



Liberty ID-WSF 2.0 Errata v1.0

Version: 1.0

Editors:

Darryl Champagne, IEEE-ISTO
Rob Lockhart, IEEE-ISTO
Eric Tiffany, IEEE-ISTO

Contributors:

Frederick Hirsch, Nokia
Paul Madsen, NTT
Greg Whitehead, Hewlett Packard
Conor Cahill, Intel Corporation
Carolina Canales-Valenzuela, Ericsson

Abstract:

This document contains errata items pertaining to the Liberty ID-WSF 2.0 specification set.

Filename: liberty-idwsf-2.0-errata-v1.0.pdf

Notice

1
2 This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the
3 document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works
4 of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact
5 the Liberty Alliance to determine whether an appropriate license for such use is available.

6 Implementation of certain elements of this document may require licenses under third party intellectual property
7 rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are
8 not and shall not be held responsible in any manner for identifying or failing to identify any or all such third party
9 intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance
10 makes any warranty of any kind, express or implied, including any implied warranties of merchantability,
11 non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementers
12 of this Specification are advised to review the Liberty Alliance Project's website (<http://www.projectliberty.org>) for
13 information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance
14 Management Board.

15 Copyright © 2007 2FA Technology; Adobe Systems; Agencia Catalana De Certificacio; America Online, Inc.;
16 American Express Company; Amssoft Systems Pvt Ltd.; Avatier Corporation; BIPAC; BMC Software, Inc.; Bank of
17 America Corporation; Beta Systems Software AG; British Telecommunications plc; Computer Associates
18 International, Inc.; Credentica; DataPower Technology, Inc.; Deutsche Telekom AG, T-Com; Diamelle Technologies,
19 Inc.; Diversinet Corp.; Drummond Group Inc.; Enosis Group LLC; Entrust, Inc.; Epok, Inc.; Ericsson; Falkin
20 Systems LLC; Fidelity Investments; Forum Systems, Inc.; France Télécom; French Government Agence pour le
21 développement de l'administration électronique (ADAE); Fugen Solutions, Inc; Fulvens Ltd.; GSA Office of
22 Governmentwide Policy; Gamefederation; Gemalto; General Motors; GeoFederation; Giesecke & Devrient GmbH;
23 Hewlett-Packard Company; Hochhauser & Co., LLC; IBM Corporation; Intel Corporation; Intuit Inc.; Kantega;
24 Kayak Interactive; Livo Technologies; Luminance Consulting Services; MasterCard International; MedCommons
25 Inc.; Mobile Telephone Networks (Pty) Ltd; NEC Corporation; NTT DoCoMo, Inc.; Netegrity, Inc.; Neustar, Inc.;
26 New Zealand Government State Services Commission; Nippon Telegraph and Telephone Corporation; Nokia
27 Corporation; Novell, Inc.; OpenNetwork; Oracle Corporation; Ping Identity Corporation; RSA Security Inc.;
28 Reactivity Inc.; Royal Mail Group plc; SAP AG; Senforce; Sharp Laboratories of America; Sigaba; SmartTrust; Sony
29 Corporation; Sun Microsystems, Inc.; Supremacy Financial Corporation; Symlabs, Inc.; Telecom Italia S.p.A.;
30 Telefónica Móviles, S.A.; Telenor R&D; Thales e-Security; Trusted Network Technologies; UNINETT AS; UTI;
31 VeriSign, Inc.; Vodafone Group Plc.; Wave Systems Corp. All rights reserved.

32 Liberty Alliance Project
33 Licensing Administrator
34 c/o IEEE-ISTO
35 445 Hoes Lane
36 Piscataway, NJ 08855-1331, USA
37 info@projectliberty.org

Contents

38		
39	1. Introduction	6
40	1.1. Template Errata Item	6
41	1.1.1. Summary	6
42	1.1.2. Resolution	6
43	2. Errata	7
44	2.1. People Service Status element formatting	7
45	2.1.1. Summary	7
46	2.1.2. Resolution	7
47	2.2. People Service Occurrence of AddKnownEntity message	7
48	2.2.1. Summary	7
49	2.2.2. Resolution	7
50	2.3. References to InvocationId	7
51	2.3.1. Summary	7
52	2.3.2. Resolution	7
53	2.4. Identity Mapping Issues	8
54	2.4.1. Summary	8
55	2.4.2. Resolution	8
56	2.5. Namespace prefix definition missing	11
57	2.5.1. Summary	11
58	2.5.2. Resolution	11
59	2.6. Format of Identity Token ("Type")	11
60	2.6.1. Summary	11
61	2.6.2. Resolution	11
62	2.7. Disco Should Refer to <ds:SvcMDRegisterResponse> Instead of <ds:ModifyResponse>	11
63	2.7.1. Summary	11
64	2.7.2. Resolution	12
65	2.8. Client Profiles: References to ServiceInstanceEPR	12
66	2.8.1. Summary	12
67	2.8.2. Resolution	12
68	2.9. Client Profiles refers to 'providerID'	12
69	2.9.1. Summary	12
70	2.9.2. Resolution	12
71	2.10. SCR ID-WSF2.0, Line 198	12
72	2.10.1. Summary	12
73	2.10.2. Resolution	12
74	2.11. Value of soapAction (HTTP Header) attributes in the WSDL	12
75	2.11.1. Summary	12
76	2.11.2. Resolution	13
77	2.12. Update Text about ReplyTo	13
78	2.12.1. Summary	13
79	2.12.2. Resolution	13
80	2.13. Update Note about wsa:To header block usage	14
81	2.13.1. Summary	14
82	2.13.2. Resolution	14
83	2.14. wsa:Action for Faults, Header faults	14
84	2.14.1. Summary	14
85	2.14.2. Resolution	14
86	2.15. People Service vague on processing of circularity in collections	15
87	2.15.1. Summary	15
88	2.15.2. Resolution	15
89	2.16. People Service vague on Count and Offset Processing	15
90	2.16.1. Summary	15

91	2.16.2. Resolution	15
92	2.17. People Service vague of Count and Offset attributes on subscription processing	15
93	2.17.1. Summary	16
94	2.17.2. Resolution	16
95	2.18. People Service Occurrence Rule too Restrictive for <Tag> element	16
96	2.18.1. Summary	16
97	2.18.2. Resolution	16
98	2.19. People Service vague on processing of Tag elements	16
99	2.19.1. Summary	16
100	2.19.2. Resolution	16
101	2.20. People Service Vague on ObjectID Management	17
102	2.20.1. Summary	17
103	2.20.2. Resolution	17
104	2.21. BugZ 816: Remove Two References to ProxyTransitedPath	17
105	2.21.1. Summary	17
106	2.21.2. Resolution	17
107	2.22. People Service Uniqueness Requirements of ObjectID element	18
108	2.22.1. Summary	18
109	2.22.2. Resolution	18
110	2.23. People Service Vague on Count and Structure Attribute	18
111	2.23.1. Summary	18
112	2.23.2. Resolution	18
113	2.24. People Service Vague on Token Notification processing	18
114	2.24.1. Summary	18
115	2.24.2. Resolution	19
116	2.25. People Service vague on relationship between Subscription support and SCR	19
117	2.25.1. Summary	19
118	2.25.2. Resolution	19
119	2.26. People Service vague on CreatePSObject in AddEntityRequest	19
120	2.26.1. Summary	19
121	2.26.2. Resolution	19
122	2.27. People Service vague AddEntityRequest Processing Rules	20
123	2.27.1. Summary	20
124	2.27.2. Resolution	20
125	2.28. Adapt examples to TokenPolicy changes	20
126	2.28.1. Summary	20
127	2.28.2. Resolution	21
128	2.29. Namespace on XPath expression in Filter element	21
129	2.29.1. Summary	21
130	2.29.2. Resolution	21
131	2.30. People Service vague on XPath Filter application	22
132	2.30.1. Summary	22
133	2.30.2. Resolution	22
134	2.31. People Service vague on processing of Subscriptions for QueryObjectsRequest	22
135	2.31.1. Summary	22
136	2.31.2. Resolution	22
137	2.32. People Service vague on processing of Subscription for a ListMembersRequest	22
138	2.32.1. Summary	22
139	2.32.2. Resolution	22
140	2.33. People Service vague on communication of known identifier	23
141	2.33.1. Summary	23
142	2.33.2. Resolution	23
143	2.34. People Service vague on Subscription processing for ListMembersRequest with structured attribute not equal to 'children'.	23
144		

145	2.34.1. Summary	23
146	2.34.2. Resolution	23
147	2.35. TestMembershipRequest requires TargetObjectID	24
148	2.35.1. Summary	24
149	2.35.2. Resolution	24
150	2.36. People Service vague on processing of unknown Objects for ResolveIdentifierRequest	24
151	2.36.1. Summary	24
152	2.36.2. Resolution	24
153	2.37. ResolveIdentifierRequest with no TargetObjectID	25
154	2.37.1. Summary	25
155	2.37.2. Resolution	25
156	2.38. Optionality of Locale attribute on DisplayName	25
157	2.38.1. Summary	25
158	2.38.2. Resolution	25
159	2.39. People Service vague on Aggregation support in Subscriptions	25
160	2.39.1. Summary	25
161	2.39.2. Resolution	25
162	2.40. People Service vague on expiration for Subscriptions to AddEntityRequests	25
163	2.40.1. Summary	26
164	2.40.2. Resolution	26
165	2.41. People Service vague on use of starts attribute for Subscriptions to AddEntityRequest	26
166	2.41.1. Summary	26
167	2.41.2. Resolution	26
168	2.42. People Service vague on includeData for a Subscription to AddEntityRequest	26
169	2.42.1. Summary	26
170	2.42.2. Resolution	26
171	2.43. Schema for ItemData in Notification	26
172	2.43.1. Summary	26
173	2.43.2. Resolution	27
174	2.44. MUST vs MAY for second level status codes	27
175	2.44.1. Summary	27
176	2.44.2. Resolution	27
177	2.45. People Service vague on dealing with existing Objects in AddToCollectionRequest	27
178	2.45.1. Summary	27
179	2.45.2. Resolution	27
180	2.46. Discovery Service EPR Profile notOnOrAfter attribute name	28
181	2.46.1. Summary	28
182	2.46.2. Resolution	28
183	2.47. Inconsistent Capitalization of InteractIfNeeded attribute	28
184	2.47.1. Summary	28
185	2.47.2. Resolution	28
186	References	29

1. Introduction

The ID-WSF v2.0 protocols, as initially specified, contained certain material errors, collectively referred to as *errata*. This document describes the errata in the Liberty ID-WSF v2.0 specification set, and the approved corrections. *This normative document, in combination with the original ID-WSF 2.0 specifications, supersedes the original specification set.* The specifications targeted by this errata document are listed in [References](#), below.

Readers of the Liberty ID-WSF v2.0 specification set should note the errata in this document and incorporate it into their reading of the specifications. To assist in this process, "red-line" versions of the affected specifications are available. Note that these "red-line" documents are only informative; in all cases the normative corrections in this document take precedence.

Additionally, implementers of the affected specifications should use the Liberty schemata and wsdl associated with the specifications listed below in place of those affected by the specified errata.

- liberty-idwsf-authn-svc-2.0-errata-v1.0.pdf
- liberty-idwsf-client-profiles-2.0-errata-v1.0.pdf
- liberty-idwsf-disco-svc-2.0-errata-v1.0.pdf
- liberty-idwsf-interaction-svc-2.0-errata-v1.0.pdf
- liberty-idwsf-people-service-1.0-errata-v1.0.pdf
- liberty-idwsf-security-mechanisms-core-2.0-diff-v1.0.pdf
- liberty-idwsf-security-mechanisms-saml-profile-2.0-errata-v1.0.pdf
- liberty-idwsf-soap-binding-2.0-errata-v1.0.pdf
- liberty-idwsf-2.0-scr-1.0-diff-v1.0.pdf

1.1. Template Errata Item

1.1.1. Summary

Each errata entry should have a Summary and a Resolution section. The Summary should contain a description of the issue and indicate which specifications are involved.

1.1.2. Resolution

The Resolution section should contain specific descriptions of the revisions to the text. These descriptions should include the section and line number(s) to be revised, and the revised text.

1. Multiple revisions may be indicated in an ordered list.
2. Where it improves clarity, an unordered list indicating the before and after state may be used:
 - From: The original text.
 - To: The changed text.

2. Errata

2.1. People Service Status element formatting

[[LibertyPeopleService](#)]

2.1.1. Summary

Language and formatting around the the Status element was inconsistent.

2.1.2. Resolution

Used consistent formatting and language for `<lu:Status>` at lines 1150, 1226, 1463, 1467, 1550, 1621, 1626, 1755, 1844, 1961 and 1967.

2.2. People Service Occurrence of AddKnownEntity message

[[LibertyPeopleService](#)]

2.2.1. Summary

In one location, the People Service refers to an "AddKnownEntity" message incorrectly. It should be "AddEntityRequest."

2.2.2. Resolution

Changed Line 710 to "AddEntityRequest."

2.3. References to InvocationId

[[LibertySecMech20](#)]

2.3.1. Summary

Vestigial reference to the InvocationIdentity header, along with an unneeded key usage URI, in [[LibertySecMech20](#)].

2.3.2. Resolution

Proposed resolution: Frederick's revised text for section 6.3.1:

1. The text at line 469-479 (section 6.3.1) should now read:

A token container type is defined to provide a uniform means to convey tokens, and allows a Web Services Security token to be directly contained in the container, or to be referenced from the container. A reference may be an external reference to a token or a reference to another local token container.

The token container type (`TokenType`) may be used to define elements in the ID-WSF namespace, and has also been used to define a `<Token>` element in the security mechanisms namespace. This `<sec:Token>` element may be used in a number of ID-WSF 2.0 schema definitions, such as:

- The security context container type used in the Discovery Service to profile EPRs,
- The mapping input and output types for the Identity Mapping Service, and
- The `AddKnownEntityRequestType` for the People Service.

2. The text at line 501 is changed to:

This specification defines the following URN values for the `usage` attribute (others may be defined elsewhere):

251 •urn:liberty:security:tokenusage:2006-08:TargetIdentity

252 •urn:liberty:security:tokenusage:2006-08:SecurityToken

253 These two URNs are used when the token is contained in an EPR to be used to create a SOAP header
254 by the Discovery Service. The TargetIdentity usage indicates that the token should be used to create a
255 <sb:TargetIdentity> header block. Any token with the SecurityToken usage in an EPR is placed in a
256 <wsse:Security> header block.

257 **2.4. Identity Mapping Issues**

258 [[LibertySecMech20](#)], [[LibertyAuthn](#)]

259 **2.4.1. Summary**

260 Several problems identified in Identity Mapping Service (IMS) in [[LibertyAuthn](#)] and [[LibertySecMech20](#)]

261 1. IMS

262 • Section 7.4.2.1 MappingOutput

263 • It is not clearly stated whether or not MappingOutputs are included when that particular mapping was
264 unsuccessful (especially in the case of a multiple MappingInput request.

265 • Section 7.7 - Example

266 • The response contains just an encrypted ID while we typically use SAML assertions for the identity token
267 (and the encrypted ID would be in the Subject of such an assertion). We do talk about using an assertion
268 for the identity token in section 7.5

269 • I would suggest a more complex example showing multiple (even just 2) inputs using the same token via
270 a reference with different destination providers.

271 2. Sec Mech

272 • Section 6.6.2 Token policy - this seems under specified... examples include:

273 • issueTo - no semantics about what this means... Is this just what should be in the field, or does this mean
274 that the issuer should figure out how to send the issued token to that party?

275 • type - since it's a URI we should define the URI that should be used for SAML tokens. "By Default"
276 should be removed and instead use something along the lines "If not specified,..."

277 • wantDSEPR - should this be binary and only WSF2.0 EPR? or should this be more "wantDSInfo" and
278 have values to indicate the particular DS info you want (2.0 DS EPR, 3.0 DS EPR, 1.0 DS RO). Same
279 comment re: "default" wording.

280 **2.4.2. Resolution**

281 Proposed resolution adjusts text in several places.

282 1. Alter text at line 911-923 (section 6.6.1) of [[LibertySecMech20](#)] to read as follows:

283 Different mechanisms may be used to convey an identity including the following:

284 • A SAML 2.0 assertion element (`saml2:Assertion`) as profiled in the Security Mechanisms SAML
285 profile [[LibertySecMech20SAML](#)]. This is a `saml2:Assertion`, and not a `saml2:EncryptedAssertion`,
286 `saml2:NameID`, or `saml2:EncryptedID`.

- 287 • An opaque value, for example a `saml2:EncryptedAssertion`, `saml2:NameID`, or
288 `saml2:EncryptedID`, WSS Binary Security Token, or non-SAML values.

289 Any identity token SHOULD be able to convey information needed for discovery. This is typically an endpoint
290 reference.

291 An identity token must have an attribute of type `IDType` that may be used as a target of a `ds:Reference`, e.g., an
292 `xml:id` or `wsu:Id` attribute.

293 Normative details using SAML 2 assertions are given in the Security Mechanisms SAML profile [[Liberty-
294 SecMech20SAML](#)].

295 A WSS `SecurityTokenReference` element may also be used to reference an identity token.

296 2. The description of the `<TokenPolicy>` at line 929-946 in [[LibertySecMech20](#)] is changed as follows:

297 • **validUntil** [Optional]

298 Indicates the duration for which the requestor would like the token to be valid. The responder MAY disregard
299 the value in favor of its own policies.

300 • **issueTo** [Optional]

301 Identifies the party to whom the identity token should be issued, if not otherwise apparent from the request or
302 policy content. Note that this is usually *not* the party requesting the token, but generally a WSP the requester
303 wishes to access.

304 For example, a `samlp:NameIDPolicy` element may be included in the `TokenPolicy` element, and in some
305 cases the value of the associated `SPNameQualifier` attribute will already indicate the party to whom the
306 token is being issued, making use of `issueTo` unnecessary.

307 • **type** [Optional]

308 Specifies the type of identity token to be returned upon an identity token request. If no type is specified then
309 the type of token returned is `Opaque` and need not necessarily be understood by the requestor.

310 The value of the type attribute is a URI. The following are defined in this document:

311 • **SecMech-SAML-2.0-Assertion:**

312 • This MUST be a SAML 2.0 assertion (`saml2:Assertion`) as profiled in the Security Mecha-
313 nisms SAML Profile. This is a `saml2:Assertion`, and not a `saml:EncryptedAssertion`,
314 `saml:NameID`, or `saml:EncryptedID`, which are all considered `Opaque` types.

315 • A `samlp2:NameIDPolicy` element SHOULD be included in the `TokenPolicy` element.

316 • URI value: *urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion*

317 • **Opaque:**

318 • The format is not specified and may be any format chosen by the IdP including, but not limited
319 to, SAML assertions, Encrypted Assertions, NameIDs, Encrypted NameIDs, WSS Binary Security
320 Tokens or other forms.

321 • URI value: *urn:liberty:security:2006-08::IdentityTokenType:Opaque*

322 • **wantDSEPR** [Optional]

323 Specifies whether the requestor would like the token to include a WSF 2.0 Endpoint Reference for the
324 Discovery Service in a token returned by that Discovery Service. The default value is 'true'.

325 • **Any Attribute** [Zero or More]

326 Any attribute can be used to describe other characteristics of the desired identity token. The wildcard is
327 necessary because of the arbitrary nature of identity tokens.

328 • **Any Element** [Zero or More]

329 Any element can be used to describe other characteristics of the desired identity token. The wildcard is
330 necessary because of the arbitrary nature of identity tokens.

331 3. The text at line 1263-1265 (section 7.4.2.2) of [[LibertyAuthn](#)] is changed as follows:

332 An <IdentityMappingResponse> consists of a status element and zero or more <MappingOutput> elements,
333 one for each successfully processed token request. Unsuccessfully processed <MappingInput> elements do not
334 result in a corresponding <MappingOutput> element. If multiple <MappingInput> elements were included
335 in a request, then each output element **MUST** contain a reqRef attribute matching it to the corresponding input
336 element.

337 4. The example at line 1330 (section 7.7) of [[LibertyAuthn](#)] is changed to the following:

338 The following example shows a request for a SAML identity token. The policy and input token indicate a request
339 to map from an identifier scoped to one SP into an identifier scoped to another. In this case, the input token is a
340 bare identifier (probably extracted from another SAML token).

```
341 <sa:IdentityMappingRequest>
342   <sa:MappingInput>
343     <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
344       <samlp2:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
345         SPNameQualifier="https://spb.example.com"/>
346     </sec:TokenPolicy>
347     <sec:Token>
348       <saml2:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
349         NameQualifier="https://idp.example.com" SPNameQualifier="https://spa.example.com">
350         DBC63923-C718-4249-83CE-1E53D80D8A4A
351       </saml2:NameID>
352     </sec:Token>
353   </sa:MappingInput>
354 </sa:IdentityMappingRequest>
```

356 The following is a possible response to the request above. The returned token is a signed SAML assertion with
357 an encrypted name identifier. The requester can establish the expiration from the response, giving it guidance as
358 to when the token might need renewal.

```

359
360     <sa:IdentityMappingResponse>
361         <sa:Status code="OK" />
362         <sa:MappingOutput>
363             <sec:Token>
364                 <saml2:Assertion Version="2.0" IssueInstant="2006-03-19T07:35:00Z"
365                     ID="e9ab6ff0-4ee0-4ce2-868f-18873bdc87de">
366                     <saml2:Issuer>https://idp.example.com</saml2:Issuer>
367                     <ds:Signature>...</ds:Signature>
368                     <saml2:Subject>
369                         <saml2:EncryptedID>
370                             <xenc:EncryptedData>U2XTCNvRX7B11NK182nmY00TEk==</xenc:EncryptedData>
371                         </saml2:EncryptedID>
372                     </saml2:Subject>
373                     <saml2:Conditions NotOnOrAfter="2006-03-19T08:35:00Z">
374                         <saml2:AudienceRestriction>
375                             <saml2:Audience>https://spb.example.com</saml2:Audience>
376                         </saml2:AudienceRestriction>
377                     </saml2:Conditions>
378                 </saml2:Assertion>
379             </sec:Token>
380         </sa:MappingOutput>
381     </sa:IdentityMappingResponse>

```

Example 1.

2.5. Namespace prefix definition missing

[LibertySecMech20]

2.5.1. Summary

The `saml2p` namespace prefix definition was omitted from Table 1 in [LibertySecMech20].

2.5.2. Resolution

The following row is added to Table 1 which starts at line 165 of [LibertySecMech20]:

<code>samlp2:</code>	<code>urn:oasis:names:tc:SAML:2.0:protocol</code>
	The prefix <code>samlp2:</code> stands for the SAML v2 protocol namespace. It is defined in [SAMLCore2].

2.6. Format of Identity Token ("Type")

[LibertySecMech20]

2.6.1. Summary

The `type` attribute of the `<TokenPolicyType>` element in [LibertySecMech20] refers to the type of token being requested. The text is unclear, and needs to explicitly state that there are two possible types of requested tokens (mapped as two possible URIs to be conveyed inside the `type` attribute): SAML assertion or other (opaque). The default value refers to opaque tokens

2.6.2. Resolution

Included in resolution in 2.4.2.

2.7. Disco Should Refer to `<ds:SvcMDRegisterResponse>` Instead of `<ds:ModifyResponse>`

400 [\[LibertyDisco\]](#)

401 **2.7.1. Summary**

402 Update Section 3.12 to Refer to `<ds:SvcMDRegisterResponse>` instead of `<ds:ModifyResponse>`..

403 **2.7.2. Resolution**

404 Proposed resolution entails the following changes to [\[LibertyDisco\]](#):

405 1. Change occurrences of `ModifyResponse` in lines 2135, 2144, 2150, 2164 and 4140 to
406 `SvcMDRegisterResponse`

407 2. Change the paragraph at lines 2146-2148 to read:

408 The Discovery Service instance SHOULD ONLY include the `<Keys>` element in `<SvcMDRegisterResponse>`
409 messages if has no `<ProviderID>` and the `<SvcMDRegister>` message included Service Metadata that relies
410 upon signed security tokens for one or more of its security mechanisms.

411 **2.8. Client Profiles: References to ServiceInstanceEPR**

412 [\[LibertyClientProfiles\]](#)

413 **2.8.1. Summary**

414 The Client Profiles spec refers to `ServiceInstanceEPR`, a deprecated construct

415 **2.8.2. Resolution**

416 Replaced occurrences of `disco:ServiceInstanceEPR` at Lines 78 and 84 with `wsa:EndpointReference`

417 **2.9. Client Profiles refers to 'providerID'**

418 [\[LibertyClientProfiles\]](#)

419 **2.9.1. Summary**

420 The Client Profiles spec refers to 'providerID', the case of the initial letter should be upper.

421 **2.9.2. Resolution**

422 Replaced multiple occurrences 'providerID' with 'ProviderID'.

423 **2.10. SCR ID-WSF2.0, Line 198**

424 [\[LibertyIDWSF20SCR\]](#)

425 **2.10.1. Summary**

426 In the list of security mechanisms that SP WSCs are required to support, at line 198 `urn:liberty:security:2006-`
427 `08:ClientTLS:SAMLV2` was meant to be `urn:liberty:security:2006-08:ClientTLS:peerSAMLV2` to match the require-
428 ment on SP WSPs at line 179.

429 **2.10.2. Resolution**

430 Update SCR ID-WSF2.0, line 198 to be `urn:liberty:security:2006-08:ClientTLS:peerSAMLV2`

431 **2.11. Value of soapAction (HTTP Header) attributes in the WSDL**

432 [\[LibertyAuthn\]](#), [\[LibertyPeopleService\]](#), [\[LibertyInteract\]](#), [\[LibertyDisco\]](#)

2.11.1. Summary

The value of the SOAPAction (HTTP Header) attributes in the WSDL may not match the wsaw:Action attributes.

2.11.2. Resolution

Delete the soapAction attribute lines from the WSDL and main body text in the following normative specifications and the associated informative WSDL files:

1. [[LibertyAuthn](#)]: lines 2150, 2220, and 2289

- liberty-idwsf-authn-svc-v2.0.wsdl: line 53

- liberty-idwsf-sso-svc-v2.0.wsdl: line 54

- liberty-idwsf-idmapping-svc-v2.0.wsdl: line 53

2. [[LibertyDisco](#)]: lines 3869, 4595, 4601-4602, 4609-4610, 4616-4617, 4623, 4629, 4635, and 4641

- liberty-idwsf-disco-svc-v2.0.wsdl: lines 165, 171-172, 179-180, 186-187, 193, 199, 205, and 211

3. [[LibertyInteract](#)]: line 777

- liberty-idwsf-interaction-svc-v2.0.wsdl: line 64

4. [[LibertyPeopleService](#)]: lines 3231, 3237, 3242, 3247, 3252, 3257, 3262, 3267, 3272, 3277, 3282, 3287, and 3292

- liberty-idwsf-people-service-v1.0.wsdl: lines 275, 281, 286, 291, 296, 301, 306, 311, 316, 321, 326, 331, and 336

2.12. Update Text about ReplyTo

[[LibertySOAPBinding](#)]

2.12.1. Summary

Line 590 states that "If this header block is not present, then no reply will be sent." This is wrong according to WS-A and should be deleted.

The sender processing rules, at line 750, say that the ReplyTo header MUST be included. This is not needed according to WS-A and should also be deleted. Note, the sender processing rules properly handle the case when the ReplyTo header is not present.

2.12.2. Resolution

1. Removed ReplyTo example at line 465:

```
<wsa:ReplyTo>
  <wsa:Address>...</wsa:Address>
</wsa:ReplyTo>
```

2. Removed ReplyTo reference from example at line 507:

```
<!-- reference params from FaultTo/ReplyTo EndpointReference -->
```

469 3. Updated text at line 590 to read:

470 "If the <wsa:ReplyTo> header block is not present, the value defaults to <http://www.w3.org/2005/03/addressing/role/anonymous>;
471 so, when constructing a message, the header block can be omitted if this is the value that would be used. This
472 typically allows the <wsa:ReplyTo> header block to be omitted during synchronous request-response mes-
473 sage exchanges over HTTP. Please refer to [WSAv1.0] for default processing rules in the absence of the
474 <wsa:ReplyTo> header block."

475 4. Updated text at line 596:

476 • From: "If not present, faults are sent to the address specified in the <wsa:ReplyTo> header block (if
477 present)."

478 • To: "If not present, faults are sent to the reply address."

479 5. Updated text at line 750:

480 • From: "sending a request message, the outgoing message MUST include exactly one <wsa:ReplyTo>
481 header block and at most one <wsa:FaultTo> header block (if the <wsa:FaultTo> header block is not
482 included, faults will be delivered to the <wsa:ReplyTo> endpoint)."

483 • To: "sending a request message, the outgoing message MUST include at most one <wsa:ReplyTo> header
484 block and at most one <wsa:FaultTo> header block (if the <wsa:FaultTo> header block is not included,
485 faults will be delivered to the reply endpoint)."

486 6. Removed ReplyTo example at 927:

```
487 <wsa:ReplyTo>  
488 <wsa:Address>http://www.w3.org/2005/03/addressing/role/anonymous</wsa:Address>  
489  
490 </wsa:ReplyTo>  
491  
492
```

493 2.13. Update Note about wsa:To header block usage

494 [[LibertySOAPBinding](#)]

495 2.13.1. Summary

496 Note regarding absence of <wsa:To> header block is slightly misleading. This is a non-normative note which is
497 misaligned with [WSAv1.0], and could therefore lead to confusion.

498 2.13.2. Resolution

499 1. Replace the sentence beginning at line 578 with "This..." and continuing on 579, with

500 "This typically allows the <wsa:To> header block to be omitted during synchronous request-response message
501 exchanges over HTTP. Please refer to [WSAv1.0] for default processing rules in the absence of the <wsa:To>
502 header block."

503 **2.14. wsa:Action for Faults, Header faults**

504 [[LibertySOAPBinding](#)]

505 **2.14.1. Summary**

506 Treatment of wsa:Action and wsa:Fault header for Fault messages in [[LibertySOAPBinding](#)] seems to contradict the
507 behavior defined in WS-Addressing [[WSAv1.0](#)].

508 **2.14.2. Resolution**

509 In [[LibertySOAPBinding](#)] after line 422 add the following:

510 **Note**

511 When reporting SOAP processing errors, the WS-Addressing action <http://www.w3.org/2005/08/addressing/soap/fault>
512 SHOULD be used. When reporting WS-Addressing processing errors, the WS-Addressing action
513 <http://www.w3.org/2005/08/addressing/fault> SHOULD be used. When reporting other processing errors, if no
514 specific WS-Addressing action is defined, then <http://www.w3.org/2005/08/addressing/soap/fault> SHOULD be used.

515 **2.15. People Service vague on processing of circularity in collections**

516 [[LibertyPeopleService](#)]

517 **2.15.1. Summary**

518 There is a need for clarification of the PS provider's responsibilities for dealing with circularity in group structure in
519 the [[LibertyPeopleService](#)] specification.

520 **2.15.2. Resolution**

521 Added after Line 1151

- 522 • MUST NOT allow the creation of circular collections. A circular collection is one which includes at any layer
523 of the structure below it a reference to the same collection such that a dereference of the collection would result
524 in an infinite loop. If the PS provider receives an <AddToCollectionRequest> message that would result in a
525 circular collection, it MUST respond with *Failed* as the code attribute of the top level <lu:Status> element, and
526 the code attribute of the second level <lu:Status> element MAY be set with the following status code:

- 527 • CircularCollection

528 **2.16. People Service vague on Count and Offset Processing**

529 [[LibertyPeopleService](#)]

530 **2.16.1. Summary**

531 There is a need for clarification of the PS provider's responsibilities for processing the offset and count attributes if
532 specified on a request message.

533 **2.16.2. Resolution**

534 Added before Line 1460.

- 535 • If the Count attribute is included in a <ListMembersRequest> message, the PS provider SHOULD NOT respond
536 with more objects than specified. A PS provider MAY return a smaller number of objects than specified by the
537 Count attribute.

- 538 • If the Offset attribute is included in a <ListMembersRequest> message and the PS provider returns any objects,
539 it MUST respond with a list of <Object> elements that starts with the <Object> element whose position in the
540 complete list is specified by the value of the Offset attribute.

541 **2.17. People Service vague of Count and Offset attributes on** 542 **subscription processing**

543 [[LibertyPeopleService](#)]

544 **2.17.1. Summary**

545 There is a need for clarification of the PS provider's responsibilities for processing messages that had both a
546 Subscription element and Count/Offset attributes.

547 **2.17.2. Resolution**

548 Added after Line 402

- 549 • `SubscribeToChildrenOnly`

550 Added following before Line 1449

- 551 • The WSC SHOULD NOT include a <Subscription> element in a <ListMembersRequest> message if the
552 offset attribute has any value other than '0'.

- 553 • The PS provider MUST, if the <ListMembersRequest> contains a Subscription and the Offset attribute has any
554 value other than '0', and it is otherwise capable of returning results, return those results as indicated by the Count
555 and Offset attributes, but still reject the Subscription by responding with "OKButNoSubscription" as the code
556 attribute of the top level <lu:Status> element. In this case the code attribute of the second level <lu:Status>
557 element MAY be set with the following status code:

- 558 • `NoSubscribeWithOffset`

559 **2.18. People Service Occurrence Rule too Restrictive for <Tag>** 560 **element**

561 [[LibertyPeopleService](#)]

562 **2.18.1. Summary**

563 The [[LibertyPeopleService](#)] schema is overly restrictive in specifying that there be, at most, a single <Tag> element
564 for a given <Object> element.

565 **2.18.2. Resolution**

- 566 1. Change line 242 in [[LibertyPeopleService](#)] from:

567 `<xs:element name="Tag" type="TagType" minOccurs="0"/>`

568 to read:

569 `<xs:element name="Tag" type="TagType" minOccurs="0" maxOccurs="unbounded"/>`

- 570 2. Corresponding change made in People Service schema.

571 **2.19. People Service vague on processing of Tag elements**

572 [[LibertyPeopleService](#)]

573 **2.19.1. Summary**

574 There is a need for clarification of the PS provider's responsibilities for processing the <Tag> element.

575 **2.19.2. Resolution**

576 1. Add after Line 306:

577 If they understand the tag space for a <Tag> element, WSCs and PS providers MAY process as appropriate.
578 WSCs and PS providers MAY ignore <Tag> elements.

579 **2.20. People Service Vague on ObjectID Management**

580 [[LibertyPeopleService](#)]

581 **2.20.1. Summary**

582 There is a need for clarification of the WSC and PS provider's responsibilities for dealing with to the creation of object
583 identifiers.

584 **2.20.2. Resolution**

585 Added the following to Section 2.1.4.

- 586 • The PS provider controls the creation of object identifiers. When a WSC requests the creation of an object, the
587 WSC MUST NOT provide an ObjectID in such a request message. If the request is successful, the PS provider
588 MUST return an object identifier for the new object in an ObjectID element in its response message - the WSC
589 MUST use this returned identifier in subsequent operations on that object.

590 Added the following after line 671.

- 591 • MUST include an ObjectID in the returned Object.

592 **2.21. BugZ 816: Remove Two References to ProxyTransitedPath**

593 [[LibertySecMech20](#)]

594 **2.21.1. Summary**

595 Remove two references to ProxyTransitedPath in [[LibertySecMech20](#)]:

- 596 • [[LibertySecMech20](#)] lines 1041 and 1047 - ProxyTransitedPath
- 597 • Note: The reference on [[LibertySecMech20](#)] line 1063 is fine.

598 Clarify provider chaining text to indicate that both proxying and generation of new requests is supported.

599 **2.21.2. Resolution**

- 600 1. Changed first paragraph of Section 7.3 (Provider Chaining) in [LibertySecMech20] (starts at line 998)
601 from: "Provider chaining refers to scenarios in which a service provider (WSP), upon receiving a request from
602 a sender, itself passes the request onto another service provider until the destination service provider is reached.
603 This mechanism allows proxying to be performed, where each provider proxies the request to the next party."
604 to: "Provider chaining refers to scenarios in which a service provider (WSP), upon receiving a request from a
605 sender, sends a request to the next service provider. This may be done by forwarding the request it received,
606 acting as a proxy, or by generating a new request. This may be done until the destination service provider is
607 reached."
608 2. Changed at line 1041 of [LibertySecMech20]
609 from: "The DS may have included <ProxyTransitedPath> in this token contained in the bootstrap EPR,"
610 to: "The DS may have included the <TransitedProviderPath> element in the security token contained in the
611 bootstrap EPR,"
612 3. Changed "ProxyTransitedPath" to "TransitedProviderPath" at line 1047 of [LibertySecMech20].

613 **2.22. People Service Uniqueness Requirements of ObjectID element** 614 [LibertyPeopleService]

615 **2.22.1. Summary**

616 The People Service specification is unnecessarily constraining on the uniqueness requirements of the ObjectID.

617 **2.22.2. Resolution**

618 Changed Line 229 to:

619 "The value of the <ObjectID> element uniquely identifies the <Object> within the set of all <Object> elements
620 that are accessible to a particular consumer of the People Service for the targeted identity."

621 Changed Line 267 to:

622 "The <ObjectID> element is defined so that WSCs can unambiguously refer to the parent <Object> elements."

623 Changed Line 279 to:

624 "Unique identifiers for different WSCs (e.g., pairwise identifiers), reuse of identifiers across different services, and
625 encrypted identifiers are potential mechanisms for addressing this concern."

626 **2.23. People Service Vague on Count and Structure Attribute** 627 [LibertyPeopleService]

628 **2.23.1. Summary**

629 There is a need for clarification of the PS provider's responsibilities for processing the Count and Structured attributes
630 on the ListMembersRequest message.

631 **2.23.2. Resolution**

632 Changed Line 1455 to:

- 633 • If the Structured attribute is set as *entities* the PS provider MUST return all the direct child and descendant
634 entity <Object> elements of the specified Object (subject to the restriction defined by the Count attribute if
635 present). Any collection <Object> elements MUST be removed and only entity <Object> elements returned.

636 **2.24. People Service Vague on Token Notification processing**

637 [[LibertyPeopleService](#)]

638 **2.24.1. Summary**

639 There is a need for clarification of the PS provider's processing for a Notification message by which a Token is returned
640 in response to AddEntityRequest message.

641 **2.24.2. Resolution**

642 1. Added after Line 471

643 There MUST be no <Aggregation> element present in a subscription.

644 2. Added after Line 473

645 Such a <Subscription> MUST be considered to have expired at such time as the PS provider has delivered a
646 <Token> for the invited user to the WSC and the WSC has acknowledged its receipt.

647 3. Added after Line 469

648 For a <Subscription> sent within an <AddEntityRequest, > message, the starts attribute SHOULD be
649 omitted. If present, the PS provider MAY ignore the attribute.

650 4. Added after Line 482

651 For a <Subscription> sent within an <AddEntityRequest, > message, the includeData attribute
652 SHOULD be omitted.

653 **2.25. People Service vague on relationship between Subscription 654 support and SCR**

655 [[LibertyPeopleService](#)]

656 **2.25.1. Summary**

657 There is a need for clarification of the relationship between Subscription Support and Static Conformance Require-
658 ments.

659 **2.25.2. Resolution**

660 1. Added after Line 1976 in People Service:

661 Notwithstanding this, supporting the invitation model is optional from a conformance point of view (i.e., as
662 defined by the [[LibertyIDWSF20SCR](#)]) and so any normative requirements expressed within this specification
663 should be understood in this context.

664 2. Changed Lines 218, 253, and 328 in SCR:

665 ... support management of groups & users, as described in sections 3.9-3.20 of [[LibertyPeopleService](#)].

666 **2.26. People Service vague on CreatePSObject in AddEntityRequest**

667 [[LibertyPeopleService](#)]

668 **2.26.1. Summary**

669 There is a need for clarification of PS provider's responsibilities for dealing with the CreatePSObject in an AddEntityRequest.
670

671 **2.26.2. Resolution**

672 Changed list items at lines 682 and 853

673 SHOULD, if the <AddEntityRequest> message contained a <CreatePSObject> element, attempt to create or
674 verify the existence of an object for the inviting user in the PS of the invited user (when made possible by a federated
675 identifier being established for the invited user).

676 It may be the case that the inviting user is already in the PS of the invited user as a result of a prior invitation sequence
677 initiated 'from the other side'. The PS of the inviting user MUST ensure that no duplicate object be added.

678 MAY, in order to determine whether an object for the inviting user already exists, query the members of the PS of the
679 invited user using the <ListMembersRequest> message and ask the invited user to assist in determining whether the
680 inviting user is already in the list. Other mechanisms (e.g., using a <TestMembershipRequest>) for making this
681 determination MAY alternatively be used.

682 SHOULD, if there is no existing object for the inviting user, request that an object be created with either the
683 <AddEntityRequest> or <AddKnownEntityRequest> messages.

684 SHOULD, if sending a <AddKnownEntityRequest> message for the addition, include a <sec:Token> element
685 carrying a token for the inviting user - this <sec:Token> obtained from the Identity Mapping Service of the inviting
686 user.

687 **2.27. People Service vague AddEntityRequest Processing Rules**

688 [[LibertyPeopleService](#)]

689 **2.27.1. Summary**

690 There is a need for clarification of PS provider's responsibilities for creating objects upon receiving an AddEntityRequest.
691

692 **2.27.2. Resolution**

693 Added at line 540:

694 "The Object element created by an <AddEntityRequest> message becomes a direct child of the root node."

695 Changed Line 671 to:

696 In responding to a successful *role="sgmltag"><AddEntityRequest>* message, the PS provider:

697 Added at Line 672:

- 698 • MUST create a new Object element as a direct child of the root node.

699 Changed Line 1284 to:

700 the WSC is indicating that it desires only the direct child collection and entity objects of the targeted object.

701 **2.28. Adapt examples to TokenPolicy changes**

702 [[LibertyPeopleService](#)]

703 **2.28.1. Summary**

704 Examples of IdentityMappingRequest messages in the People Service have outdated TokenPolicy structure

705 **2.28.2. Resolution**

706 1. Changed XML in Lines 2249-2259 to

```
707 <soap:Envelope>
708 <soap:Header>
709 <ws:Security>
710 <saml:Assertion ID="assertionid">
711 <credentials for Bob at IDPb
712 </saml:Assertion>
713 </ws:Security>
714 </soap:Header>
715 <soap:Body>
716 <ims:IdentityMappingRequest>
717 <ims:MappingInput>
718 <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
719 <samplp2:NameIDPolicy SPNameQualifier="https://psa.com"
720 <Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
721 </sec:TokenPolicy>
722 <sec:Token ref="#assertionid"/>
723 </ims:MappingInput>
724 </ims:IdentityMappingRequest>
725 </soap:Body>
726 </soap:Envelope>
727
728
```

729 2. Changed Line 1908 to

```
730 <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
```

731 3. Changed Line 1909 to

```
732 <samplp2:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
733 <SPNameQualifier="https://psa.com"/>
```

734 4. Changed Line 2251 to

```
735 <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
736 <samplp2:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
737 <SPNameQualifier="https://psa.com"/> </sec:TokenPolicy>
```

738 5. Changed Lines 2342 and 2352 to

```
739 <sec:TokenPolicy type="urn:liberty:security:2006-08:IdentityTokenType:SAML20Assertion">
```

740 **2.29. Namespace on XPath expression in Filter element**

741 [\[LibertyPeopleService\]](#)

742 **2.29.1. Summary**

743 In the People Service, the Filter element should carry an XPath expression specifying the matching criteria on a
744 QueryObjectsRequest. The examples of such an expression are not namespace qualified.

745 **2.29.2. Resolution**

746 Added namespace qualifiers to Xpath expressions on Lines 1678, 1683, and 1750.

747 **2.30. People Service vague on XPath Filter application**

748 [\[LibertyPeopleService\]](#)

749 **2.30.1. Summary**

750 The People Service does not clearly define the document against which a Filter's XPath expression should be matched.

751 **2.30.2. Resolution**

752 Added after Line 1675

753 The XPath expression MUST be evaluated against the set of Objects that would be returned to a hypothetical
754 ListMembersRequest that had targeted the root node and in which the Structured attribute was set to "tree"

755 **2.31. People Service vague on processing of Subscriptions for 756 QueryObjectsRequest**

757 [\[LibertyPeopleService\]](#)

758 **2.31.1. Summary**

759 There is a need for clarification of PS provider's responsibilities for processing of subscription requests.

760 **2.31.2. Resolution**

761 Added after Line 1756

- 762 • A PS provider MAY choose to reject Subscriptions within certain <QueryObjectsRequest> messages (as might
763 be desirable should the computational expense of determining when to send Notifications be prohibitive).

764 The PS provider MAY, if the <QueryObjectsRequest> contained a Subscription it does not wish to accept,
765 and it is otherwise capable of returning results, return those results but still reject the Subscription by responding
766 with "OKButNoSubscription" as the code attribute of the top level <lu:Status> element. In this case, the code
767 attribute of the second level <lu:Status> element MAY be set with the following status code:

- 768 • PolicyDoesNotAllow

769 An "OKButNoSubscription" Status code SHOULD NOT be interpreted as reflecting a PS provider's overall ability
770 to support subscriptions.

771 **2.32. People Service vague on processing of Subscription for a**
772 **ListMembersRequest**

773 [[LibertyPeopleService](#)]

774 **2.32.1. Summary**

775 There is a need for clarification of the set of objects against which the implicit trigger of 'changed' should be assessed.

776 **2.32.2. Resolution**

777 Added after Line 1258

778 For a subscription within a <ListMembersRequest> message, the PS provider MUST assess 'changes' against
779 whatever was returned in the original <ListMembersResponse>. A <Notify> MUST be sent if, were the WSC to
780 resend the same request, the results would be different than originally sent.

781 **2.33. People Service vague on communication of known identifier**

782 [[LibertyPeopleService](#)]

783 **2.33.1. Summary**

784 There is a need for clarification on how a user-supplied known identifier for an invited user should be communicated
785 to the PS Provider.

786 **2.33.2. Resolution**

787 Changed Line 829 to

788 When the token is not a identity token (as is the likely case when the known identifier is provided by the inviting user),
789 the WSC SHOULD use a SAML <saml:NameID> element within the <Token> element. If the WSC knows the
790 format of the known identifier, it SHOULD use the appropriate value for the Format attribute on the <saml:NameID>
791 element.

792 **2.34. People Service vague on Subscription processing for**
793 **ListMembersRequest with structured attribute not equal to 'children'.**

794 [[LibertyPeopleService](#)]

795 **2.34.1. Summary**

796 There is a need for clarification of PS provider's responsibilities for processing a ListMembersRequest on which the
797 structured attribute had a value other than 'children' was insufficiently defined.

798 **2.34.2. Resolution**

799 Added after Line 405

800 • `SubscribeToChildrenOnly`

801 Added after Line 1471

802 • The PS provider SHOULD, if the `<ListMembersRequest>` contains a Subscription and the Structured attribute
803 has any value other than 'children', and it is otherwise capable of returning results, return those results but
804 still reject the Subscription by responding with "OKButNoSubscription" as the code attribute of the top level
805 `<lu:Status>` element. In this case the code attribute of the second level `<lu:Status>` element MAY be set
806 with the following status code:

807 • `SubscribeToChildrenOnly`

808 An "OKButNoSubscription" Status code SHOULD NOT be interpreted as reflecting a PS provider's overall ability
809 to support subscriptions, rather simply its unwillingness to accept the particular subscription requested.

810 **2.35. TestMembershipRequest requires TargetObjectID**

811 [[LibertyPeopleService](#)]

812 **2.35.1. Summary**

813 The schema for the TargetObjectID of a TestMembershipRequest makes its occurrence optional but the message
814 requires it.

815 **2.35.2. Resolution**

816 Changed Line 1776 to:

817 `<TargetObjectID>` [**Optional**] The `<TargetObjectID>` element is used to convey the ObjectID of the target
818 group Object for which the membership of a user is being tested.

819 Absence of the `<TargetObjectID>` indicates a request to test if the identity associated with
820 the submitted token is associated to an Object (of type entity) in the PS.

821 **2.36. People Service vague on processing of unknown Objects for 822 ResolveIdentifierRequest**

823 [[LibertyPeopleService](#)]

824 **2.36.1. Summary**

825 There is a need for clarification of PS provider's responsibilities for processing a ResolveIdentifierRequest message
826 that includes ObjectIDs that are "unknown" or refer to collections.

827 **2.36.2. Resolution**

828 Added before Line 1954

829 • The PS provider MUST, if unable to find one or more (but not all) specified input objects, respond PartialSuccess as
830 the code attribute of the top level `<lu:Status>` element, and SHOULD use second level `<lu:Status>` elements
831 that containing a ref attribute equal to the associated `<ResolveInput>`'s reqID attribute. If unable to find any
832 input objects, the PS provider MUST respond Failed as the code attribute of the top level `<lu:Status>` element.
833 The second level `<lu:Status>` elements corresponding to such failed inputs MUST be set with the following
834 status code:

835 • `CannotFindObject`

836 Added after Line 1969

837 • The PS provider MUST, if one or more (but not all) specified input objects is of type "collection" , the PS provider
838 MUST respond PartialSuccess as the code attribute of the top level <lu:Status> element, and SHOULD use
839 second level <lu:Status> elements that containing a ref attribute equal to the associated <ResolveInput>'s
840 reqID attribute. If all input objects are of type "collection," the PS provider MUST respond Failed as the code
841 attribute of the top level <lu:Status> element. The second level <lu:Status> elements corresponding to such
842 failed inputs MUST be set with the following status code:

843 • ObjectIsCollection

844 **2.37. ResolveIdentifierRequest with no TargetObjectID**

845 [[LibertyPeopleService](#)]

846 **2.37.1. Summary**

847 There is a need for clarification of PS provider's responsibilities for processing a ResolveIdentifierRequest message
848 that does not include a TargetObjectID.

849 **2.37.2. Resolution**

850 Changed Line 1232 to:

851 If a WSC does not specify a TargetObjectID element in the ListMembersRequest message, the default targeted Object
852 is the root node.

853 **2.38. Optionality of Locale attribute on DisplayName**

854 [[LibertyPeopleService](#)]

855 **2.38.1. Summary**

856 There is a need for clarification on the use of the 'IsDefault' attribute on multiple DisplayName elements.

857 **2.38.2. Resolution**

858 Changed Line 290 to

859 `<xs:attribute name="Locale" type="xs:language" use="required"/>`

860 Added after Line 295

861 The <Locale> attribute specifies the language in which the display name is expressed. If not present, providers
862 SHOULD determine how to best display the name through other means.

863 The <IsDefault> attribute identifies which <DisplayName> element, if there are multiple, is default. There MUST
864 NOT be more than one <DisplayName> element with IsDefault set as "true."

865 **2.39. People Service vague on Aggregation support in Subscriptions**

866 [[LibertyPeopleService](#)]

867 **2.39.1. Summary**

868 There is a need for clarification of support for the Aggregation mechanism in Subscriptions.

869 **2.39.2. Resolution**

870 Added after Line 471

871 There MUST be no <Aggregation> element present in a subscription.

872 **2.40. People Service vague on expiration for Subscriptions to** 873 **AddEntityRequests**

874 [[LibertyPeopleService](#)]

875 **2.40.1. Summary**

876 There is a need for clarification of how Subscriptions created within AddEntityRequests should expire.

877 **2.40.2. Resolution**

878 Added after Line 473

879 Unless the value of the *expires* attribute specifies that expiration should occur earlier, for a <Subscription> sent
880 within an <AddEntityRequest> message, expiration is considered to have occurred at such time as the PS provider
881 has delivered a <Token> for the invited user to the WSC and the WSC has acknowledged its receipt.

882 **2.41. People Service vague on use of starts attribute for Subscriptions** 883 **to AddEntityRequest**

884 [[LibertyPeopleService](#)]

885 **2.41.1. Summary**

886 There is a need for clarification of whether Subscriptions created within AddEntityRequests should have a starts
887 attribute.

888 **2.41.2. Resolution**

889 Added after Line 469

890 For a <Subscription> sent within an <AddEntityRequest> message, the *starts* attribute SHOULD be omitted.
891 If present, the PS provider MAY ignore the attribute.

892 **2.42. People Service vague on includeData for a Subscription to** 893 **AddEntityRequest**

894 [[LibertyPeopleService](#)]

895 **2.42.1. Summary**

896 There is a need for clarification of whether Subscriptions created within AddEntityRequests should have an include-
897 Data attribute.

898 **2.42.2. Resolution**

899 Added after Line 482

900 For a <Subscription> sent within an <AddEntityRequest> message, the *includeData* attribute SHOULD be
901 omitted.

902 **2.43. Schema for ItemData in Notification**

903 [[LibertyPeopleService](#)]

904 **2.43.1. Summary**

905 The text and processing rules for Subscriptions and Notifications allow for the possibility of a PS provider returning
906 an empty ItemData but the schema stipulates a minOccurs="1."

907 Additionally, the ItemData schema does not allow for a Notification to include a Token, this necessary when a
908 Subscription is used within an AddEntityRequest

909 **2.43.2. Resolution**

910 ItemDataType at Line 531 and at Line 2929 changed as follows

```
911  
912 <xs:complexType name="ItemDataType">  
913     <xs:choice>  
914         <xs:element ref="Object" minOccurs="0" maxOccurs="unbounded"/>  
915         <xs:element ref="sec:Token" minOccurs="0"/>  
916     </xs:choice>  
917 </xs:complexType>  
918
```

919 **2.44. MUST vs MAY for second level status codes**

920 [[LibertyPeopleService](#)]

921 **2.44.1. Summary**

922 The People Service was overly constraining on the processing of second level status codes.

923 **2.44.2. Resolution**

924 1. Changed Line 421

925 *Failed* The value *Failed* means that the processing of the request message has failed. A second level
926 status code MAY be used to indicate the reason for the failure.

927 2. Replaced multiple occurrences of:

928 and the code attribute of the second level <lu:Status> element MUST be set with the following status code:

929 with:

930 A second level <lu:Status> MAY be inserted. If so, the code attribute of that second level <lu:Status>
931 element MUST be set with the following status code:

932 **2.45. People Service vague on dealing with existing Objects in**

933 **AddToCollectionRequest**

934 [[LibertyPeopleService](#)]

935 **2.45.1. Summary**

936 There is a need for clarification on how a PS provider should deal with an AddToCollectionRequest that asked for an
937 Object to be added to a collection to which it already belonged.

938 **2.45.2. Resolution**

939 1. Added after Line 395

940 DuplicateObject

941 2. Added after Line 1151

942 • MUST, if the `Object` specified by the value of a `<ObjectID>` element is already a member of the targeted
943 collection, respond with *Failed* as the code attribute of the top level `<lu:Status>` element. A second level
944 `<lu:Status>` MAY be inserted. If so, the code attribute of that second level `<lu:Status>` element MUST
945 be set with the following status code:

946 • DuplicateObject

947 3. Added after Line 1227

948 • MUST, if the `Object` specified by the value of a `<ObjectID>` element is not a member of the targeted
949 collection, respond with *Failed* as the code attribute of the top level `<lu:Status>` element. A second level
950 `<lu:Status>` MAY be inserted. If so, the code attribute of that second level `<lu:Status>` element MUST be
951 set with the following status code:

952 • CannotFindObject

953 2.46. Discovery Service EPR Profile `notOnOrAfter` attribute name

954 [[LibertyDisco](#)]

955 2.46.1. Summary

956 The `notOnOrAfter` attribute for the ID-WSF Endpoint Reference Profile (section 2.3.1.3) documents the name of
957 the attribute as having an initial lower case 'n' in the normative text as well as in all of the examples throughout the
958 document. However, the schema snippet in that section, the schema appendix, and the separate schema file all define
959 the attribute with an upper case 'N'.

960 2.46.2. Resolution

961 The intent was for the attribute to have a lower case initial character and this follows the general Liberty naming
962 paradigm for attributes, so the schema and schema snippet will be updated to reflect the attribute name with the initial
963 lower case 'n'. The specific changes include:

- 964 1. Change the name of the attribute on line 404 of the specification to be `notOnOrAfter`.
- 965 2. Change the name of the attribute on line 4099 of the specification to be `notOnOrAfter`.
- 966 3. Change the name of the attribute on line 82 of the Discovery Service schema file to be `notOnOrAfter`.

967 2.47. Inconsistent Capitalization of `InteractIfNeeded` attribute

968 [[LibertySOAPBinding](#)]

969 2.47.1. Summary

970 The value for the `interact` attribute is shown as "InteractIfNeeded" on line 1721, but the schema examples on lines
971 1756 and 2265 show the default value as "interactIfNeeded" (different initial capitalization).

972 2.47.2. Resolution

973 Changed occurrences on referenced lines to have uppercase initial character.

References

974

975 [LibertyAuthn] Hodges, Jeff, Aarts, Robert, Madsen, Paul, Cantor, Scott, eds. "Liberty ID-WSF Authentication,
976 Single Sign-On, and Identity Mapping Services Specification ," Version 2.0-errata-v1.0, Liberty Alliance
977 Project (28 November, 2006). <http://www.projectliberty.org/specs>

978 [LibertyClientProfiles] Aarts, Robert, Kainulainen, Jukka, Kemp, John, eds. "Liberty ID-WSF Profiles for Liberty-
979 Enabled User Agents and Devices," Version 2.0-errata-v1.0, Liberty Alliance Project (22 January, 2007).
980 <http://www.projectliberty.org/specs>

981 [LibertyDisco] Cahill, Conor, Hodges, Jeff, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0-
982 errata-v1.0, Liberty Alliance Project (29 November, 2006). <http://www.projectliberty.org/specs>

983 [LibertyInteract] Aarts, Robert, Madsen, Paul, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0-
984 errata-v1.0, Liberty Alliance Project (21 April, 2007). <http://www.projectliberty.org/specs>

985 [LibertyPeopleService] Koga, Yuzo, Madsen, Paul, eds. "Liberty ID-WSF People Service Specification," Version
986 1.0-errata-v1.0, Liberty Alliance Project (06 March, 2007). <http://www.projectliberty.org/specs>

987 [LibertyIDWSF20SCR] Whitehead, Greg, eds. Version 1.0 errata v1.0, Liberty Alliance Project (21 April, 2007).
988 <http://www.projectliberty.org/specs>

989 [LibertySecMech20] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version 2.0-errata-v1.0,
990 Liberty Alliance Project (21 April, 2007). <http://www.projectliberty.org/specs>

991 [LibertySecMech20SAML] Hirsch, Frederick, eds. "ID-WSF 2.0 SecMech SAML Profile," Version 2.0-errata-v1.0,
992 Liberty Alliance Project (08 November, 2006). <http://www.projectliberty.org/specs>

993 [LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, Whitehead, Greg, Madsen, Paul, eds. "Liberty
994 ID-WSF SOAP Binding Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007).
995 <http://www.projectliberty.org/specs>

996 [WSAv1.0] "Web Services Addressing (WS-Addressing) 1.0," Gudgin, Martin, Hadley, Marc, Rogers, Tony, eds.
997 World Wide Web Consortium W3C Recommendation (9 May 2006). [http://www.w3.org/TR/2006/REC-ws-
998 addr-core-20060509/](http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/)