Liberty ID-SIS Presence Service Specification
Version: 1.0-10

Editors:
Peter Saint-Andre, Jabber, Inc.

Contributors:
Shin Adachi, NTT
Rajeev Angal, SUN
David Castellanos, Ericsson
Salima Fazal-Karim, France Télécom
Jukka Kainulainen, Nokia
Alessandro Maccari, Nokia
Steve Monroe, AOL
Takao Nakanishi, NTT
Kenji Takahashi, NTT
Yoshitsugu Tsuchiya, NTT
Kurt Kolok, IEEE-ISTO
Rob Lockhart, IEEE-ISTO

Abstract:
This specification defines a web service for presence information within the context of the Liberty Alliance project.

Filename: draft-liberty-id-sis-presence-v1.0-10.pdf
Notice

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

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

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

1. Introduction ................................................................. 4  
2. Derivation of ID-SIS-PRES from DST and WSF ............................ 6  
3. Discovery Option Keywords .................................................... 9  
4. Subscriptions and Notifications .............................................. 10  
5. Selection Mechanisms .......................................................... 11  
6. XML Schemata .................................................................. 12  
References ......................................................................... 14
1. Introduction

1.1. Informational Description of Presence Service

The Liberty ID-WSF Data Services Template Specification [LibertyDST] introduced the concept of a data service. In general, a data service hosts information about a Principal and provides the ability to interact with that information (e.g., by querying for or modifying the information). A Presence Service implements the data service pattern by hosting information about the Principal’s availability, enabling other services to query the Presence Service for that information, and enabling the Principal or other authorized entities to modify the Principal’s presence information.

The core meaning of “presence” refers to a Principal’s availability for communications over a network. Examples include availability to talk over a traditional or mobile telephony network, chat over an instant messaging (IM) network, and participate in a video conference. In this core sense, presence is a boolean, “on/off” indicator of network availability.

Over time, this core notion of presence has been extended to include other information about the Principal that changes quickly or that affects the Principal’s interest in or ability to engage in communications. Examples of such “extended presence” include the Principal’s proximity to or interaction with a user agent (e.g., “away” or “do not disturb”), activity (e.g., “driving”), mood (e.g., “grumpy”), and date/time ranges for availability.

This specification normatively defines the nature of a Presence Service in the context of the Liberty Alliance Project.

1.2. Encapsulation of Standard Presence Formats

The Presence Service data format defined herein encapsulates data from three existing presence standards, each of which is identified by a distinct service type:

1. IMPS (formerly Wireless Village) as defined in the Wireless Village 1.1 Presence Attributes specification [PA11] published by the Open Mobile Alliance (OMA); the ServiceType designator is "urn:liberty:id-sis-pres:imps:2005-02."

2. SIMPLE as defined in [RFC3863]: Presence Information Data Format (PIDF) published by the Internet Engineering Task Force (IETF); the ServiceType designator is "urn:liberty:id-sis-pres:simple:2005-02."


N.B. Because these service types depend on protocols produced by other standards development organizations, those implementing this specification need to be aware that the non-Liberty protocols may be subject to change and that such change is outside the control or influence of the Liberty Alliance Project. This specification supports only the versions of the non-Liberty protocols defined in the specific documents referred to herein (i.e., [PA11], [RFC3863], and [RFC3921]).

1.3. Namespaces and Namespace Prefixes

There are three namespaces used for Liberty Alliance Presence Services, depending on the service type:

* urn:liberty:id-sis-pres:imps:2005-02
* urn:liberty:id-sis-pres:simple:2005-02
* urn:liberty:id-sis-pres:xmpp:2005-02
The following table summarizes the [XML] namespaces referenced herein as well as their conventional prefixes.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
</table>

1.4. Notational conventions

The capitalized key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
2. Derivation of ID-SIS-PRES from DST and WSF

As defined herein, a Presence Service is an instance of a data service as described in the [LibertyDST]. All stipulations of the [LibertyDST] are hereby incorporated by reference unless expressly waived or modified in this document.

The following table summarizes the [LibertyDST] general service parameters supported for all service types in this specification.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceType for SIMPLE Support</td>
<td>urn:liberty:id-sis-pres:pidf:2005-02</td>
</tr>
<tr>
<td>ServiceType for XMPP Support</td>
<td>urn:liberty:id-sis-pres:xmpp:2005-02</td>
</tr>
<tr>
<td>Discovery Options</td>
<td>See Section 2</td>
</tr>
<tr>
<td>Data Schema</td>
<td>See Section 6</td>
</tr>
<tr>
<td>SelectType Definition</td>
<td>See Section 5</td>
</tr>
<tr>
<td>Semantics of the &lt;Select&gt; element</td>
<td>See Section 5</td>
</tr>
<tr>
<td>Element Uniqueness</td>
<td>Use XML id attribute for Entity and Account elements; use lang and script attributes for localizable elements</td>
</tr>
<tr>
<td>Data Extension Supported</td>
<td>The service types defined herein encapsulate data defined by other standards development organizations (IETF and OMA), and there is no separate schema for Liberty Alliance Presence Service formats; therefore, data extension MUST be pursued through the mechanisms specified for the encapsulated protocols and MUST NOT be attempted through use of the &lt;Extension&gt; element specified in the [LibertyDST].</td>
</tr>
</tbody>
</table>

The following table summarizes the [LibertyDST] query parameters supported for all service types in this specification.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support querying</td>
<td>MUST</td>
</tr>
<tr>
<td>Multiple &lt;Query&gt; elements</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Multiple &lt;QueryItem&gt; elements</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>Support sorting</td>
<td>No</td>
</tr>
<tr>
<td>&lt;SortType&gt; definition</td>
<td>N/A</td>
</tr>
<tr>
<td>Support changedSince</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Supported formats</td>
<td>All</td>
</tr>
<tr>
<td>Support includeCommonAttributes</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Support pagination</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>Support static sets</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>&lt;Extension&gt; in &lt;Query&gt;</td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

The following table summarizes the [LibertyDST] modify parameters supported for all service types in this specification.
### Table 4. [LibertyDST] Modify Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Modification</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Multiple <code>&lt;Modify&gt;</code> elements</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Multiple <code>&lt;Modification&gt;</code> elements</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Support partial success</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Support <code>notChangedSince</code></td>
<td>SHOULD</td>
</tr>
<tr>
<td><code>&lt;Extension&gt;</code> in <code>&lt;Modify&gt;</code></td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

The following table summarizes the [LibertyDST] subscribe parameters supported for all service types in this specification.

### Table 5. [LibertyDST] Subscribe Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support subscribing to notifications</td>
<td>SHOULD</td>
</tr>
<tr>
<td>Use of the <code>&lt;Subscribe&gt;</code> element for modifying and renewing subscriptions</td>
<td>Modifying, renewing, and canceling existing subscriptions SHOULD be supported</td>
</tr>
<tr>
<td>Multiple <code>&lt;Subscribe&gt;</code> elements</td>
<td>MAY</td>
</tr>
<tr>
<td>Multiple <code>&lt;Subscription&gt;</code> elements</td>
<td>MAY</td>
</tr>
<tr>
<td>Use of the <code>&lt;NotifyEndedTo&gt;</code> element</td>
<td>MAY</td>
</tr>
<tr>
<td><code>&lt;TriggerType&gt;</code> definition</td>
<td>MAY</td>
</tr>
<tr>
<td>Start of a subscription</td>
<td>MAY</td>
</tr>
<tr>
<td>Subscription expiration</td>
<td>If the ServiceType is <code>urn:liberty:id-sis-pres:pidf:2005-02</code>, subscription expiration SHOULDN'T be supported; if the ServiceType is <code>urn:liberty:id-sis-pres:imps:2005-02</code> or <code>urn:liberty:id-sis-pres:xmpp</code>, then subscription expiration SHOULDN'T be supported</td>
</tr>
<tr>
<td>Use of <code>expires</code> and duration attributes</td>
<td>Both MAY be used if subscription expiration is supported</td>
</tr>
<tr>
<td>Support <code>expires==starts</code></td>
<td>MAY</td>
</tr>
<tr>
<td>Support zero duration</td>
<td>MAY</td>
</tr>
<tr>
<td><code>&lt;Extension&gt;</code> in <code>&lt;Subscribe&gt;</code></td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

The following table summarizes the [LibertyDST] QuerySubscriptions parameters supported for all service types in this specification.

### Table 6. [LibertyDST] QuerySubscriptions Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support querying subscriptions</td>
<td>MAY</td>
</tr>
<tr>
<td>Multiple <code>&lt;QuerySubscriptions&gt;</code> elements</td>
<td>MAY</td>
</tr>
<tr>
<td><code>&lt;Extension&gt;</code> in <code>&lt;QuerySubscriptions&gt;</code></td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

The following table summarizes the [LibertyDST] notify parameters supported for all service types in this specification.

### Table 7. [LibertyDST] Notify Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support notifications</td>
<td>MUST</td>
</tr>
<tr>
<td>Are notifications acknowledged</td>
<td>MAY</td>
</tr>
<tr>
<td><code>&lt;Extension&gt;</code> in <code>&lt;Notify&gt;</code></td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

The following table summarizes the [LibertyDST] EndNotify parameters supported for all service types in this specification.
Table 8. [LibertyDST] EndNotify Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support end notifications</td>
<td>MAY</td>
</tr>
<tr>
<td>Are notifications acknowledged</td>
<td>MAY</td>
</tr>
<tr>
<td>&lt;Extension&gt; in &lt;Ended&gt;</td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>
3. Discovery Option Keywords

The [LibertyDST] specifies that discovery option keywords MAY be supported by an implementation and also defines a number of discovery option keywords. Although support for discovery option keywords is OPTIONAL, if a Liberty Alliance Presence Service supports discovery option keywords, it MUST publish the keywords shown in the following table if the relevant features are supported. Also included below is the [RFC2119] conformance level for implementation of the relevant features by a Liberty Alliance Presence Service.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:liberty:dst:allPaths</td>
<td>Does not apply, since [XPATH] expressions are not supported</td>
<td>N/A</td>
</tr>
<tr>
<td>urn:liberty:dst:can:extend</td>
<td>Support for ability to extend schema</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>urn:liberty:dst:changeHistorySupported</td>
<td>Support for changedSince and notChangedSince attributes</td>
<td>SHOULD</td>
</tr>
<tr>
<td>urn:liberty:dst:extend</td>
<td>Support for some schema extensions</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>urn:liberty:dst:fullXPath</td>
<td>Does not apply, since [XPATH] expressions are not supported</td>
<td>N/A</td>
</tr>
<tr>
<td>urn:liberty:dst:multipleModification</td>
<td>Support for ability to modify more than one item per resource at a time</td>
<td>SHOULD</td>
</tr>
<tr>
<td>urn:liberty:dst:multipleQueryItems</td>
<td>Support for ability to query a resource for more than one QueryItem at a time</td>
<td>MUST</td>
</tr>
<tr>
<td>urn:liberty:dst:multipleResources</td>
<td>Support for ability to query more than one resource at a time</td>
<td>SHOULD</td>
</tr>
<tr>
<td>urn:liberty:dst:noModify</td>
<td>No support for modification of items</td>
<td>MAY</td>
</tr>
<tr>
<td>urn:liberty:dst:noQuery</td>
<td>No support for querying of items</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>urn:liberty:dst:noSubscribe</td>
<td>No support for subscriptions and notifications</td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

It may prove helpful for Liberty Alliance Presence Services to advertise the specific presence features that they support (e.g., support for basic availability, free-form status descriptions, user mood, and the like). Support for these keywords is OPTIONAL.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>urn:liberty:id-sis-pres:features:availabilityStatus</td>
<td>Basic on/off availability</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:presenceAddress</td>
<td>The presence address of the presentity</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:statusActivity</td>
<td>A presentity’s current activity</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:statusIcon</td>
<td>An icon associated with the presentity’s current status</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:statusMood</td>
<td>A presentity’s current mood or emotional/physical state</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:statusText</td>
<td>A free-form status description</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:timestamp</td>
<td>A timestamp for the latest presence information</td>
</tr>
<tr>
<td>urn:liberty:id-sis-pres:features:timezone</td>
<td>The time zone in which the presentity is located</td>
</tr>
</tbody>
</table>
4. Subscriptions and Notifications

The [LibertyDST] enables a service provider to request subscriptions to data hosted by another service provider and to receive notifications when specified changes occur related to that data. This functionality maps well to the problem space of presence and availability, since most existing presence technologies enable an entity to subscribe to another entity’s presence data and subsequently receive notifications whenever that presence data changes.

The following considerations apply:

1. There are no differences between the use of the <Select> element inside the <Subscription> element and its use in the <Query> and <Modify> elements.

2. The use of the <Type> element is not defined.

3. The use of the <Trigger> element is not defined; the only presence notifications that MUST be supported (if subscriptions are supported) are triggered when the data specified by the <Select> element have changed.

4. A service MAY support acknowledgement of receiving <Notify> messages, but the decision to do so is implementation-specific and/or a matter of local deployment policy and is not mandated by this service specification.
5. Selection Mechanisms

5.1. Query Selection

In the context of querying for presence data, the <Select> element MAY contain any query selection mechanism defined for the encapsulated protocol (IMPS, SIMPLE, or XMPP). However, definition of such mechanisms MUST be provided by the encapsulated protocol and is out of scope for the Liberty Alliance Presence Service specification. Therefore, the definition of the <SelectType> element is also out of scope. The following broad guidelines may prove helpful to implementers, but are non-normative.

For IMPS, the IMPS specifications developed by the Wireless Village Initiative and now maintained by the Open Mobile Alliance (OMA) allow sending of the IMPS PresenceSubList element with empty child elements as a mechanism of querying for presence information; see section 4.2.1 of [PA11DTD]. However, the IMPS PresenceSubList element is defined by a DTD, not an XML schema. Therefore, while a schema for the IMPS presence namespace is imported by the schema for the "urn:liberty:id-sis-pres:imps:2005-07" namespace, it is non-normative.

For SIMPLE, the IETF's SIMPLE Working Group is in the process of defining query selection mechanisms for presence information but has not yet settled on a stable schema for such mechanisms. If an implementer wishes to use such mechanisms, it can include the relevant formal description in a modified version of the schema for the "urn:liberty:id-sis-pres:simple:2005-07" namespace (if validation is required) or simply include the relevant XML in the <Select> element, since elements from any namespace are allowed per the schema for the "urn:liberty:id-sis-pres:simple:2005-07" namespace. One mechanism under consideration by the SIMPLE WG is defined in Internet-Draft "draft-simple-filter-format," which is a work in progress.

For XMPP, no query selection mechanisms have been defined by the IETF's XMPP WG and a request for presence yields complete information about all resources associated with an XMPP presentity. Therefore, the SelectType for XMPP services is empty.

5.2. Modify Selection

In the context of modifying presence data, a <Modification> element MAY contain a <Select> child element to indicate that only the data specified within the <NewData> element is to be changed. If the <Modification> element does not contain a <Select> child element, then the presence data for the relevant resource shall be changed as specified within the <NewData> element. If the overrideAllowed attribute is set to a value of "False" (this is the default), then the data specified within the <NewData> element shall add new values only and shall not remove or replace existing data. If the overrideAllowed attribute is set to a value of "True," then the data specified within the <NewData> element shall remove or replace the data specified. In order to delete a specific data element, that element shall be included as an empty element within the <NewData> element. The <NewData> may contain any element or combination of elements qualified by the namespace or namespaces used to represent data appropriate to the relevant service type (e.g., in the case of IMPS, the namespace "http://www.wireless-village.org/PA1.1").
6. XML Schemata

The XML Schemata for ID-SIS Presence (IMPS Section 6.1, SIMPLE Section 6.2, and XMPP Section 6.3) follow.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:liberty:id-sis-pres:imps:2005-07"
  xmlns="urn:liberty:id-sis-pres:imps:2005-07"
  xmlns:imps="http://www.wireless-village.org/PA1.1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified">
  <xs:include schemaLocation="liberty-idwsf-dst-v2.0.xsd"/>
  <xs:include schemaLocation="liberty-idwsf-dst-dt-v2.0.xsd"/>
  <xs:import namespace="http://www.wireless-village.org/PA1.1"
    schemaLocation="imps-presence-non-normative-v1.0.xsd"/>
  <xs:complexType name="SelectType">
    <xs:sequence>
      <xs:element ref="imps:PresenceSubList"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SortType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="TriggerType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="TypeType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```


```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="urn:liberty:id-sis-pres:simple:2005-07"
  xmlns="urn:liberty:id-sis-pres:simple:2005-07"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified">
  <xs:include schemaLocation="liberty-idwsf-dst-v2.0.xsd"/>
  <xs:include schemaLocation="liberty-idwsf-dst-dt-v2.0.xsd"/>
  <xs:complexType name="SelectType">
    <xs:sequence>
      <xs:any namespace="##other"
        minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SortType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="TriggerType">
    <xs:complexContent>
      <xs:restriction base="EmptyType"/>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```
<xs:restriction base="EmptyType"/>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="TypeType">
<xs:complexContent>
<xs:restriction base="EmptyType"/>
</xs:complexContent>
</xs:complexType>
</xs:schema>


<?xml version="1.0" encoding="UTF-8"?>
<xs:schema

targetNamespace="urn:liberty:id-sis-pres:xmpp:2005-07"
xmlns="urn:liberty:id-sis-pres:xmpp:2005-07"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="unqualified">
<xs:include schemaLocation="liberty-idwsf-dst-v2.0.xsd"/>
<xs:include schemaLocation="liberty-idwsf-dst-dt-v2.0.xsd"/>
<xs:complexType name="SelectType">
<xs:complexContent>
<xs:restriction base="EmptyType"/>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="SortType">
<xs:complexContent>
<xs:restriction base="EmptyType"/>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="TriggerType">
<xs:complexContent>
<xs:restriction base="EmptyType"/>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="TypeType">
<xs:complexContent>
<xs:restriction base="EmptyType"/>
</xs:complexContent>
</xs:complexType>
</xs:schema>
References

Normative


Informative