profile [test case](#) is an optional test case. It follows the SAML 2.0 requirements for the
52 General Service Administration (GSA) of the US Government. The technical requirements for this
53 test case come from the GSA SAML Profile in [GSAInterface], [GSAAdoptSchm] and
54 [GSATechAppr]. These documents should be consulted for further explanation of the GSA
55 requirements.

56 **POST Binding**

57 Although the POST binding is not included in the SAML SCR, it is permitted with the SAML
58 specification and has some user deployment. POST Binding is an optional Liberty designation
59 conformance mode. It involves use of POST binding for AuthnRequest, Name ID Management and
60 SLO. Certification in the POST Binding mode is done through successfully completing this [test case](#).

61 Technical Requirements

62 Metadata

63 There are no normative requirements in [SAMLConf] regarding the content or processing of
64 metadata as described in [SAMLMeta]. However, for purposes of this certification event,
65 implementations are required to:

- 66 • Furnish correct metadata, and
- 67 • Process metadata furnished by other testing partners

68 While metadata is not specified for SAML Attribute Requesters, interoperability with SAML
69 Authorities is very difficult without it, and for this certification event it is required that SAML
70 Attribute Requesters provide metadata as described in the draft metadata extension specification
71 [SAMLMetaExt]. It is not necessary or meaningful for an ECP to produce or consume metadata.

72 IdP Authentication

73 SAML does not normatively specify any requirements for user authentication at IdP for Web SSO.
74 In fact, user authentication is explicitly described as “out of scope” [SAMLProf]. However, for
75 purposes of interoperability testing, it is required that IdP implementations offer at least one of these
76 authentication methods:

- 77 1. HTTP Basic Auth.
- 78 2. HTTP Form Post
- 79 3. HTTP Get

80 Similarly, it is required that user agents, particularly ECP implementations, be able to authenticate
81 using at least one of these methods.

82 Trivial Processing

83 Several features specified by SAML (e.g., IdP Proxy) can be implemented such that any request
84 simply returns an error response. While this trivial behavior is, strictly speaking, in conformance
85 with the specifications, it is not meaningful in the context of interoperability testing. Except where
86 explicitly indicated (e.g., for certain Name Identifier formats) all testing steps will require non-trivial
87 responses in order to be deemed successful.

88 Authentication Contexts

89 Some of the SAML Modes rely on a well-defined ordering of authentication contexts. The SAML
90 specifications do not normatively specify an ordering [SAMLAuthnCxt] and leave the comparison
91 decisions up to the implementation [SAMLCore]. However, for purposes of testing we will
92 arbitrarily define an ordering of authentication contexts to be used in the tests. This arbitrary listing
93 of authentication class URIs, in order of increasing strength, is:

- 94 1. any defined authentication context not listed below
- 95 2. urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession

96 3. urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol

97 4. urn:oasis:names:tc:SAML:2.0:ac:classes>Password

98 This ordering should be observed by all implementations testing SAML modes where authentication
99 contexts must be compared. The overall concept of the testing of the Authentication Authority is to
100 create several different assertions using different authentication contexts. Then these are queried
101 using the query terms (“exact”, “better”, “maximum”, “minumum”) and a reference authentication
102 context.

103 NOTE: Complete implementation of these authentication contexts is not required. These
104 authentication context URIs should simply be asserted in requests and responses to demonstrate
105 interoperability of authentication context processing rules.

106 **Name Identifier Formats**

107 The following Name Identifier Formats are defined by [SAMLCore]:

108 1. Unspecified

109 2. Email

110 3. X.509 Subject

111 4. Windows

112 5. Kerberos

113 6. Entity

114 7. Persistent

115 8. Transient

116 Every implementation is required to accept messages containing any of these formats, but
117 [SAMLCore] only requires that the last two be processed.

118 **XML Signatures**

119 The [SAMLConf] does not specifically indicate where XML Signatures are required, but the
120 underlying specifications in [SAMLProf] make signing required for certain profiles. Specifically,
121 these are:

122 1. Web SSO: The assertion element(s) in the <Response> MUST be signed for the HTTP POST
123 binding.

124 2. ECP Profile: The assertion element(s) in the <Response> issued by the IdP MUST be signed.

125 3. Single Logout: The <LogoutRequest> and <LogoutResponse> MUST be signed for the
126 HTTP redirect binding.

127 4. Name Identifier Management: The <ManageNameIDRequest> and
128 <ManageNameIDResponse> MUST be signed for the HTTP redirect binding.

129 SP and IdP implementations may indicate via metadata a desire for requests or responses to be
130 signed for other bindings than those indicated above. However, such stipulations in metadata are not
131 binding and adherence is not required.

132 **XML Encryption**

133 [SAMLConf] stipulates several different encryption algorithms and key transport mechanisms that
134 MUST be implemented. However, these testing procedures do not require demonstration of support
135 for all these combinations and instead rely on successful interoperability as a measure of
136 conformance. Implementations should take care to ensure that elements to be encrypted include any
137 XML namespace prefix declarations so that, when decrypted, the element will remain valid
138 independent of context. One method for achieving this is described in [ExcXMLCan], but other
139 approaches will work.

140 Note that while the <ds:KeyInfo> and <xenc:EncryptedKey> elements are not required in the SAML
141 specifications or related schemas, these elements MUST be included in messages for interoperability
142 testing. There is no normative mechanism for exchanging these keys out-of-band. The precise
143 location of these elements in the message is underspecified; the most common practice among
144 interoperable SAML implementations is that in each encrypted element there be one
145 <xenc:EncryptedKey> element in parallel with the <xenc:EncryptedData>, and that this
146 <xenc:EncryptedKey> be inferred as the relevant key information for decryption without relying on
147 any references within the subelements. An erratum has been created to clarify this; see PE43 in
148 [SAMLErrata]. For this certification event, this most common practice stated above SHOULD be
149 done.

150 Finally, encryption coupled with deflation and URL encoding may create URLs that exceed the
151 maximum length supported by some browsers. Consequently, encryption is contraindicated for the
152 MNI HTTP-Redirect testing steps.

153 **Attribute Profiles**

154 [SAMLConf] makes no normative statements about which Attribute Profiles in [SAMLProf] are
155 required to be supported by SAML Attribute Authority or a SAML Requestor. These are the profiles
156 described in [SAMLProf] except for X.500/LDAP which is described in [SAMLLDAP]:

- 157 1. Basic
- 158 2. X.500/LDAP
- 159 3. UUID
- 160 4. DCE PAC
- 161 5. XACML

162 Of these, this document only describes testing procedures for the Basic profile, and does not describe
163 any testing procedures regarding the other profiles.

164 Test Cases

165 **Overview of Test Case Description**

166 Each test case is setup with the first part listing an overview of the test steps in the test case. The
167 second part describes the details of the individual test steps to carry out the test case. The test step
168 overview lists the sequence of test steps along with a general description of the message or action or
169 configuration setting required. The test step details provide more information on the expected test
170 steps.
171

172 **Test Cases Associated with Conformance Modes**

173 In order to achieve certification in one or more of the Liberty SAML Conformance Modes, the
174 associated test cases must be completed with all test participants with aligning modes. For example,
175 a product testing for an IdP conformance mode must complete Test Cases A, B, C, E, F, G, H and I
176 against all products testing for a SP and SP Lite conformance mode. The specific pairing among
177 participants will be given at the beginning of the certification event. A conformance mode may not
178 require the completing of all the test steps in the associated test cases. The individual test cases
179 provide details of test steps that may or must be omitted depending on the conformance mode.
180

Conformance Mode	Test Cases
IdP	A, B, E, F, G, H, I, N
IdP Extended	D
IdP Lite	A, B, E, F, G, H, I
SP	A, B, C, E, F, G, H, I, N
SP Extended	D
SP Lite	A, B, C, E, F, G, H, I, N
ECP	I
SAML Attribute Authority	K
SAML Authorization Decision Authority	L
SAML Authentication Authority	J
SAML Requester	M

181

182 **General Test Case Requirements**

183 For all test cases, the following requirements are to be followed unless a test case specifically states
184 otherwise:

- 185 • SAML AuthnRequest MUST be signed.
- 186 • For POST bindings, the assertion MUST be signed.
- 187 • For POST bindings, the entire response message MAY be signed, but if signed, the receiving
188 partner MUST validate the signature.
- 189 • Encryption of NameIDs and Assertions MUST be enabled.

190 **Test Case A – Redirect Binding**

191 **Preconditions: Metadata exchanged and loaded**

192 **Conformance Modes: IdP, SP, IdP Lite, SP Lite**

193 **NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps**

194

195 **Test Step Overview**

Steps	Action/Message/Setting
1	Encryption Enabled
2	Web SSO HTTP redirect / Persistent / Federate
3	MNI IdP-Initiated / HTTP redirect (signed)
4	SLO SP-Initiated / HTTP redirect (signed)
5	Web SSO HTTP redirect / Not Federated
6	SLO IdP-initiated / HTTP redirect (signed)
	Destroy Federation and NameIds
7	Web SSO HTTP redirect / Federate
8	MNI SP-Initiated / HTTP redirect (signed)
9	SLO SP-Initiated / HTTP redirect (signed)
10	Web SSO HTTP redirect
11	SLO IdP-Initiated / HTTP redirect (signed)
12	Encryption Disabled

196

197

198 **Test Steps Details**

199 1. User is logged into SP. IdP enables encryption of NameId and Assertion.

200 IdP CONFIRM: IdP has enabled encryption for NameId and Assertion.

201

202 2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where

203 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is

204 through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML
205 Response message through HTTP POST binding.

206 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
207 HTTP Redirect binding.

208 IdP CONFIRM: User has been federated

209 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

210

211 3. IdP sends signed ManageNameIdRequest message to the SP using HTTP Redirect binding. SP
212 returns signed ManageNameIdResponse message using HTTP Redirect binding. SP Lite and IdP
213 Lite modes omit this step.

214 SP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.

215 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

216

217 4. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User
218 session. IdP returns a signed LogoutResponse message.

219 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

220 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

221

222 5. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE.
223 SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect
224 binding. IdP provides assertion of User and IdP returns a signed SAML Response message through
225 HTTP POST binding.

226 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
227 HTTP Redirect binding.

228 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

229

230 6. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect
231 binding. SP returns a signed LogoutResponse message. IdP sends signed ManageNameIdRequest
232 message with the Terminate element to the SP using HTTP Redirect binding. Federation for User is
233 terminated. SP returns signed ManageNameIdResponse message using HTTP Redirect binding. User
234 logs out of session.

235 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

236 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

237 SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP
238 Redirect binding.

239 SP CONFIRM: User federation is terminated.

240 SP CONFIRM: User logs out of session.

241 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

242 IdP CONFIRM: User federation is terminated.

243 IdP CONFIRM: User logs out of session.

244

245 7. User/SP does Single Sign-On with Federate where AllowCreate is set to TRUE. SP
246 communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.
247 IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP
248 POST binding.

249 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
250 HTTP Redirect binding.
251 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
252
253 8. SP sends signed ManageNameIdRequest message to the IdP using HTTP Redirect binding. IdP
254 returns signed ManageNameIdResponse message using HTTP Redirect binding. SP Lite and IdP
255 Lite modes omit this step.
256 IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.
257 SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.
258
259 9. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User
260 session. IdP returns a signed LogoutResponse message.
261 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
262 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
263
264 10. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
265 Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed
266 SAML Response message through HTTP POST binding.
267 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
268 HTTP Redirect binding.
269 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
270
271 11. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP
272 Redirect binding. SP returns a signed LogoutResponse message.
273 SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
274 IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
275
276 12. IdP disables encryption of NameIDs and Assertions. Steps 2-11 are repeated.
277 IDP CONFIRM: IdP has disabled encryption for NameId and Assertion.
278 IDP/SP CONFIRM: Steps 2-11 are successfully executed.

279 **Test Case B – SOAP Binding**

280 **Preconditions: Metadata exchanged and loaded**

281 **Conformance Modes: IdP, SP, IdP Lite, SP Lite**

282 **NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps**

283

284 **Test Step Overview**

Steps	Action/Message/Setting
1	Encrypted IDs and Assertions
2	Web SSO Artifact / Persistent / Federate / Artifact Resolution SOAP
3	MNI IdP-Initiated / SOAP (SP may use HTTP Redirect)
4	SLO SP-Initiated / SOAP (SP Lite / IdP Lite may use HTTP Redirect)
5	Web SSO Artifact / Persistent / Not Federated / Artifact Resolution SOAP
6	SLO IdP-initiated / SOAP (SP Lite / IdP Lite may use HTTP Redirect)
7	Web SSO Artifact / Artifact Resolution SOAP
8	MNI SP-Initiated / SOAP (SP may use HTTP Redirect)
9	SLO IdP-Initiated / SOAP (SP Lite / IdP Lite may use HTTP Redirect)
10	Web SSO Artifact / Artifact Resolution SOAP
11	SLO IdP-Initiated / HTTP redirect (signed)

285

286 **Test Steps Details**

287 1. User is logged into SP. IdP enables encryption of NameId.and Assertions.

288 IdP CONFIRM: IdP has enabled encryption.

289

290 2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where
291 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
292 sent through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML
293 Response message through Artifact binding. The IdP and SP resolve the artifact via a SOAP binding.

294 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
295 HTTP Redirect binding.

296 IdP CONFIRM: Artifact resolution is properly done.

297 IdP CONFIRM: User has been federated

298 SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.

299 SP CONFIRM: Artifact resolution is properly done.

300

301 3. IdP sends signed ManageNameIdRequest message to the SP using SOAP binding (SP modes may
302 use HTTP Redirect binding). SP returns signed ManageNameIdResponse message using SOAP
303 binding. SP Lite and IdP Lite modes omit this step.
304 SP CONFIRM: Receives signed ManageNameIdRequest on SOAP binding (or possibly
305 HTTP Redirect binding for SP modes).
306 IdP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding (or possibly
307 HTTP Redirect binding for SP modes).
308

309 4. SP sends a signed LogoutRequest message to IdP using SOAP binding (SP Lite and IdP Lite
310 modes may use HTTP Redirect binding). IdP logs out User session. IdP returns a signed
311 LogoutResponse message.
312 IdP CONFIRM: Receives signed LogoutRequest on SOAP binding (or possibly HTTP
313 Redirect binding for SP Lite and IdP Lite modes).
314 SP CONFIRM: Receives signed LogoutResponse on SOAP binding (or possibly HTTP
315 Redirect binding for SP Lite and IdP Lite modes).
316

317 5. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE.
318 SP communication to the IdP for the SAML Authentication Request is sent through HTTP Redirect
319 binding. IdP provides assertion of User and IdP returns a signed SAML Response message through
320 Artifact binding. The IdP and SP resolve the artifact via a SOAP binding.
321 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
322 HTTP Redirect Binding.
323 IdP CONFIRM: Artifact resolution is properly done.
324 SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.
325 SP CONFIRM: Artifact resolution is properly done.
326

327 6. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using SOAP binding
328 (SP Lite and IdP Lite modes may use HTTP Redirect binding). SP returns a signed LogoutResponse
329 message.
330 IdP CONFIRM: Receives signed LogoutRequest on SOAP binding (or possibly HTTP
331 Redirect binding for SP Lite and IdP Lite modes).
332 SP CONFIRM: Receives signed LogoutResponse on SOAP binding (or possibly HTTP
333 Redirect binding for SP Lite and IdP Lite modes).
334

335 7. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
336 Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed
337 SAML Response message through Artifact binding. The IdP and SP resolve the artifact via SOAP
338 binding.
339 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
340 HTTP Redirect binding.
341 IdP CONFIRM: Artifact resolution is properly done.
342 IdP CONFIRM: User has been federated
343 SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.
344 SP CONFIRM: Artifact resolution is properly done.
345

346 8. SP sends signed ManageNameIdRequest message to the IdP using SOAP binding (SP modes may
347 use HTTP Redirect binding). IdP returns signed ManageNameIdResponse message using SOAP
348 binding. SP Lite and IdP Lite modes omit this step.
349 IdP CONFIRM: Receives signed ManageNameIdRequest on SOAP binding (or possibly
350 HTTP Redirect binding for SP modes).
351 SP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding (or possibly
352 HTTP Redirect binding for SP modes).
353

354 9. IdP logs out User session. IdP sends a signed LogoutRequest message to IdP using SOAP binding
355 (SP Lite and IdP Lite modes may use HTTP Redirect binding). SP returns a signed LogoutResponse
356 message.
357 SP CONFIRM: Receives signed LogoutRequest on SOAP binding (or possibly HTTP
358 Redirect binding for SP Lite and IdP Lite modes).
359 IdP CONFIRM: Receives signed LogoutResponse on SOAP binding (or possibly HTTP
360 Redirect binding for SP Lite and IdP Lite modes).
361

362 10. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
363 Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed
364 SAML Response message through Artifact binding. The IdP and SP resolve the artifact via a SOAP
365 binding.
366 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
367 HTTP Redirect binding.
368 IdP CONFIRM: Artifact resolution is properly done.
369 SP CONFIRM: IdP returns signed SAML Response through HTTP Artifact binding.
370 SP CONFIRM: Artifact resolution is properly done.
371

372 11. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using SOAP binding
373 (SP Lite and IdP Lite modes may use HTTP Redirect binding). SP returns a signed LogoutResponse
374 message.
375 IdP CONFIRM: Receives signed LogoutRequest on SOAP binding (or possibly HTTP
376 Redirect binding for SP Lite and IdP Lite modes).
377 SP CONFIRM: Receives signed LogoutResponse on SOAP binding (or possibly HTTP
378 Redirect binding for SP Lite and IdP Lite modes).

379 **Test Case C – POST Binding**
 380 **Preconditions: Metadata exchanged and loaded**
 381 **Conformance Modes: POST Binding**
 382

383 **Test Step Overview**

Steps	Action/Message/Setting
1	Encrypted Enabled
2	Web SSO / POST (signed) / Persistent / Federate
3	MNI IdP-Initiated / POST (signed)
4	SLO SP-Initiated / POST (signed)
5	Web SSO POST (signed) / Not Federated
6	SLO IdP-initiated / POST (signed)
7	Web SSO POST (signed)
8	MNI IdP-initiated / POST / Terminate
9	Web SSO POST (signed)
10	MNI SP-Initiated / POST (signed)
11	SLO SP-Initiated / POST (signed)
12	Web SSO POST (signed)
13	SLO IdP-Initiated / POST (signed)

384
 385 **Test Steps Details**

- 386 1. User is logged into SP. IdP enables encryption of NameId and Assertion.
 387 IdP CONFIRM: IdP has enabled encryption.
 388
- 389 2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where
 390 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
 391 through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML
 392 Response message through HTTP POST binding.
 393 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
 394 HTTP POST binding.
 395 IdP CONFIRM: User has been federated
 396 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
 397
- 398 3. IdP sends signed ManageNameIdRequest message to the SP using HTTP POST binding. SP
 399 returns signed ManageNameIdResponse message using HTTP POST binding.
 400 SP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.
 401 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.
 402
- 403 4. SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs out User
 404 session. IdP returns a signed LogoutResponse message.

405 IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
406 SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.
407
408 5. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE.
409 SP communication to the IdP for the SAML Authentication Request is through HTTP POST
410 binding. IdP provides assertion of User and IdP returns a signed SAML Response message through
411 HTTP POST binding.
412 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
413 HTTP POST binding.
414 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
415
416 6. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP POST
417 binding. SP returns a signed LogoutResponse message.
418 IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
419 SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.
420
421 7. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
422 Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed
423 SAML Response message through HTTP POST binding.
424 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
425 HTTP POST binding.
426 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
427
428 8. IdP sends signed ManageNameIdRequest message with the Terminate element to the SP using
429 HTTP POST binding. User session is terminated. SP returns signed ManageNameIdResponse
430 message using HTTP POST binding.
431 SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP
432 POST binding.
433 SP CONFIRM: User session is terminated.
434 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.
435 IdP CONFIRM: User session is terminated.
436
437 9. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
438 Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed
439 SAML Response message through HTTP POST binding.
440 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
441 HTTP POST binding.
442 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
443
444 10. SP sends signed ManageNameIdRequest message to the IdP using HTTP POST binding. IdP
445 returns signed ManageNameIdResponse message using HTTP POST binding.
446 IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.
447 SP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.
448

449 11. SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs out User
450 session. IdP returns a signed LogoutResponse message.
451 IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
452 SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.
453
454 12. User/SP does Single Sign-On. SP communication to the IdP for the SAML Authentication
455 Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed
456 SAML Response message through HTTP POST binding.
457 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
458 HTTP POST binding.
459 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
460
461 13. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP POST
462 binding. SP returns a signed LogoutResponse message.
463 SP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
464 IdP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.
465
466

467 **Test Case D – Extended SAML Modes**

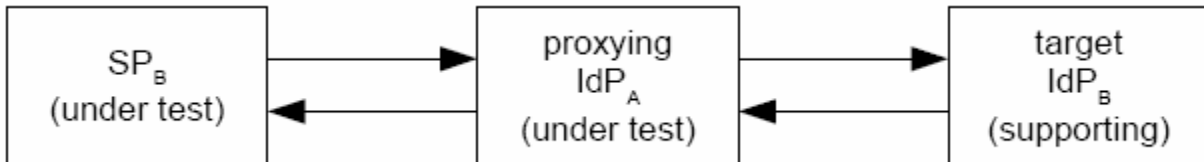
468 **Preconditions: Metadata exchanged and loaded**

469 **Conformance Modes: IdP Extended, SP Extended**

470

471 **Background on IdP Proxy**

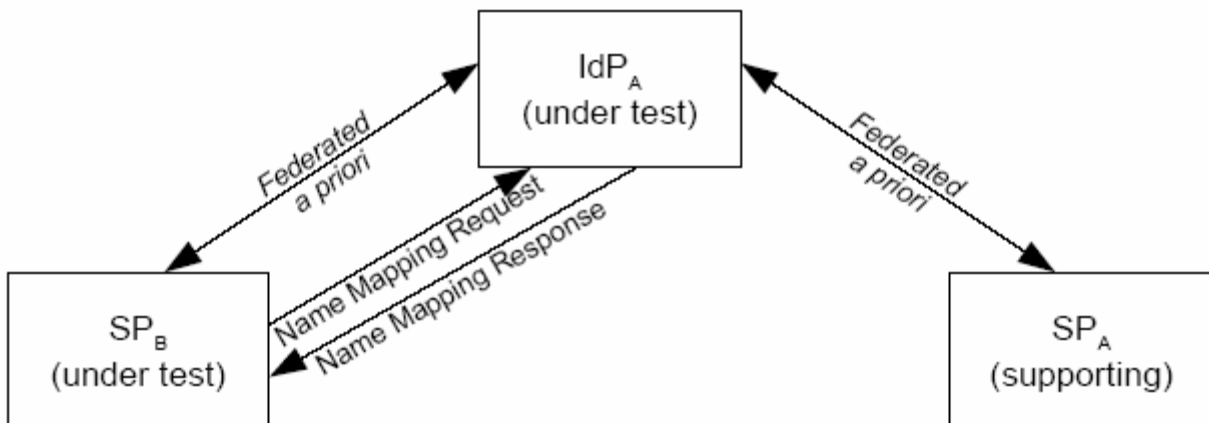
472 The IdP Proxy feature requires two IdP implementations and one SP implementation. If we
 473 have teams A and B, the following diagram depicts the roles of the test participants,
 474 assuming that IdP_A and SP_B are the implementations under test:



475

476 **Background on Name Identifier Mapping Feature**

477 The name identifier mapping feature requires that an IdP provide an indirect reference for a
 478 principal at SP_A in response to a request from SP_B. Assuming again that teams A and B are
 479 testing IdP_A and SP_B, it is necessary for the principal to federate her identity at both SP_B and
 480 SP_A with IdP_A. This can be depicted as follows:



481

482

483

Test Step Overview

Steps	Action/Message/Setting
1	Encryption Enabled
2	ProxyCount=0 (proxy disallowed)
3	Web SSO HTTP Redirect (signed) to IdP _A
4	SLO SP-initiated / HTTP Redirect
5	ProxyCount missing (proxy allowed)
6	Web SSO / HTTP Redirect (signed) to IdP _A
7	SLO SP-initiated / HTTP Redirect

8	ProxyCount=1 (proxy allowed)
9	Web SSO / HTTP Redirect (signed)
10	SLO SP-initiated / HTTP Redirect
11	Web SSO HTTP Redirect (signed) / Persistent
12	SLO IdP-initiated / HTTP Redirect
13	NameIDMappingRequest / NameIDMappingResponse

- 484
485 **Test Steps Details**
486 1. IdP_A and IdP_B enables encryption of NameId and Assertion.
487 IdP_A CONFIRM: IdP_A has enabled encryption.
488 IdP_B CONFIRM: IdP_B has enabled encryption.
489
490 2. SP sets ProxyCount=0 where proxy is disallowed.
491 SP CONFIRM: SP has disallowed proxy.
492
493 3. User/SP does Single Sign-On. SP communication to the IdP_A for the SAML Authentication
494 Request is through HTTP Redirect binding. IdP provides assertion of User and IdP_A returns a signed
495 SAML Response message through HTTP POST binding.
496 IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through
497 HTTP Redirect binding.
498 SP CONFIRM: IdP_A returns signed SAML Response through HTTP POST binding.
499
500 4. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A logs out
501 User session. IdP_A returns a signed LogoutResponse message.
502 IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
503 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
504
505 5. SP removes ProxyCount where proxy is allowed.
506 SP CONFIRM: SP has allowed proxy.
507
508 6. User/SP does Single Sign-On. SP communication to the IdP_A for the SAML Authentication
509 Request is through HTTP Redirect binding. IdP_A proxies the Authentication Request to IdP_B. IdP_B
510 provides assertion of User to IdP_A in a SAML Response message through HTTP POST binding and
511 IdP_A returns a signed SAML Response message through HTTP POST binding.
512 IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through
513 HTTP Redirect binding.
514 IdP_B CONFIRM: IdP_A proxies Authentication Request.
515 SP CONFIRM: IdP_A returns signed SAML Response through HTTP POST binding.
516
517 7. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A logs out
518 User session. IdP_A returns a signed LogoutResponse message.
519 IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
520 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

521
522 8. SP sets ProxyCount=1 where proxy is allowed.
523 SP CONFIRM: SP has allowed 1 proxy.
524
525 9. User/SP does Single Sign-On. SP communication to the IdP_A for the SAML Authentication
526 Request is through HTTP Redirect binding. IdP_A proxies the Authentication Request to IdP_B. IdP_B
527 provides assertion of User to IdP_A in a SAML Response message through HTTP POST binding and
528 IdP_A returns a signed SAML Response message through HTTP POST binding.
529 IdP_A CONFIRM: SP successfully communicated SAML Authentication Request through
530 HTTP Redirect binding.
531 IdP_B CONFIRM: IdP_A proxies Authentication Request.
532 SP CONFIRM: IdP_A returns signed SAML Response through HTTP POST binding.
533
534 10. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A logs out
535 User session. IdP_A returns a signed LogoutResponse message.
536 IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
537 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
538
539 11. User/SP_A does Single Sign-On with Persistent Name Identifier. SP_A communication to the IdP
540 for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of
541 User and IdP returns a signed SAML Response message through HTTP POST binding.
542 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request through
543 HTTP POST binding.
544 IdP CONFIRM: User has been federated
545 SP_A CONFIRM: IdP returns signed SAML Response through HTTP Redirect binding.
546
547 12. IdP logs out User session. IdP sends a signed LogoutRequest message to SP_A using HTTP
548 Redirect binding. SP_A returns a signed LogoutResponse message.
549 SP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
550 IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
551
552 13. SP_B sends NameIdMapping Request message to the IdP requesting an alternative name identifier
553 for User. IdP maps the request to the User identity from SP_A. IdP returns signed
554 NameIdMappingResponse message using HTTP POST binding.
555 IdP CONFIRM: Receives NameIdMapping Request.
556 SP CONFIRM: Receives NameIdMappingResponse Response.
557
558

559 **Test Case E – IDP Introduction**

560 **Preconditions: Metadata exchanged and loaded**

561 **Conformance Modes: IdP, SP, IdP Lite, SP Lite**

562 **NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps**

563

564 **Test Step Overview**

Steps	Action
1	Enables Encryption / Clear Cookies
2	IdP Login / Federate / Set Cookie
3	SSO at SP using common domain cookie
4	MNI Destroy Federation / IdP-Initiated / HTTP Redirect

565

566 **Test Step Detail**

567 1. IdP enables encryption of NameId and Assertion. Cookies are cleared from User Browser

568 IdP CONFIRM: IdP has enabled encryption.

569

570 2. User logs in at IdP with Federation. Cookie is set.

571 IdP CONFIRM: User logged in cookie is set.

572

573 3. User/SP does Single Sign-On using a common domain cookie. SP communication to the IdP for
574 the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User
575 and IdP returns a signed SAML Response message through HTTP POST binding.

576 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
577 HTTP Redirect binding.

578 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

579

580 4. IdP sends signed ManageNameIdRequest message with the Terminate element to the SP using
581 HTTP Redirect binding. User session is terminated. SP returns signed ManageNameIdResponse
582 message using HTTP Redirect binding.

583 SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP
584 Redirect binding.

585 SP CONFIRM: User session is terminated.

586 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

587 IdP CONFIRM: User session is terminated.

588 **Test Case F – Single Session Logout**

589 **Preconditions: Metadata exchanged and loaded**

590 **Conformance Modes: IdP, SP, IdP Lite, SP Lite**

591 **NOTE – IdP Lite and SP Lite actors are to ignore Name ID Management steps**

592

593 **Test Step Overview**

Steps	Action/Message/Setting
1	Web SSO (Browser A) / Federate / HTTP Redirect
2	Web SSO (Browser B) HTTP Redirect
3	SLO (Browser A) SP-Initiated / HTTP redirect (signed) Browser B session remains active
4	Web SSO (Browser A) HTTP Redirect
5	SLO (Browser A) IdP-Initiated / HTTP redirect (signed) Browser B session remains active
6	MNI SP-initiated (Browser B) / Redirect (signed) / Terminate

594

595

596 **Test Steps**

597 1. User A/Browser A does Single Sign-On with Federate where AllowCreate is set to TRUE. SP
598 communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.
599 IdP provides assertion of User A and IdP returns a signed SAML Response message through HTTP
600 POST binding.

601 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
602 HTTP Redirect binding.

603 IdP CONFIRM: User A has been federated.

604 IdP CONFIRM: User A has been logged in through Browser A (Session A).

605 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

606

607 2. User A/Browser B does Single Sign-On. SP communication to the IdP for the SAML
608 Authentication Request is through HTTP Redirect binding. IdP provides assertion of User B and IdP
609 returns a signed SAML Response message through HTTP POST binding.

610 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
611 HTTP Redirect binding.

612 IdP CONFIRM: User B has been logged in through Browser B (Session B).

613 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

614

615 3. User A/Browser A logs off of SP. SP sends a signed LogoutRequest message to IdP using HTTP
616 Redirect binding. IdP logs out User A in Browser A (Session A). User A remains logged in through
617 Browser B (Session B). IdP returns a signed LogoutResponse message.

618 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

619 IdP CONFIRM: User A logs out in Browser A (Session A).

620 IdP CONFIRM: User A is logged in through Browser B (Session B).

621 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
622 SP CONFIRM: User A logs out in Browser A (Session A).
623 IdP CONFIRM: User A is logged in through Browser B (Session B).
624
625 4. User A/Browser A does Single Sign-On. SP communication to the IdP for the SAML
626 Authentication Request is through HTTP Redirect binding. IdP provides assertion of User A and IdP
627 returns a signed SAML Response message through HTTP POST binding.
628 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
629 HTTP Redirect binding.
630 IdP CONFIRM: User A has been logged in through Browser A (Session A).
631 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
632
633 5. User A/Browser A logs off of IdP. IdP sends a signed LogoutRequest message to SP using HTTP
634 Redirect binding. SP logs out User A in Browser A (Session A). User A remains logged in through
635 Browser B (Session B). SP returns a signed LogoutResponse message.
636 SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
637 SP CONFIRM: User A logs out in Browser A (Session A).
638 SP CONFIRM: User A is logged in through Browser B (Session B).
639 IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
640 IdP CONFIRM: User A logs out in Browser A (Session A).
641 IdP CONFIRM: User A is logged in through Browser B (Session B).
642
643 6. SP sends signed ManageNameIdRequest message with the Terminate element to the IdP using
644 HTTP Redirect binding. Federation for User A is terminated. IdP returns signed
645 ManageNameIdResponse message using HTTP Redirect binding. User A logs out of Browser B
646 (Session B).
647 IdP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP
648 Redirect binding.
649 IdP CONFIRM: Federation of User A is terminated.
650 IdP CONFIRM: User A is logged out in Browser B (Session B).
651 SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.
652 SP CONFIRM: Federation of User A is terminated.
653 SP CONFIRM: User A is logged out in Browser B (Session B).
654
655

656 **Test Case G – Unsolicited <Response>**

657 **Preconditions: Metadata exchanged and loaded**

658 **Conformance Modes: IdP, SP, IdP Lite, SP Lite**

659

660 **Test Step Overview**

Steps	Action
1	IdP Unsolicited SSO Response / Transient / HTTP POST (signed)
2	SLO SP-Initiated / HTTP redirect (signed)
3	IdP Unsolicited SSO Response / Transient / HTTP artifact / Artifact Resolution (SOAP)
4	SLO IdP-Initiated / HTTP redirect (signed)

661

662 **Test Step Detail**

663 1. User/IdP does Single Sign-On. IdP provides assertion of User and Name ID is Transient. IdP
664 sends a signed SAML Response message through HTTP POST binding.

665 IdP CONFIRM: User has been federated

666 SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.

667

668 2. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User
669 session. IdP returns a signed LogoutResponse message.

670 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

671 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

672

673 3. User/IdP does Single Sign-On. IdP provides assertion of User and Name ID is Transient. IdP
674 sends a signed SAML Response message through HTTP Artifact using an HTTP Redirect URL. The
675 IdP and SP resolve the artifact via a SOAP binding.

676 IdP CONFIRM: Artifact resolution is properly done.

677 IdP CONFIRM: User has been federated

678 SP CONFIRM: IdP sends signed SAML Response through HTTP Artifact.

679 SP CONFIRM: Artifact resolution is properly done.

680

681 4. IdP sends a signed LogoutRequest message to SP using SOAP binding. SP logs out User session.
682 SP returns a signed LogoutResponse message.

683 SP CONFIRM: Receives signed LogoutRequest on SOAP binding.

684 IdP CONFIRM: Receives signed LogoutResponse on SOAP binding.

685

686

687 **Test Case H – Affiliations**

688 **Preconditions: Metadata exchanged and loaded**

689 **Conformance Modes: IdP, SP, IdP Lite, SP Lite**

690

691 **Test Step Overview**

Steps	Action
1	SPNameQualifier=[affiliation id]
2	Web SSO HTTP Redirect / Persistent / Federate
3	SLO IdP-initiated / HTTP Redirect (signed)
4	Web SSO HTTP Redirect / Not Federate
5	SLO SP-initiated / HTTP Redirect (signed)
6	SPNameQualifier=[sp provider id]

692

693 **Test Step Detail**

694 1. SP sets Name Qualifier to Affiliation ID.

695 SP CONFIRM: SPNameQualifier=[affiliation id]

696

697 2. User/SP does Single Sign-On with Persistent Name Identifier and with Federate where
698 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
699 through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML
700 Response message through HTTP POST binding.

701 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
702 HTTP POST binding.

703 IdP CONFIRM: User has been federated

704 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

705

706 3. IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect
707 binding. SP returns a signed LogoutResponse message.

708 SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

709 IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

710

711 4. User/SP does Single Sign-On with Federation already done where AllowCreate is set to FALSE.
712 SP communication to the IdP for the SAML Authentication Request is through HTTP POST
713 binding. IdP provides assertion of User and IdP returns a signed SAML Response message through
714 HTTP POST binding.

715 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
716 HTTP POST binding.

717 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

718

719 5. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User
720 session. IdP returns a signed LogoutResponse message.

721 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

722 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
723
724 6. SP sets Name Qualifier to SP Provider ID.
725 SP CONFIRM: SPNameQualifier=[sp provider id]
726 NOTE: It is recommended that the SP verifies it has changed back the SPNameQualifier
727 through a SSO/SLO test with other participants.
728
729

730 **Test Case I – ECP**

731 **Preconditions: Metadata exchanged and loaded**

732 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, ECP**

733

734 **Test Step Overview**

Steps	Action
1	Federate (NameIDPolicy, AllowCreate=True)
2	Enhanced ClientProxy SSO, PAOS
3	ECP conveys Response to SP

735

736 **Test Step Detail**

737 1. User attempts to access SP through ECP. Settings are Persistent Name Identifier and with
738 Federate where AllowCreate is set to TRUE to SP.

739 ECP CONFIRM: Connects to SP.

740 SP CONFIRM: ECP connects.

741

742 2. SP issues SAML AuthnRequest message to ECP through PAOS binding. ECP sends SAML
743 Request message to IdP through SOAP binding. IdP returns assertion in SAML Response to ECP.

744 IdP CONFIRM: ECP successfully communicated SAML Authentication Request through
745 SOAP binding.

746 SP CONFIRM: IdP returns signed SAML Response through HTTP SOAP binding.

747

748 3. ECP conveys Response to SP. SP grants access to User. User closes browser and session ends.

749 SP CONFIRM: Proper Response is received.

750 ECP CONFIRM: User is granted access to SP.

751

752 **Test Case J – SAML Authentication Authority**
 753 **Conformance Modes: SAML Authentication Authority**
 754 **Preconditions: Metadata exchanged and loaded**
 755 **Note: Section [[AuthenticationContexts](#)] within this document describes the strength of**
 756 **the AuthnContext classes used for comparison.**

757
758 **Test Step Overview**

Steps	Action/Message/Setting
1	Web SSO / POST (signed) / Persistent
2	AC Comparison="exact" / HTTP Basic Authentication
3	Authentication Query / SOAP
4	AC Comparison="better"
5	Authentication Query / SOAP
6	AC Comparison="minimum"
7	Authentication Query / SOAP
8	AC Comparison="maximum"
9	Authentication Query / SOAP

759
760 **Test Step Detail**

761 1. User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for
 762 the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User
 763 and IdP returns a signed SAML Response message through HTTP POST binding.
 764 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
 765 HTTP POST binding.
 766 IdP CONFIRM: User has been federated
 767 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
 768
 769 2. SAML Requester sets AC comparison to “exact”. SAML Requester/Responder enables HTTP
 770 Basic Authentication.
 771 SAML Requester CONFIRM: AC comparison="exact".
 772 SAML Requester CONFIRM: HTTP Basic Authentication enabled.
 773 SAML Responder CONFIRM: HTTP Basic Authentication enabled.
 774
 775 3. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.
 776 SAML Responder returns SAML Response.
 777 SAML Responder CONFIRM: SAML Requester sent Authentication Query.
 778 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
 779
 780 4. SAML Requester sets AC comparison to “better”.
 781 SAML Requester CONFIRM: AC comparison="better".

782
783 5. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.
784 SAML Responder returns SAML Response.
785 SAML Responder CONFIRM: SAML Requester sent Authentication Query.
786 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
787
788 6. SAML Requester sets AC comparison to “minimum”.
789 SAML Requester CONFIRM: AC comparison=”minimum”.
790
791 7. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.
792 SAML Responder returns SAML Response.
793 SAML Responder CONFIRM: SAML Requester sent Authentication Query.
794 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
795
796 8. SAML Requester sets AC comparison to “maximum”.
797 SAML Requester CONFIRM: AC comparison=” maximum”.
798
799 9. SAML Requester sends Authentication Query to SAML Responder through SOAP binding.
800 SAML Responder returns SAML Response.
801 SAML Responder CONFIRM: SAML Requester sent Authentication Query.
802 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
803
804

805 **Test Case K – SAML Attribute Authority**
 806 **Conformance Modes: SAML Attribute Authority**
 807 **Preconditions: Metadata exchanged and loaded**
 808

809 **Test Step Overview**
 810

Steps	Action/Message/Setting
1	Web SSO / POST (signed) / Persistent
2	Attribute Query No Attributes
3	Attribute Query / SOAP
4	Attribute Query Attribute Named
5	Attribute Query / SOAP
6	Attribute Query Attribute Value
7	Attribute Query / SOAP
8	Encrypted Attribute / Attribute Query Attribute Named
9	Attribute Query / SOAP

811
 812 **Test Step Detail**

- 813 1. User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for
 814 the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User
 815 and IdP returns a signed SAML Response message through HTTP POST binding.
 816 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
 817 HTTP POST binding.
 818 IdP CONFIRM: User has been federated
 819 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
 820
 821 2. SAML Responder sets attribute query to no attributes.
 822 SAML Responder CONFIRM: Attribute Query No Attributes.
 823
 824 3. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML
 825 Responder returns SAML Response.
 826 SAML Responder CONFIRM: SAML Requester sent Attribute Query.
 827 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
 828
 829 4. SAML Responder sets attribute query to attribute named.
 830 SAML Responder CONFIRM: Attribute Query Attribute Named.
 831
 832 5. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML
 833 Responder returns SAML Response.
 834 SAML Responder CONFIRM: SAML Requester sent Attribute Query.
 835 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

836
837 6. SAML Responder sets attribute query to attribute value.
838 SAML Responder CONFIRM: Attribute Query Attribute Value.
839
840 7. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML
841 Responder returns SAML Response.
842 SAML Responder CONFIRM: SAML Requester sent Attribute Query.
843 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
844
845 8. SAML Responder sets attribute query to attribute named. SAML Responder enables attribute for
846 encryption.
847 SAML Responder CONFIRM: Attribute Query Attribute Named.
848 SAML Responder CONFIRM: Encryption assertion enabled.
849
850 9. SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML
851 Responder returns SAML Response.
852 SAML Responder CONFIRM: SAML Requester sent Attribute Query.
853 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
854

855 **Test Case L – SAML Authorization Decision Authority**
 856 **Conformance Modes: SAML Authorization Decision Authority**
 857 **Preconditions: Metadata exchanged and loaded**
 858

859 **Test Step Overview**

Steps	Action/Message/Setting
1	Web SSO / POST (signed) / Persistent
2	HTTP Basic Authentication
3	AuthzQuery Resource=never (never permitted)
4	Authorization Decision Query / SOAP
5	AuthzQuery Resource=maybe (permitted if auth match)
6	Authorization Decision Query / SOAP
7	AuthzQuery Resource=always (always permitted)
8	Authorization Decision Query / SOAP

860

861 **Test Step Detail**

862 1. User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for
 863 the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User
 864 and IdP returns a signed SAML Response message through HTTP POST binding.

865 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
 866 HTTP POST binding.

867 IdP CONFIRM: User has been federated

868 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

869

870 2. SAML Requester enables HTTP Basic Authentication.

871 SAML Requester CONFIRM: HTTP Basic Authentication enabled.

872

873 3. SAML Responder sets Authorization Query to never permitted which means subject is never
 874 authorized for access.

875 SAML Responder CONFIRM: AuthzQuery Resource=never

876

877 4. SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML
 878 Responder returns SAML Response.

879 SAML Responder CONFIRM: SAML Requester sent Authorization Query.

880 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

881

882 5. SAML Responder sets authorization query to maybe permitted if authentication is matched which
 883 means subject is authorized if it is a “particular” subject.

884 SAML Responder CONFIRM: AuthzQuery Resource=maybe

885

886 6. SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML
887 Responder returns SAML Response.
888 SAML Responder CONFIRM: SAML Requester sent Authorization Query.
889 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
890
891 7. SAML Responder sets Authorization Query to always permitted which means subject is always
892 authorized.
893 SAML Responder CONFIRM: AuthzQuery Resource=always
894
895 8. SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML
896 Responder returns SAML Response.
897 SAML Responder CONFIRM: SAML Requester sent Authorization Query.
898 SAML Requester CONFIRM: SAML Responder returned the SAML Response.
899

900 **Test Case M – SAML URI Binding**

901 **Conformance Modes: SAML Attribute Authority, SAML Authorization Decision**
902 **Authority, SAML Authentication Authority, SAML Requester**

903 **Preconditions: Metadata exchanged and loaded**

904

905 **Test Step Overview**

Steps	Action/Message/Setting
1	HTTP Basic Authentication
2	Request for Assertion by Identifier / SOAP
3	HTTP Basic Authentication
4	SAML URI Binding

906

907 **Test Step Detail**

908 1. Requester enables HTTP Basic Authentication

909 Requester CONFIRM: HTTP Basic Authentication enabled.

910

911 2. Requester sends SAML Request message to Responder in SOAP over HTTP. Request message
912 contains <AssertionIDRequest> and HTTP basic authentication. Assertion ID is assigned at test
913 time. Responder returns identified <Assertion> in SAML Response in SOAP over HTTP.

914 Requester and Responder CONFIRM: Use SOAP over HTTP

915 Requester and Responder CONFIRM: Use HTTP basic authentication.

916 Requester and Responder CONFIRM: Request message contains <AssertionIDRequest> for
917 assigned assertion ID.

918 Requester and Responder CONFIRM: Response message contains identified <Assertion>.

919

920 3. Requester enables HTTP Basic Authentication

921 Requester CONFIRM: HTTP Basic Authentication enabled.

922

923

924 4. Requester sends HTTP GET message to Responder with URI containing assertion ID. Assertion
925 ID assigned at test time. HTTP GET message contains HTTP basic authentication. Responder
926 returns HTTP response with SAML Response containing assigned <Assertion>.

927 Requester and Responder CONFIRM: Use HTTP basic authentication.

928 Requester and Responder CONFIRM: HTTP GET URI contains assigned assertion ID.

929 Requester and Responder CONFIRM: Assigned <Assertion> is returned in response.

930

931

932 **Test Case N – Error Testing**
 933 **Conformance Modes: IdP, SP, SP Lite**
 934 **Preconditions: Metadata exchanged and loaded**

935
 936 **Test Step Overview**

Steps	Action/Message/Setting
1	Artifact Refused
2	Successful Response Message
3	Repost of Assertion
4	Altered data, signature mismatch
5	Wrongkey used to sign
6	SubjectConfirmation Recipient !=assertion service consumer URL
7	Unknown SubjectConfirmationMethod
8	IncorrectAudienceRestriction != requestor
9	SubjectConfirmation NoOnOrAfter expired
10	Unknown Condition

937
 938 **Test Step Detail**

939
 940 NOTE – Test Steps 2-9 involve the Liberty Error Test Tool. Metadata for conducting these tests will
 941 be exchanged.

- 942
 943 2. Test Harness POSTs an unsolicited SAML Response message containing a valid assertion.
 944 SP CONFIRM: SAML Response was received and assertion accepted.
 945
 946 3. Test Harness re-POSTs the assertion that was successful during the initialization of this test
 947 sequence.
 948 SP CONFIRM: Assertions are not replayed within the validity period of the assertion.
 949
 950 4. The assertion of the SAML Response from Step 2 is altered and sent without re-signing in a
 951 HTTP POST from Test Harness.
 952 SP CONFIRM: SP rejects the message.
 953
 954 5. The assertion of the SAML Response from Step 2 is sent but signed with the wrong signing key in
 955 a HTTP POST from Test Harness.
 956 SP CONFIRM: SP rejects the message.
 957
 958 6. The Test Harness constructs a SAML Response message with an incorrect Recipient attribute.
 959 Recipient attribute is in the <SubjectConfirmationData> element.
 960 SP CONFIRM: SP detects and rejects the message.

- 961
962 7. The Test Harness sends an altered assertion in the SAML Response. A different Method URN is
963 substituted in the assertion's <SubjectConfirmation> element other than the required Method of
964 urn:oasis:names:tc:SAML:2.0:cm:bearer.
965 SP CONFIRM: SP detects and rejects the message.
966
- 967 8. The Test Harness POSTs a SAML Response containing an assertion which does not contain an
968 <AudienceRestriction> including the SP's unique identifier as an <Audience>.
969 SP CONFIRM: SP rejects the assertion.
970
- 971 9. The Test Harness sets the NotOnOrAfter attribute to a future value which has passed.
972 SP CONFIRM: The SP to reject the assertion because of the NotOnOrAfter attribute.
973
- 974 10. The Test Harness includes a <Condition> extension element in the <Conditions> element of the
975 assertion which cannot be understood.
976 SP CONFIRM: The SP rejects the assertion.
977
978

979 **Test Case O – GSA Profile**
 980 **Preconditions: Metadata exchanged and loaded**
 981 **Conformance Modes: IdP, SP**

982
 983 **Test Step Overview**

Steps	Action/Message/Setting
1	IdP Discovery
2	Web SSO AuthnRequest / HTTP Redirect (signed)
3	Web SSO AuthnResponse / HTTP POST (signed)
4	SLO SP-initiated / HTTP Redirect(signed)
5	IdP Discovery
6	Web SSO at IdP AuthnResponse / HTTP POST (signed)
7	SLO SP-initiated / HTTP Redirect(signed)

984
 985

986 **Pre-Test Setup**

- 987 1. TLS certificates MUST be trusted by default in commonly used browsers. Certificates and
 988 security toolkits MUST follow GSA requirements.
 989 2. Common domain name cookies must be properly created with all IdPs.

990

991 **Test Steps Details**

- 992 1. User logs into SP through TLS. User/SP does Single Sign-On. SP uses common domain cookie
 993 for IdP Discovery to find IdP.
 994 REQUIRED: The SP MUST present a tailored list of compatible IdP featuring, at a
 995 minimum, compatible IdP(s) in the CDC.
 996
 997 2. SP sends a SAML Authentication Request to the IdP through HTTP Redirect binding.
 998 Authentication Request must be signed and delivered over TLS. The following list contains either
 999 requirements for the request which must occur, conditional actions which if done must be handled in
 1000 a specific way or optional conditions which may or may not be done.

1001

<AuthnRequest>

1002

REQUIRED: <Issuer> MUST be present, MUST be the identifier of the SP and
 1003 MUST be a URL within the SP domain.

1004

CONDITIONAL: ProtocolBinding is optional but if present it MUST be
 1005 urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST.

1006

CONDITIONAL: <RequestedAuthnContext> MAY be included in authentication
 1007 request but if present, the Comparison attribute MUST be set to minimum.

1008

OPTIONAL: IsPassive MAY be used.

1009

OPTIONAL: Within <NameIDPolicy>, AllowCreate attribute MAY be present.

1010

CONDITIONAL: SP MAY set ForceAuthn to true. If ForceAuthn is set to TRUE,
 1011 IsPassive MUST either be omitted or set to FALSE.

1011

1012 CONDITIONAL: If SP sets ForceAuthn to true, IdP MUST authenticate regardless of
1013 user's authentication session status.

1014 CONDITIONAL: Within <NameIDPolicy>, if Format is present, it MUST use one of
1015 the following:

1016 urn:oasis:names:tc:SAML:2.0:nameid-format:persistent

1017 urn:oasis:names:tc:SAML:2.0:nameid-format:transient

1018

1019

1020 3. IdP returns the Authentication Response with the assertion to the SP through HTTP POST binding
1021 over TLS to the SP's <Assertion> Consumer Service. The following list contains either requirements
1022 for the request which must occur or optional conditions which may be done.

1023 <Response>

1024 REQUIRED: Version attribute MUST be set to "2.0".

1025 REQUIRED: Each <Response> MUST contain no more than one

1026 <EncryptedAssertion>.

1027 CONDITIONAL: Consent attribute MAY be used, but if used MUST use one of the
1028 following:

1029 urn:oasis:names:tc:SAML:2.0:consent:obtained

1030 urn:oasis:names:tc:SAML:2.0:consent:prior

1031 urn:oasis:names:tc:SAML:2.0:consent:current-implicit

1032 urn:oasis:names:tc:SAML:2.0:consent:current-explicit

1033 urn:oasis:names:tc:SAML:2.0:consent:unspecified

1034

1035 <Assertion> element within the <Response>

1036 REQUIRED: An <Assertion> MUST be returned within <EncryptedAssertion> by
1037 the IdP and MUST be signed and encrypted.

1038 REQUIRED: Version MUST be 2.0.

1039 REQUIRED: <Issuer> MUST be present, MUST be the identifier of the IdP and
1040 MUST be a URL reference within the domain of the IdP.

1041 REQUIRED: There MUST be exactly one <Subject> per <Assertion>.

1042

1043 <Subject> element within the <Assertion>

1044 REQUIRED: An <Assertion> MUST contain exactly one <Subject> indicating the
1045 end user to which <Assertion> pertains.

1046 REQUIRED: <NameID> MUST contain a Format attribute set either:

1047 urn:oasis:names:tc:SAML:2.0:nameid-format:persistent

1048 urn:oasis:names:tc:SAML:2.0:nameid-format:transient

1049

1050 <AuthnStatement> element within the <Assertion>

1051 REQUIRED: <AuthnStatement> MUST include the SessionIndex of the end user.

1052 OPTIONAL: <AuthnStatement> MAY contain SessionNotOnOrAfter but SP is NOT
1053 REQUIRED to honor SessionNotOnOrAfter.

1054

1055 <AttributeStatement> element within the <Assertion>

1056 REQUIRED: The first transmission of an <Assertion> MUST contain exactly one
1057 <AttributeStatement> for a particular <Subject>. Each subsequent <Assertion>
1058 MUST contain no more than one <AttributeStatement>.

1059 REQUIRED: Each <Attribute MUST not be encrypted.

1060 CONDITIONAL: If present, NameFormat of <Attribute> MUST be one of the
1061 following:

 urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified

1062 urn:oasis:names:tc:SAML:2.0: attrname-format:uri

1063 urn:oasis:names:tc:SAML:2.0: attrname-format:basic

1064 REQUIRED: <Attribute> Name MUST be a URI.

1065 CONDITIONAL: Where the definition of an attribute includes one or more
1066 descriptors for the attribute, FriendlyName, if present, MUST be one of the defined
1067 descriptors.
1068

1069
1070 CONDITIONAL: If <AuthnStatement> accompanies <AttributeStatement>, the following
1071 attributes MUST be present and follow the specified format

- 1072 • us:gov:e-authentication:basic:assuranceLevel MUST be one of 1, 2, 3, 4, or test and use
1073 datatype of xs:string.
- 1074 • urn:oid:2.5.4.3 MUST have first name followed by optional middle name or initial
1075 followed by last name delimited by spaces and MUST be <=256 characters in length and
1076 use datatype of xs:string.
- 1077 • us:gov:e-authentication:basic:specVer MUST be “2.0” for this interface specification and
1078 use datatype of xs:string.:

1079
1080 CONDITIONAL: These attributes are optional but may provide a richer attribute set for end
1081 user.

- 1082 • urn:oid:2.5.4.4 MUST be <=128 characters in length and of datatype xs:string.
- 1083 • urn:oid:2.5.4.42 MUST be <=128 characters in length and of datatype xs:string.
- 1084 • urn:oid:1.3.6.1.4.1.1466.101.120.34 MUST be <=128 characters in length and of
1085 datatype xs:string.
- 1086 • urn:oid: 2.5.4.44 MUST be <=20 characters in length and of datatype xs:string.
- 1087 • us:gov:e-authentication:basic:PSSN MUST be 4 digits and of datatype xs:integer.
- 1088 • us:gov:e-authentication:basic:birthMonth MUST be 2 digits and MUST contain a value
1089 in the range of 01 - 12 and of datatype xs:integer.
- 1090 • us:gov:e-authentication:basic:birthDay MUST be 2 digits and MUST contain a value in
1091 the range of 01 – 31 and of datatype xs:integer.
- 1092 • us:gov:e-authentication:basic:birthYear MUST be 4 digits (yyyy) and of datatype
1093 xs:integer.
- 1094 • us:gov:e-authentication:basic:address1 MUST be <= 50 characters in length and of
1095 datatype xs:string.
- 1096 • us:gov:e-authentication:basic:address2 MUST be <= 50 characters in length and of
1097 datatype xs:string.
- 1098 • urn:oid:2.5.4.7 MUST be <= 28 characters in length and of datatype xs:string.
- 1099 • urn:oid:2.5.4.8 MUST be 2 character length state code and of datatype xs:string.

1100 • urn:oid:2.5.4.17 MUST be either 5 digit format or 5digit-4digit (including the dash)
1101 format and of datatype xs:string .
1102 • us:gov:e-authentication:basic:Sid MUST be <= 128 characters in length and of datatype
1103 xs:string.
1104
1105 4. SP sends a signed LogoutRequest message to IdP over TLS 1.0 using HTTP Redirect binding. IdP
1106 logs out User session. IdP returns a signed LogoutResponse message over TSL 1.0 using the HTTP
1107 Redirect binding.
1108 <Logout Request>
1109 REQUIRED: Version attribute MUST be set to “2.0”.
1110 <Logout Response>
1111 REQUIRED: Version attribute MUST be set to “2.0”.
1112
1113 5. SP and IdP configure Common Domain Cookie for IdP Discovery. Domain name will be provided
1114 at time of test.
1115 REQUIRED: SP and IdP have proper cookie for IdP Discovery.
1116
1117 6. User logs in at IdP does Single Sign-On. IdP sends an unsolicited AuthnResponse to SP through
1118 HTTP POST binding over TLS to the SP’s <Assertion> Consumer Service. The following list
1119 contains either requirements for the request which must occur or optional conditions which may be
1120 done.
1121 <Response>
1122 REQUIRED: Version attribute MUST be set to “2.0”.
1123 REQUIRED: Each <Response> MUST contain no more than one
1124 <EncryptedAssertion>.
1125 CONDITIONAL: Consent attribute MAY be used, but if used MUST use one of the
1126 following:
1127 urn:oasis:names:tc:SAML:2.0:consent:obtained
1128 urn:oasis:names:tc:SAML:2.0:consent:prior
1129 urn:oasis:names:tc:SAML:2.0:consent:current-implicit
1130 urn:oasis:names:tc:SAML:2.0:consent:current-explicit
1131 urn:oasis:names:tc:SAML:2.0:consent:unspecified
1132
1133 <Assertion> element within the <Response>
1134 REQUIRED: An <Assertion> MUST be returned within <EncryptedAssertion> by
1135 the IdP and MUST be signed and encrypted.
1136 REQUIRED: Version MUST be 2.0.
1137 REQUIRED: <Issuer> MUST be present, MUST be the identifier of the IdP and
1138 MUST be a URL reference within the domain of the IdP.
1139 REQUIRED: There MUST be exactly one <Subject> per <Assertion>.
1140
1141 <Subject> element within the <Assertion>
1142 REQUIRED: An <Assertion> MUST contain exactly one <Subject> indicating the
1143 end user to which <Assertion> pertains.
1144 REQUIRED: <NameID> MUST contain a Format attribute set either:

1145 urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
1146 urn:oasis:names:tc:SAML:2.0:nameid-format:transient

1147
1148 <AuthnStatement> element within the <Assertion>
1149 REQUIRED: <AuthnStatement> MUST include the SessionIndex of the end user.
1150 OPTIONAL: <AuthnStatement> MAY contain SessionNotOnOrAfter but SP is NOT
1151 REQUIRED to honor SessionNotOnOrAfter.

1152
1153 <AttributeStatement> element within the <Assertion>
1154 REQUIRED: The first transmission of an <Assertion> MUST contain exactly one
1155 <AttributeStatement> for a particular <Subject>. Each subsequent <Assertion>
1156 MUST contain no more than one <AttributeStatement>.
1157 REQUIRED: Each <Attribute MUST not be encrypted.
1158 CONDITIONAL: If present, NameFormat of <Attribute> MUST be one of the
1159 following:
1160 urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified
1161 urn:oasis:names:tc:SAML:2.0: attrname-format:uri
1162 urn:oasis:names:tc:SAML:2.0: attrname-format:basic
1163 REQUIRED: <Attribute> Name MUST be a URI.
1164 CONDITIONAL: Where the definition of an attribute includes one or more
1165 descriptors for the attribute, FriendlyName, if present, MUST be one of the defined
1166 descriptors.

1167
1168 CONDITIONAL: If <AuthnStatement> accompanies <AttributeStatement>, the following
1169 attributes MUST be present and follow the specified format
1170 • us:gov:e-authentication:basic:assuranceLevel MUST be one of 1, 2, 3, 4, or test and use
1171 datatype of xs:string.
1172 • urn:oid:2.5.4.3 MUST have first name followed by optional middle name or initial
1173 followed by last name delimited by spaces and MUST be <=256 characters in length and
1174 use datatype of xs:string.
1175 • us:gov:e-authentication:basic:specVer MUST be “2.0” for this interface specification and
1176 use datatype of xs:string.:

1177
1178 CONDITIONAL: These attributes are optional but may provide a richer attribute set for end
1179 user.
1180 • urn:oid:2.5.4.4 MUST be <=128 characters in length and of datatype xs:string.
1181 • urn:oid:2.5.4.42 MUST be <=128 characters in length and of datatype xs:string.
1182 • urn:oid:1.3.6.1.4.1.1466.101.120.34 MUST be <=128 characters in length and of
1183 datatype xs:string.
1184 • urn:oid: 2.5.4.44 MUST be <=20 characters in length and of datatype xs:string.
1185 • us:gov:e-authentication:basic:PSSN MUST be 4 digits and of datatype xs:integer.
1186 • us:gov:e-authentication:basic:birthMonth MUST be 2 digits and MUST contain a value
1187 in the range of 01 - 12 and of datatype xs:integer.

- 1188 • us:gov:e-authentication:basic:birthDay MUST be 2 digits and MUST contain a value in
1189 the range of 01 – 31 and of datatype xs:integer.
- 1190 • us:gov:e-authentication:basic:birthYear MUST be 4 digits (yyyy) and of datatype
1191 xs:integer.
- 1192 • us:gov:e-authentication:basic:address1 MUST be <= 50 characters in length and of
1193 datatype xs:string.
- 1194 • us:gov:e-authentication:basic:address2 MUST be <= 50 characters in length and of
1195 datatype xs:string.
- 1196 • urn:oid:2.5.4.7 MUST be <= 28 characters in length and of datatype xs:string.
- 1197 • urn:oid:2.5.4.8 MUST be 2 character length state code and of datatype xs:string.
- 1198 • urn:oid:2.5.4.17 MUST be either 5 digit format or 5digit-4digit (including the dash)
1199 format and of datatype xs:string .
- 1200 • us:gov:e-authentication:basic:Sid MUST be <= 128 characters in length and of datatype
1201 xs:string.
- 1202

1203 7. IdP logs out User session. IdP sends a signed LogoutRequest message to SP over TLS 1.0 using
1204 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message
1205 over TSL 1.0 using the HTTP Redirect binding.

1206 <Logout Request>

1207 REQUIRED: Version attribute MUST be set to “2.0”.

1208 <Logout Response>

1209 REQUIRED: Version attribute MUST be set to “2.0”.

1210
1211
1212
1213

1214 References

- 1215 [SAMLTP2] Eric Tiffany et al, “SAML 2.0 Interoperability Testing Procedures, V2.0,”
1216 Liberty Alliance Project (July 2006),
1217 [http://www.projectliberty.org/liberty/content/download/952/6702/file/LAP-](http://www.projectliberty.org/liberty/content/download/952/6702/file/LAP-SAML-TP-Rev2.0-Final_7192006165451.pdf)
1218 [SAML-TP-Rev2.0-Final_7192006165451.pdf](http://www.projectliberty.org/liberty/content/download/952/6702/file/LAP-SAML-TP-Rev2.0-Final_7192006165451.pdf)
- 1219 [ExcXMLCan] John Boyer et al, “Exclusive XML Canonicalization Version 1.0, W3C
1220 Recommendation”, W3C (July 2002), <http://www.w3.org/TR/xml-exc-c14n/>
- 1221 [SAMLAuthnCxt] J. Kemp et al, “Authentication Context for the OASIS Security Assertion
1222 Markup Language (SAML) V2.0,” OASIS SSTC (March 2005), [http://](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf)
1223 [docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf) [open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf).
- 1224 [SAMLBind] Scott Cantor et al, “Bindings for the OASIS Security Assertion Markup
1225 Language (SAML) V2.0,” OASIS SSTC (March 2005), [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
1226 [open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 1227 [SAMLConf] Prateek Mishra et al, “Conformance Requirements for the OASIS Security
1228 Assertion Markup Language (SAML) V2.0,” OASIS SSTC (March 2005).
1229 <http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>.
- 1230 [SAMLCore] S. Cantor et al, “Assertions and Protocols for the OASIS Security Assertion
1231 Markup Language (SAML) V2.0,” OASIS SSTC (March 2005),
1232 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.
- 1233 [SAMLErrata] Jahan Moreh, “Errata for the OASIS Security 2 Assertion Markup Language
1234 (SAML) V2.0, Working Draft 28,” OASIS SSTC (May 8, 2006),
1235 [http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-](http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-2.0-draft-28.pdf)
1236 [2.0-draft-28.pdf](http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-2.0-draft-28.pdf)
- 1237 [SAMLLDAP] S. Cantor et al, “SAML V2.0 X.500/LDAP Attribute Profile,” OASIS SSTC
1238 (December 19, 2006), [http://docs.oasis-open.org/security/saml/SpecDrafts-](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-attribute-x500-cd-01.pdf)
1239 [Post2.0/sstc-saml-attribute-x500-cd-01.pdf](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-attribute-x500-cd-01.pdf)
- 1240 [SAMLMeta] S. Cantor et al, “Metadata for the OASIS Security Assertion Markup
1241 Language (SAML) V2.0,” OASIS SSTC (March 2005), [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
1242 [open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf).
- 1243 [SAMLMetaExt] Tom Scavo et al, “SAML Metadata Extension for Query Requesters,
1244 Committee Draft 01”, OASIS SSTC (March 2006), [http://www.oasis-](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
1245 [open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
1246 [01.pdf](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
- 1247 [SAMLProf] S. Cantor et al, “Profiles for the OASIS Security Assertion Markup Language
1248 (SAML) V2.0,” OASIS SSTC (March 2005), [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
1249 [open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf).
- 1250 [SAMLSec] Frederick Hirsch et al, “Security and Privacy Considerations for the OASIS
1251 Security Assertion Markup Language (SAML) V2.0,” OASIS SSTC (March

1252 2005), [http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)
1253 [os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)

1254 [GSATechAppr] Dave Silver et al, “Technical Approach for the Authentication Service
1255 Component” vs. 2.0.0 GSA (May 2007),
1256 <http://www.cio.gov/eauthentication/TechnicalArchitecture.htm>

1257 [GSAAdoptSchm] Dave Silver et al, “E-Authentication Federation Adopted Schemes” vs. 1.0.0
1258 GSA (May 2007),
1259 <http://www.cio.gov/eauthentication/TechnicalArchitecture.htm>

1260

1261 [GSAInterface] Dave Silver et al, “E-Authentication Federation Architecture 2.0 Interface
1262 Specifications” vs. 1.0.0 GSA (May 2007),
1263 <http://www.cio.gov/eauthentication/TechnicalArchitecture.htm>

1264

1265

1266

1267 About Drummond Group

1268 Drummond Group Inc. (DGI) is an independent, privately held company that works with software
1269 vendors, vertical industries and the standards community to drive adoption for standards by
1270 conducting interoperability and conformance testing, publishing related strategic research and
1271 developing vertical industry strategies. Founded in 1999, DGI represents best-of-breed in the
1272 industry on linking horizontal infrastructure technologies, standards and interoperability issues with
1273 the needs of vertical industries such as retail, grocery, health care, transportation, government and
1274 automotive. For more information, please visit www.drummondgroup.com or email:
1275 info@drummondgroup.com.