



# Liberty ID-SIS Presence Service Implementation Guidelines

Version: 1.0-12

## **Editors:**

Justin Uberti, AOL

## **Contributors:**

David Castellanos, Ericsson  
Salima Fazal-Karim, France Télécom  
Jukka Kainulainen, Nokia  
Peter Saint-Andre, Jabber  
Yoshitsugu Tsuchiya, NTT  
Rob Lockhart, IEEE-ISTO

## **Abstract:**

This document is an informative guide to implementers of the Liberty Alliance Presence Service, as described in the Liberty ID-SIS Presence Service Specification. It provides basic examples and explanations of how to perform common tasks when using this specification.

The reader is expected to be familiar with the Liberty ID-WSF Web Services Framework Overview, XML, SOAP, and presence services in general.

**Filename:** draft-liberty-id-sis-presence-guidelines-v1.0-12.pdf

1 **Notice**

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.** Implementors  
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 © 2005 America Online, Inc.; American Express Travel Related Services; Ericsson; France Télécom; The  
16 International Security, Trust, and Privacy Alliance (ISTPA); Jabber, Inc.; Nippon Telegraph and Telephone  
17 Corporation; Nokia Corporation; Sun Microsystems, Inc.; and Vodafone Group Plc. All rights reserved.

18  
19 Liberty Alliance Project  
20 Licensing Administrator  
21 c/o IEEE-ISTO  
22 445 Hoes Lane  
23 Piscataway, NJ 08855-1331, USA  
24 info@projectliberty.org  
25

## 26 **Contents**

27	<a href="#">1. Introduction</a>	<a href="#">4</a>
28	<a href="#">2. Recommended Liberty Presence Service Implementation</a>	<a href="#">5</a>
29	<a href="#">3. Privacy Considerations</a>	<a href="#">10</a>
30	<a href="#">References</a>	<a href="#">13</a>

## 1. Introduction

This document provides a rationale and guidance for implementers of the Liberty Presence (ID-SIS-PRES) Service and focuses on guidelines for Presence Service Providers (PSPs) and Presence Service Clients (PSCs). A companion document, Liberty ID-SIS Presence Service Technical Specification [\[LibertyPresence\]](#), normatively describes the ID-SIS-PRES. This document is intended to provide recommendations to developers for matters that are not explicitly formalized in [\[LibertyPresence\]](#). Definitions for Liberty-specific terms can be found in the [\[LibertyGlossary\]](#).

If there is disagreement between this document and [\[LibertyPresence\]](#), the Specification is prescriptive.

### 1.1. Document Audience

This document is intended for application developers and implementers. The reader is presumed to be familiar with XML, SAML, SOAP, and presence services in general. The reader should be familiar, as well, with the Liberty ID-FF Architectural Overview [\[LibertyIDFFOverview\]](#) and the Liberty ID-WSF Web Services Framework Overview [\[LibertyIDWSFOverview\]](#).

Apart from this implementation guidelines document, readers and implementers of the [\[LibertyPresence\]](#) specification will also benefit from the information contained in the following documents:

- Liberty ID-WSF Implementation Guidelines [\[LibertyIDWSFGuide\]](#),
- Liberty ID-WSF Security and Privacy Overview [\[LibertyIDWSFSecurityPrivacyGuidelines\]](#), and
- Privacy and Security Best Practices [\[LibertyPrivacy\]](#)

### 1.2. High Level View of ID-SIS-PRES

The ID-SIS-PRES is provided by an attribute provider (AP) [\[LibertyIDWSFGuide\]](#), which we will refer to as a PSP in this document. The PSP is an ID-WSF web service that hosts the ID-SIS-PRES. PSCs, which may be Service Providers (SPs) themselves, make requests to the PSP on behalf of the Principal [\[LibertyIDWSFGuide\]](#).

## 2. Recommended Liberty Presence Service Implementation

As mentioned in [LibertyPresence], ID-SIS-PRES is a wrapper of sorts around existing presence protocols. Specifically, the service allows a PSC to make [LibertyDST] requests to a PSP concerning the presence information of a Principal; contained in those requests and in the associated replies are XML elements from either the SIMPLE PIDF [RFC3863] or XMPP [RFC3921] protocols specified by the IETF, or the IMPS [PA11] protocol specified by the Open Mobile Alliance. Examples of this encapsulation will be discussed in this section.

In each [LibertyDST] request, the target Principal is identified in the Resource element. It may also be the case with certain encapsulated protocols that the target is also identified in the encapsulated elements. In this case, both occurrences of the ID should match. If not, the value in the encapsulated protocol should be ignored.

### 2.1. Transaction Type Examples

The following are examples of [LibertyDST] operations for presence data. For clarity, the standard SOAP envelope is omitted from these examples; only the SOAP body contents are shown.

#### 2.1.1. Queries for Presence Information

To query for presence information, the client will send a [LibertyDST] Query to the PSP for the Principal for which presence is to be retrieved. The XML namespace used for the query identifies which presence protocol is to be used, and the ResourceID element specifies a URI that references the aforementioned Principal. The PSP will reply with a [LibertyDST] QueryResponse back to the client with a Status element indicating the result of the request and, if successful, a Data element that contains XML elements from the chosen presence protocol.

##### 2.1.1.1. SIMPLE Query/QueryResponse

```
<Query xmlns="urn:liberty:id-sis-pres:simple:2005-07">
  <ResourceID>pres:user@example.com</Resource ID>
  <QueryItem/>
</Query>

<QueryResponse xmlns="urn:liberty:id-sis-pres:simple:2005-07">
  <Status code="OK"/>
  <Data>
    <presence entity="pres:user@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
      <tuple id="bs35r9">
        <status>
          <basic>open</basic>
        </status>
      </tuple>
    </presence>
  </Data>
</QueryResponse>
```

##### 2.1.1.2. XMPP Query/QueryResponse

```
<Query xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
  <ResourceID>pres:user@example.com</ResourceID>
  <QueryItem/>
</Query>

<QueryResponse xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
  <Status code="OK"/>
  <Data>
    <presence from="user@example.com/bs35r9" xmlns="urn:jabber:client"/>
```

```

103     <priority>10</priority>
104   </presence>
105 </Data>
106 </QueryResponse>
107

```

### 108 2.1.1.3. IMPS Query/QueryResponse

```

109
110
111 <Query xmlns="urn:liberty:id-sis-pres:imps:2005-07">
112   <ResourceID>pres:user@example.com</ResourceID>
113   <QueryItem/>
114 </Query>
115
116 <QueryResponse xmlns="urn:liberty:id-sis-pres:imps:2005-07">
117   <Status code="OK" />
118   <Data>
119     <PresenceSubList xmlns="http://www.wireless-village.org/PAL1.1">
120       <UserAvailability>
121         <PresenceValue>AVAILABLE</PresenceValue>
122       </UserAvailability>
123     </PresenceSubList>
124   </Data>
125 </QueryResponse>
126

```

## 127 2.1.2. Modification of Presence Information

128 To modify presence information (typically for itself), the client will send a [\[LibertyDST\]](#) Modify to the PSP. The XML  
 129 namespace used for the request identifies the type of data contained inside, the ResourceID element specifies a URI  
 130 that references the Principal, and the NewData element contains the XML elements from the chosen presence protocol  
 131 that represent the new presence state of the Principal. The PSP will reply with a [\[LibertyDST\]](#) ModifyResponse  
 132 back to the client with a Status element indicating the result of the request.

### 133 2.1.2.1. SIMPLE Modify/ModifyResponse

```

134
135
136 <Modify xmlns="urn:liberty:id-sis-pres:simple:2005-07">
137   <ResourceID>pres:user@example.com</Resource ID>
138   <Modification>
139     <NewData>
140       <tuple id="bs35r9" xmlns="urn:ietf:params:xml:ns:pidf">
141         <status>
142           <basic>closed</basic>
143         </status>
144       </tuple>
145     </NewData>
146   </Modification>
147 </Modify>
148
149 <ModifyResponse xmlns="urn:liberty:id-sis-pres:simple:2005-07">
150   <Status code="OK" />
151 </ModifyResponse>
152

```

### 153 2.1.2.2. XMPP Modify/ModifyResponse

```

154
155
156 <Modify xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
157   <ResourceID>pres:user@example.com</Resource ID>
158   <Modification>

```

```

159     <NewData>
160         <presence from="user@example.com/bs35r9"
161             type="unavailable" xmlns="urn:jabber:client" />
162     </NewData>
163 </Modification>
164 </Modify>
165
166 <ModifyResponse xmlns="urn:liberty:id-sis-pres:xmpp:2005-07">
167     <Status code="OK" />
168 </ModifyResponse>
169

```

### 170 2.1.2.3. IMPS Modify/ModifyResponse

```

171
172
173 <Modify xmlns="urn:liberty:id-sis-pres:imps:2005-07">
174     <ResourceID>pres:user@example.com</Resource ID>
175     <Modification>
176         <NewData>
177             <UserAvailability xmlns="http://www.wireless-village.org/PAL.1">
178                 <PresenceValue>NOT_AVAILABLE</PresenceValue>
179             </UserAvailability>
180         </NewData>
181     </Modification>
182 </Modify>
183
184 <ModifyResponse xmlns="urn:liberty:id-sis-pres:imps:2005-07">
185     <Status code="OK" />
186 </ModifyResponse>
187

```

### 188 2.1.3. Subscriptions for Presence Information

189 In the interest of efficiency, PSCs/PSPs can support subscriptions for presence information so that the PSC is notified  
 190 by the PSP whenever the presence information for the subscribed-to Principal changes, instead of having to ask  
 191 the PSP continually for updates. To start a subscription, the PSC will send a [\[LibertyDST\]](#) Subscribe request,  
 192 again specifying the XML namespace to identify the presence protocol and the ResourceID to identify the user to  
 193 watch, along with an Endpoint URL to indicate where the notifications should be sent. The response to this is a  
 194 [\[LibertyDST\]](#) SubscribeResponse, which contains a Status code indicating the success or failure of the request,  
 195 and, if successful, a Notification element, which is described below.

#### 196 2.1.3.1. SIMPLE Subscribe/SubscribeResponse

```

197
198
199 <Subscribe xmlns="urn:liberty:id-sis-pres:simple:2005-07">
200     <ResourceID>pres:user@example.com</ResourceID>
201     <Subscription expires="2005-10-27T16:49:29Z" invokeID="798fewq">
202         <NotifyTo>
203             <sb-ext:Endpoint xmlns:sb-ext="urn:liberty:sb:2004-04">
204                 http://notify.example.com/sink
205             </sb-ext:Endpoint>
206         </NotifyTo>
207     </Subscription>
208 </Subscribe>
209
210 <SubscribeResponse xmlns="urn:liberty:id-sis-pres:simple:
211 2005-07" timeStamp="2005-07-03T14:49:10Z">
212     <Status code="OK" />
213     <Notification invokeID="798fewq" subscriptionID="ui87" expires="2005-10-27T16:49:29Z">
214         <Data>
215             <presence entity="pres:user@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
216                 <tuple id="bs35r9">

```

```

217         <status>
218             <basic>open</basic>
219         </status>
220     </tuple>
221 </presence>
222 </Data>
223 </Notification>
224 </SubscribeResponse>
225

```

## 2.1.4. Async Notification of Presence Information

When the presence state of a Principal changes, the PSP will notify all subscribed PSCs. This is done by sending a [LibertyDST] Notify request to the PSC, which contains a Notification element with details about the subscription, and, within it, a Data element that contains the XML elements of the chosen presence protocol indicating the subscribed-to Principal's current presence information. Upon receipt of this notification, the PSC will reply with a NotifyResponse, indicating that it processed the Notify properly (or not).

### 2.1.4.1. SIMPLE Notify/NotifyResponse

```

233
234
235 <Notify xmlns="urn:liberty:id-sis-pres:simple:2005-07" timeStamp="2005-07-05T01:19:10Z">
236     <Notification subscriptionID="ui87" expires="2005-10-27T16:49:29Z">
237         <Data>
238             <presence entity="pres:user@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
239                 <tuple id="bs35r9">
240                     <status>
241                         <basic>closed</basic>
242                     </status>
243                 </tuple>
244             </presence>
245         </Data>
246     </Notification>
247 </Notify>
248
249 <NotifyResponse xmlns="urn:liberty:id-sis-pres:simple:2005-07">
250     <Status code="OK"/>
251 </NotifyResponse>
252

```

## 2.2. Suggested Mapping of Presence Elements

As mentioned above, each of the three supported presence protocols have their own format for their presence information. When handling responses in different formats, it may be desirable to have a way to unify the various responses so that the individual presence protocol details are hidden from the higher layers of the application. Table 1 contains a simple, non-normative mapping for some common presence elements.

Table 1. Recommended Mapping of Presence Elements

Presence Field	SIMPLE (PIDF [RFC3863]) Element	(XMPP [RFC3921] Element	IMPS [PA11] Element
Availability: Status	<basic> element (values: "open" or "closed")	'type' attribute (values: "unavailable," if absent; assumed to be "available")	<OnlineStatus> element (values: "T," "F")
Availability Address	:<contact> element	'from' attribute	<Address> element
Availability Free-form status	:<note> element	<status>, or <show> elements	<StatusText> element



259 Mappings for less commonly-used elements are contained in [Table 2](#) below. Note that the SIMPLE elements  
 260 mentioned are from the most recent [\[RPID\]](#) and [\[GEOPRIV\]](#) Internet Drafts, and may be subject to change.

261 **Table 2. Suggested Mapping of Presence Elements**

Presence Field	SIMPLE (PIDF) Element	XMPP <a href="#">[RFC3921]</a> Element	IMPS <a href="#">[PA11]</a> Element
Motion status summary	<a href="#">[RPID]</a> <activities> element	<a href="#">[XMPP_user_activity]</a> <activity> element	N/A
Geolocation	<a href="#">[GEOPRIV]</a> <geopriv>	<geoloc>	GeoLocation element
Mood	<a href="#">[RPID]</a> <Mood> element	<a href="#">[XMPP_user_mood]</a> <mood> element	<StatusMood> element
User Icon	<a href="#">[RPID]</a> <status-icon> element	<photo> element	N/A

262 It is worth noting that some of these common fields in the supported presence protocols overlap with services offered in  
 263 the Liberty framework. Among these services are the Liberty Personal Profile service [\[LibertyIDPP\]](#), which contains  
 264 information about how to contact the user, and the Liberty Geolocation service [\[LibertyGL\]](#). Both of these services  
 265 offer information that is richer from what would be contained in the presence document and should be used, when  
 266 possible.

## 3. Privacy Considerations

Presence information, by its very nature, is fairly sensitive. Even a simple on/off presence indication allows a watcher to have an idea of where a user may be and his or her typical schedule. When detailed status information is added to the presence info, control of the presence information becomes even more of a concern.

### 3.1. Authorization of PSCs

At a minimum, a user should be able to specify which PSCs are able to request presence data for him or her. The consequence of this to the PSP implementer is that incoming presence requests must be checked to ensure that their requesting PSC is authorized by the target Principal to receive his or her presence information.

Basic authorization of PSCs can be accomplished making use of the Liberty ID-WSF-defined SOAP headers:

- Liberty ID-WSF `<Provider>` header block, as defined by [\[LibertySOAPBinding\]](#), provides a means for a sender to claim that it is represented by a given `providerID` value.
- Liberty ID-WSF `<ResourceAccessStatement>` element within the `<saml:Assertion>` element of the Liberty ID-WSF `<wsse:Security>` header block, as defined by [\[LibertySecMech\]](#), conveys sufficient information regarding the accessing entity and the resource for which access is being attempted.

While the implementation and maintenance of any authorization lists is specific to the individual PSP, Liberty supplies some mechanisms to indicate the requestor of the presence data and how the data will be used.

#### 3.1.1. Consent

The [\[LibertySOAPBinding\]](#) specification defines a `<Consent>` header block that allows PSCs to indicate to the PSP that they have obtained consent to query or modify the presence data for a Principal. PSPs can use this mechanism only to allow requests from clients that have explicitly been authorized by the Principal.

##### 3.1.1.1. Example

The sample message below shows the `<Consent>` header block in a SOAP message requesting a particular Principal's presence information:

```
<S:Envelope>
  <S:Header>
    <Consent id="A124395732495743"
      uri="urn:liberty:consent:obtained"
      timestamp="2112-03-15T11:12:10Z"/>
  </S:Header>
  <S:Body>
    <Query xmlns="urn:liberty:id-sis-pres:simple:2005-07">
      <ResourceID>pres:user@example.com</Resource ID>
      <QueryItem/>
    </Query>
  </S:Body>
</S:Envelope>
```

#### 3.1.2. Usage Directives

[\[LibertySOAPBinding\]](#) also provides a `<UsageDirective>` header block that allows PSCs and PSPs to specify limits on how the user's presence data shall be used. The semantics for this block have not yet been fully specified, but one possible usage would involve a URL being specified in the `UsageDirective` block in the presence response, and this URL would reference a policy that describes a set of obligations that the requesting PSC must fulfill or be in breach.

311 Alternatively, the PSC could use the <UsageDirective> to specify a URL that indicates its policy for handling the  
 312 presence data.

### 313 3.1.2.1. Example

```
314
315 <S:Envelope>
316   <S:Header>
317     <UsageDirective S:mustUnderstand="1">
318       <example:PrivacyPolicyReference>
319         http://example.com/policies/presence
320       </example:PrivacyPolicyReference>
321     </UsageDirective>
322   </S:Header>
323   <S:Body>
324     <Query xmlns="urn:liberty:id-sis-pres:simple:2005-07">
325       <ResourceID>pres:user@example.com</Resource ID>
326       <QueryItem/>
327     </Query>
328   </S:Body>
329 </S:Envelope>
330
331
```

### 332 3.1.3. User Interaction

333 The [\[LibertyInteract\]](#) specification is an ID-WSF specification that defines schemas and profiles that enable a PSP to  
 334 interact with the owner of the resource that is exposed by that PSP. [\[LibertyInteract\]](#) defines a profile that enables a  
 335 PSC and a PSP to cooperate in redirecting the resource owner to the PSP and back to the PSC as well as elements,  
 336 processing rules, and the WSDL that, together, define an identity-based interaction service that can be made available  
 337 temporarily by the PSC or offered on a more permanent basis by a party that has the necessary permanent channel to  
 338 the Principal.

339 By definition, an Interaction Service is capable of interacting with the Principal at any time, for example, by using  
 340 special protocols, mechanisms, or channels such as instant messaging or WAP Push. Upon receiving the above  
 341 request from the PSP, the Interaction Service is responsible for "rendering" a "form" to the Principal appropriate to the  
 342 interaction channel.

343 In the context of ID-SIS-PRES, the user-interaction mechanisms defined in [\[LibertyInteract\]](#) will be primarily used  
 344 when the PSP can not accurately decide on the release of the requested presence information.

345 In general, interaction mechanisms also may be used in order to obtain the actual value of an attribute being requested.  
 346 However, in the context of ID-SIS-PRES, this might be applicable only for specific presence information (e.g.,  
 347 "Availability: Free From Status"), but not all, since a PSP should be able to determine some of the Principal's  
 348 presence information itself (e.g., "Status: Online/Offline").

#### 349 3.1.3.1. Example

350 An example of an InteractionRequest sent to the Interaction Service by the PSP that needs to obtain consent for  
 351 the release of the corresponding Principal's presence information to a specific PSC is shown below.

```
352
353 <InteractionRequest xmlns="urn:liberty:is:2003-08">
354   <ResourceID>data:d8ddw6dd7m28v628</ResourceID>
355   <Inquiry title="Profile Provider Question">
356     <Help moreLink="http://presence.example.com/help/consent">
357       Example.com is requesting your presence information. Please pick one of the
358       provided options. Note that the last two options will ensure that you won't be
359       asked this question when Example.com asks for your presence information again.
360     </Help>
361   </Inquiry>
362 </InteractionRequest>
```

```
362     <Select name="presencechoice">
363         <Label>Do you want to share your presence information with Example.com?</Label>
364         <Value>no</Value>
365         <Item label="Not this time" value="no">
366             <Hint> We won't give out your presence information this time but we'll ask you
367                 again next time
368             </Hint>
369         </Item>
370         <Item label="Yes, once" value="yes">
371             <Hint>We will share your presence information and will ask again next time.
372             </Hint>
373         </Item>
374         <Item label="No, never" value="never">
375             <Hint>We won't give out your presence information and won't ask you again
376             </Hint>
377         </Item>
378         <Item label="Yes, always" value="always">
379             <Hint>We will share your presence information now and in the future with Example.com
380             </Hint>
381         </Item>
382     </Select>
383 </Inquiry>
384 </InteractionRequest>
385
```

# References

## Informative

- 386  
387
- 388 [GEOPRIV] Peterson, J., eds. (09 September, 2004). "A Presence-based GEOPRIV Location Object Format," Draft,  
389 The Internet Engineering Task Force <http://www.ietf.org/internet-drafts/draft-ietf-geopriv-pidf-lo-03.txt>
- 390 [LibertyDST] "Liberty ID-WSF Data Services Template Specification," Version 2.0-06, Liberty Alliance Project (22  
391 November, 2004). <http://www.projectliberty.org/specs> Kainulainen, Jukka, Ranganathan, Aravindan, eds.
- 392 [LibertyGL] Kainulainen, Jukka, eds. "Liberty ID-SIS Geolocation Service Specification," Version 1.0-12, Liberty  
393 Alliance Project (23 February, 2005). <http://www.projectliberty.org/specs>
- 394 [LibertyGlossary] Hodges, Jeff, eds. "Liberty Technical Glossary," Version v2.0-01, Liberty Alliance Project (00 Nov  
395 2004). <http://www.projectliberty.org/specs>
- 396 [LibertyIDFFOverview] Wason, Thomas, eds. "Liberty ID-FF Architecture Overview," Version 1.2-errata-v1.0,  
397 Liberty Alliance Project (12 September 2004). <http://www.projectliberty.org/specs>
- 398 [LibertyIDPPP] Kellomäki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.1, Liberty  
399 Alliance Project (25 May, 2005). <http://www.projectliberty.org/specs>
- 400 [LibertyIDWSFGuide] Weitzel, David, eds. (13 January 2005). "Liberty ID-WSF Impelmentation Guideline," Draft  
401 version 2.0-02, Liberty Alliance Project <http://www.projectliberty.org/specs/>
- 402 [LibertyIDWSFOverview] Tourzan, Jonathan, Koga, Yuzo, eds. "Liberty ID-WSF Web Services  
403 Framework Overview," Version 1.0-errata-v1.0, Liberty Alliance Project (12 September 2004).  
404 <http://www.projectliberty.org/specs>
- 405 [LibertyIDWSFSecurityPrivacyGuidelines] Landau, Susan, eds. "Liberty ID-WSF Security and Privacy Overview,"  
406 Version 1.0, Liberty Alliance Project (8 October 2003). <http://www.projectliberty.org/specs>
- 407 [LibertyInteract] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 2.0-01, Liberty  
408 Alliance Project (22 November 2004). <http://www.projectliberty.org/specs>
- 409 [LibertyPresence] Saint-Andre, Peter, eds. "Liberty ID-SIS Presence Service Specification," Version 1.0-10, Liberty  
410 Alliance Project (08 July, 2005). <http://www.projectliberty.org/specs>
- 411 [LibertyPrivacy] Korentayer, E., eds. (14 April 2003). "Project Liberty Privacy and Security Best Practices," Release  
412 2.0, Liberty Alliance Project [http://www.projectliberty.org/specs/Project\\_Liberty\\_Best\\_Practices4.14.03.pdf](http://www.projectliberty.org/specs/Project_Liberty_Best_Practices4.14.03.pdf)  
413
- 414 [LibertySecMech] Ellison, Gary, Madsen, Paul, eds. "Liberty ID-WSF Security Mechanisms," Version 2.0-03, Liberty  
415 Alliance Project (22 November 2004). <http://www.projectliberty.org/specs>
- 416 [LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, eds. "Liberty ID-WSF SOAP Binding Specifica-  
417 tion," Version 2.0-01, Liberty Alliance Project (22 November 2004). <http://www.projectliberty.org/specs>
- 418 [PA11] "Wireless Village Presence Attributes Version 1.1, WV Internal Tracking Number: WV-029," Wireless Village  
419 via the Open Mobile Alliance (2002). <http://www.openmobilealliance.org/tech/affiliates/wv/wvindex.html>
- 420 [RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet  
421 Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt> [March 1997].
- 422 [RFC3863] Sugano, H., Fujimoto, S., Kline, G., Bateman, A., Carr, W., Peterson, J., eds. (August 2004).  
423 "Presence Information Data Format (PIDF)," RFC 3863, The Internet Engineering Task Force  
424 <http://www.ietf.org/rfc/rfc3863.txt> [Aug 2004].

- 425 [RFC3921] Saint-Andre, P., eds. (October 2004). "Extensible Messaging and Presence Protocol (XMPP): Instant  
426 Messaging and Presence," RFC 3921, The Internet Engineering Task Force <http://www.ietf.org/rfc/rfc3921.txt>  
427 [Oct 2004].
- 428 [RPID] Schulzrinne, H., Gurbani, V., Kyzivat, P., Rosenberg, J., eds. (19 February, 2004). "RPID: Rich Presence  
429 Extensions to the Presence Information Data Format (PIDF)," Draft, The Internet Engineering Task Force  
430 <http://www.ietf.org/internet-drafts/draft-ietf-simple-rpid-05.txt>
- 431 [XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C.M., Maler, Eve, eds. (Oct 2000). "Extensible  
432 Markup Language (XML) 1.0 (Second Edition)," Recommendation, World Wide Web Consortium  
433 <http://www.w3.org/TR/2000/REC-xml-20001006>
- 434 [XMPP\_user\_activity] "JEP-0108: User Activity," Saint-Andre, Peter, Meijer, Ralph, eds. Jabber Council v1.0 (20  
435 October, 2004). <http://www.jabber.org/jeps/jep-0108.html>
- 436 [XMPP\_user\_mood] "JEP-0107: User Mood," Saint-Andre, Peter, Meijer, Ralph, eds. Jabber Council v1.0 (20  
437 October, 2004). <http://www.jabber.org/jeps/jep-0107.html>