



Liberty ID-SIS Geolocation Service Specification

Version: 1.0-12

Editors:

Jukka Kainulainen, Nokia Corporation

Contributors:

David Castellanos, Ericsson

Corina Graham, Ericsson

Rachid Oulahal, France Télécom

Rob Lockhart, IEEE-ISTO

Abstract:

The Liberty ID-SIS Geolocation (ID-SIS-GL) defines a web service. It offers geolocation information regarding a Principal. ID-SIS-GL is an instance of a data oriented identity web service. ID-SIS-GL is using the Liberty ID-WSF Data Services Template and readers of this document should be familiar with that as well as the rest of the Liberty ID-WSF framework. The geolocation related data is mostly from the Mobile Location Protocol version 3.1 specified by the Open Mobile Alliance and readers should be familiar with that also.

Filename: draft-liberty-id-sis-gl-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 American Express Travel Related Services; Ericsson; France Télécom The International Security,
16 Trust, and Privacy Alliance; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Sun Microsystems,
17 Inc.; Symlabs, 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	1. Overview	4
28	2. Service Parameters	6
29	3. Extensions	9
30	4. Discovery Option Keywords	10
31	5. SelectType	12
32	6. TriggerType	18
33	7. Geolocation Information	22
34	8. Examples	25
35	9. Schema	27
36	10. Abstract WSDL	33
37	References	34

1. Overview

The Geolocation Service is a Liberty identity service which enables the sharing of the geolocation information of a Principal. The Geolocation Service (ID-SIS-GL) uses the protocols provided by the Liberty ID-WSF Data Services Template [LibertyDST] to carry the geolocation requests and responses, This specification must be read together with [LibertyDST]. This specification defines, normatively, how [LibertyDST] is used in the Geolocation Service and what geolocation information and parameters are supported by the Geolocation Service.

The geolocation information includes the position of a Principal, speed and direction related information, and information related to the quality of the position information.

A Web Service Provider (WSP) hosting an ID-SIS-GL service may support both querying a Principal's geolocation information and subscribing to a Principal's geolocation information. In special cases, a WSP may also support modifying a Principal's geolocation information through the ID-SIS-GL interface. Usually, a WSP obtains the geolocation information by other means than input from a Web Service Consumer (WSC). The information is usually calculated by a network or a device.

For most of the geolocation-specific data and parameters, the definitions in [MLPv3.1] are used. For the corresponding XML elements and attributes, the normative semantic definitions are in [MLPv3.1], which is referred to in relevant places in this document. In this specification for those XML elements and attributes only, an informative description is given about their semantics. If there are any conflicts in those descriptions, the definition from [MLPv3.1] is used. As [MLPv3.1] uses DTD and this specification uses W3C XML schema [Schema1], the [XML] elements and attributes used from [MLPv3.1] are rewritten using the W3C XML Schema language in this specification. This specification provides the normative type definitions for those elements and attributes. Please note that there are some changes in the type definitions when more specific type definitions provided by the XML schema are used.

In addition to this normative specification, a separate implementation guidelines document is available [LibertyIDGLGuide]. It provides more information related to the ID-SIS-GL (e.g., it includes comparison to [MLPv3.1] and has important privacy notes).

This specification has four main parts. First, the use of [LibertyDST] is specified, how different alternatives features are used, and the needed parameters are defined. These define what kind of requests ID-SIS-GL supports. After that, the supported geolocation-specific parameters and data are defined (ie., what can be received from a WSP hosting an ID-SIS-GL service and what can be sent as possible parameters in requests). After these two parts, some examples are shown and the full XML schema and an abstract WSDL for ID-SIS-GL are presented.

In case of disagreement between this document and any guidelines or XML description, this document is prescriptive. Any published errata are incorporated in this document by reference and, as such, are normative.

1.1. Notation

This specification uses schema documents conforming to W3C XML Schema (see [Schema1]) and normative text to describe the syntax and semantics of XML-encoded protocol messages. Note: Phrases and numbers in brackets [] refer to other documents; details of these references may be found at the end of this document.

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," "MAY NOT," and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]: "they MUST only be used where it is actually required for interoperability or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)."

These keywords are thus capitalized when used to specify, unambiguously, requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

The following namespaces are used in the schema definitions:

81 • The prefix `gl`: stands for the Liberty ID-SIS Geolocation Service schema namespace (`urn:liberty:id-sis-gl:2005-02`).

82 • The prefix `xs`: stands for the W3C XML schema namespace (`http://www.w3.org/2001/XMLSchema`)
83 [\[Schema1\]](#).

84 • The prefix `xml`: stands for the W3C XML namespace (`http://www.w3.org/XML/1998/namespace`) [\[XML\]](#).

85 Please note that the DST schemas [\[LibertyDST\]](#) included do not define any namespace. Elements from those schemas
86 will be in the ID-SIS-GL namespace when used for the ID-SIS-GL.

87 This specification uses the following typographical conventions in text: `<Element>`, `<ns:ForeignElement>`,
88 `attribute`, `Datatype`, and `OtherCode`.

89 Definitions for Liberty-specific terms may be found in [\[LibertyGlossary\]](#).

90 1.2. Time Values

91 All Liberty time values have the type `dateTime`, which is built in to the W3C XML Schema Datatypes specification.
92 Liberty time values MUST be expressed in UTC form, indicated by a "Z" immediately following the time portion of
93 the value.

94 Liberty requesters and responders SHOULD NOT rely on other applications supporting time resolution finer than sec-
95 onds, as implementations MAY ignore fractional second components specified in timestamp values. Implementations
96 MUST NOT generate time instants that specify leap seconds.

2. Service Parameters

The Geolocation Service is an instance of a data service as described by [LibertyDST] and all stipulations of [LibertyDST] are hereby incorporated unless expressly waived or modified in this specification.

The following tables define how the different features of the Data Services Template are used in the Geolocation Service. The templates for these tables are copied from [LibertyDST].

Table 1. General Service Parameters (1/2)

Parameter	Value
<ServiceType>	urn:liberty:id-sis-gl:2005-02
Discovery Options	See Section 4
Data Schema	For full schema see Section 9
SelectType Definition	A special structure is used, see Section 5.
Semantics of the <Select> element	See Section 5.

Table 2. General Service Parameters (2/2)

Parameter	Value
Element uniqueness	Not applicable. No multiple elements with same name used.
Data Extension Supported	The <Extension> element MAY be used to add new geolocation-specific data, but the use is not specified by this specification. See Section 3.

Table 3. Query Parameters (1/2)

Parameter	Value
Support querying	Queries MUST be supported
Multiple <Query> elements	Multiple <Query> elements MAY be supported.
Multiple <QueryItem> elements	Multiple <QueryItem> elements MUST NOT be used.
Support sorting	Not applicable.
SortType definition	Not applicable as sorting is not supported, empty definition used: <pre><xs:complexType name="SortType"> <xs:complexContent> <xs:restriction base="EmptyType"/> </xs:complexContent> </xs:complexType></pre>
Support changedSince	The changedSince attribute MAY be supported when a special need is found for it, but it is RECOMMENDED that it is not used as the geolocation information is very dynamic and rapidly changing and a client should just specify whether it wants current or last geolocation information.
Supported formats	When the changedSince attribute is used, the whole set of current information the requesting WSC is allowed to get is returned as the set is relatively small and usually all the data changes at the same time.

Table 4. Query Parameters (2/2)

Parameter	Value
Support includeCommonAttributes	MUST NOT be supported. Only one common attribute used and a WSP decides when to add it.
Support pagination	Not applicable. MUST NOT be used.
Support static sets	Not applicable. MUST NOT be used.
<Extension> in <Query>	The <Extension> element in <Query> MAY be used for new parameters, but the use is not specified by this specification. See Section 3.

106

Table 5. Modify Parameters

Parameter	Value
Support modification	Modifications MAY be supported in special cases. A WSC MUST NOT expect that in a general case.
Multiple <Modify> elements	Multiple <Modify> elements MAY be supported.
Multiple <Modification> elements	Multiple <Modification> elements MUST NOT be supported.
Support partial success	Not applicable as multiple <Modification> elements are not allowed.
Support notChangedSince	notChangedSince SHOULD NOT be supported.
<Extension> in <Modify>	The <Extension> element in <Modify> MAY be used to specify new parameters, but the use is not specified by this specification. See Section 3 .

107

Table 6. Subscribe Parameters (1/2)

Parameter	Value
Support subscribing to notifications	Subscriptions and notifications SHOULD be supported.
Use of the <Subscribe> element for modifying and renewing subscriptions.	If notification subscriptions are supported, modifying existing subscriptions MAY be supported, but canceling and renewing MUST be supported.
Multiple <Subscribe> elements	Multiple <Subscribe> elements MAY be supported.
Multiple <Subscription> elements	Multiple <Subscription> elements MAY be supported.
Use of the <NotifyEndedTo> element	The <NotifyEndedTo> element MUST be supported, if end notifications are used.
Type definition	The <Type> element is not used, so an empty definition is used for it: <pre><xs:complexType name="TypeType"> <xs:complexContent> <xs:restriction base="EmptyType"/> </xs:complexContent> </xs:complexType></pre>

108

Table 7. Subscribe Parameters (2/2)

Parameter	Value
TriggerType definition	See Section 6 for the description and the type definition of the <Trigger> element.
Start of a subscription	If notifications are supported, the starts attribute MUST be supported.
Subscription expiration	Subscription expiration MUST be used.
Use of expires and duration attributes	Both expires and duration MUST be supported.
Support expires==starts	The same value for both the starts and the expires attribute MUST be supported.
Support zero duration	The value zero MUST be supported for the duration attribute.
<Extension> in <Subscribe>	The <Extension> element in <Subscribe> MAY be used for new parameters, but the use is not specified by this specification. See Section 3 .

109

Table 8. QuerySubscriptions Parameters

Parameter	Value
Support querying existing subscriptions	MAY be supported.
Multiple <QuerySubscriptions> elements	Multiple <QuerySubscriptions> elements MAY be supported.
<Extension> in <QuerySubscriptions>	The <Extension> element in <QuerySubscriptions> MAY be used for new parameters, but the use is not specified by this specification. See Section 3 .

110

Table 9. Notify Parameters

Parameter	Value
Support notifications	Notifications SHOULD be supported.
Are notifications acknowledged	Notifications SHOULD be acknowledged.
<Extension> in <Notify>	The <Extension> element in <Notify> MAY be used to pass additional data, but the use is not specified by this specification. See Section 3 .

111

Table 10. EndNotify Parameters

Parameter	Value
Support end notifications	End notifications MAY be supported.
Are end notifications acknowledged	End notifications SHOULD be acknowledged.
<Extension> in <Ended>	The <Extension> element in <Ended> MAY be used to pass additional data, but the use is not specified by this specification. See Section 3 .

112 3. Extensions

113 The data schema has <Extension> elements which permit arbitrary schema extensions under them. An implemen-
114 tation MAY use those <Extension> elements to support data not defined in this specification. That new data MUST
115 NOT be in the namespace of the ID-SIS-GL and it MUST be geolocation-related data.

116 If an implementation supports new data inside <Extension> elements, it MAY define new discovery option keywords
117 for the extended data. New discovery option keywords MAY be specified also for other purposes.

118 There are also <Extension> elements inside the parameters for different requests. An implementation MAY use
119 those <Extension> elements to support other parameters not defined in this specification. Possible other parameters
120 MUST NOT be in the namespace of the ID-SIS-GL.

121 Although those <Extension> elements exist in the ID-SIS-GL schema, they are not mentioned in this specification
122 when different XML elements are introduced, as the use of those elements is specified in this chapter. The possible
123 content for the <Extension> elements SHOULD be specified so that they may also be ignored and such new data
124 or features are then just not available or used. A WSC or a WSP, compliant to this specification, MAY ignore those
125 <Extension> elements and the other party MUST be prepared for this.

126 For a number of elements and attributes, allowed values are enumerated in this specification. New values for those
127 elements and attributes may be defined, see [\[LibertyReg\]](#).

4. Discovery Option Keywords

A number of discovery option keywords are defined to be included as <Option> elements in discovery registrations and queries, see [LibertyDisco]. A WSP may choose, which keywords it wants to use, if any, as that is an optional feature, but, when used, each keyword MUST be used only for the purpose for which it has been defined.

4.1. Data Availability Discovery Option Keywords

A data availability keyword describes what geolocation information might be available related to a Principal. A WSP MUST NOT register a discovery option keyword if it does not support the geolocation information to which a keyword refers. However, the geolocation information referred to by a keyword which a WSP has registered at a discovery service may not be available to a requesting WSC due to permissions or to the results of positioning operations. Also, a WSP may not register any keyword even if it supports all geolocation data specified in this document. A WSP MUST NOT register a data availability discovery option keyword if it is very likely that the data is not available.

The table below defines the data availability keywords for ID-SIS-GL. The left column lists the keywords and the right column lists the corresponding XML element names. When a keyword is registered at a discovery service, it means that the corresponding XML element should be available. See later chapters for definitions of the XML elements mentioned in the table.

Table 11. Data Availability Keywords

Keyword	The data supported
urn:liberty:id-sis-gl:shape	<shape>
urn:liberty:id-sis-gl:speed	<speed>
urn:liberty:id-sis-gl:direction	<direction>
urn:liberty:id-sis-gl:heading	<Heading>
urn:liberty:id-sis-gl:alt	<alt>
urn:liberty:id-sis-gl:civil-data	<CivilData>

4.2. Other ID-SIS-GL Discovery Option Keywords

A position may be indicated using different coordinate reference systems. A WSP MUST support [WGS-84], but it MAY also support other coordinate reference systems, especially the ones used locally in the area a WSP serves. A WSP MAY register keywords indicating which other coordinate reference systems it supports in addition to the [WGS-84]. Those keywords have the format urn:liberty:id-sis-gl:srs:xxx where xxx is replaced by an identifier for the supported coordinate reference system using the same format as for the srsName attribute (e.g., urn:liberty:id-sis-gl:srs:www.epsg.org:#4004). (See [MLPv3.1] for the definition of the srsName attribute.)

Different WSPs may be able to provide the position in different levels of accuracy. A WSP MAY register one of the keywords defined in the table below at a discovery service to indicate what level of accuracy it should be able to reach for the position of a Principal. Please note that the actual accuracy reached at a specific moment varies based on the situation and also through such mechanisms as the use of the <resp_req> element in the request (see Section 5.1).

Table 12. Position Accuracy Keywords

Keyword	Accuracy
urn:liberty:id-sis-gl:accuracy:high	Better than 50 meters
urn:liberty:id-sis-gl:accuracy:medium	50-500 meters
urn:liberty:id-sis-gl:accuracy:low	Lower than 500 meters

A WSC will often be interested only in the position of a user *relative* to some other location (e.g., airport, store, other principal) rather than their actual location. A WSP MAY support these requests for relative locations. A reference area against which the user's actual location can be compared is added in a request for a relative location and a WSP

160 just returns true or false depending on the comparison result (see [Section 5.5](#)). If a WSP supports requests for relative
161 location, it MAY register the following discovery option keyword `urn:liberty:id-sis-gl:areacomparison`.
162 If a WSP supports only relative location queries and no other features like querying actual location, it MAY register
163 the following discovery option keyword `urn:liberty:id-sis-gl:onlyareacomparison`.

164 **4.3. DST Discovery Option Keyword**

165 [\[LibertyDST\]](#) also defines a number of discovery option keywords. The following keywords defined in [\[LibertyDST\]](#)
166 MAY be used by a WSP providing ID-SIS-GL service:

- 167 • `urn:liberty:dst:extend`
- 168 • `urn:liberty:dst:can:extend`
- 169 • `urn:liberty:dst:multipleResources`
- 170 • `urn:liberty:dst:noModify`
- 171 • `urn:liberty:dst:noSubscribe`
- 172 • `urn:liberty:dst:noQuerySubscriptions`

5. SelectType

[LibertyDST] recommends [XPath] to be used for selecting the data a WSC wants to access. This is, however, not used in the ID-SIS-GL. The ID-SIS-GL also needs other parameters for queries and so a special XML structure is used for the <Select> element. Most of the parameters are from [MLPv3.1]. The <Select> element used in a request may define the geolocation data a WSC wants to have, but it also may define different accuracy and quality aspects a WSC requires from the response.

A <Select> element MAY contain one or more of the following elements: <eqop>, <geo_info>, <loc_type>, <prio>, and <AreaComparison>. Note: a basic query doesn't need the <Select> at all as by default a WSP returns, as allowed by the privacy rule, all geolocation information it has related to a Principal, with the accuracy it is capable of achieving, and uses WGS-84 as the coordinate reference system.

The <Select> element MUST NOT be used in modifications as a WSC MUST provide all available geolocation data related to a Principal at once and not just modify one part of it.

5.1. <eqop> element

The <eqop> element specifies requested quality for the position information. The <eqop> element is from [MLPv3.1] with some modifications. The <eqop> element has the following elements:

- <ll_acc> defines longitude and latitude accuracy in seconds. For the normative semantic definition, see [MLPv3.1].
- <hor_acc> defines horizontal accuracy in meters and is an alternative to the <ll_acc> element. For the normative semantic definition, see [MLPv3.1].
- <alt_acc> defines altitude accuracy in meters. For the normative semantic definition, see [MLPv3.1].
- <max_loc_type> defines maximum allowed age in seconds for the location information. For the normative semantic definition, see [MLPv3.1].
- <resp_req> defines the response time requirement using type attribute, which has following enumeration values (for the normative semantic definition, see [MLPv3.1]):
 - NO_DELAY, a WSP should return any current location information it has related to a Principal immediately.
 - LOW_DELAY, a WSP should give higher priority to the response time than to the location accuracy.
 - DELAY_TOL, a WSP should give higher priority to the accuracy than to the response time. This is the default value.

202 All the elements inside an `<eqop>` element are optional as shown by the schema:

```
203
204
205     <xs:element name="eqop" minOccurs="0">
206       <xs:complexType>
207         <xs:sequence>
208           <xs:element name="resp_req" minOccurs="0">
209             <xs:complexType>
210               <xs:attribute name="type" type="xs:string" default="DELAY_TOL"/>
211             </xs:complexType>
212           </xs:element>
213           <xs:choice minOccurs="0">
214             <xs:element name="ll_acc" type="xs:float"/>
215             <xs:element name="hor_acc" type="xs:float"/>
216           </xs:choice>
217           <xs:element ref="alt_acc" minOccurs="0"/>
218           <xs:element name="max_loc_age" type="xs:integer" minOccurs="0"/>
219           <xs:element ref="Extension" minOccurs="0"/>
220         </xs:sequence>
221       </xs:complexType>
222     </xs:element>
223
224
```

225 If a WSP knows that it can not meet the requirements set by a WSC using any of the elements `<ll_acc>`, `<hor_acc>`,
226 `<alt_acc>`, or `<max_loc_age>`, it MUST NOT accept the request. For queries, this means that the processing of a
227 `<QueryItem>` element MUST fail and a second level `QopNotAttainable` status code SHOULD be used in addition
228 to the top level `Failed` status code. For subscriptions, this means that a subscription is not accepted if a WSP knows
229 that, most probably, it will not meet the requirement in general. The same `QopNotAttainable` status code SHOULD
230 be used, in addition to the top level `Failed` status code, when responding to a subscription request in this kind of
231 a case. If a subscription had been accepted earlier, but a WSP, later, can not meet the requirement with individual
232 positioning, it SHOULD still send a notification when the defined reasons to send a notification are met and it MUST
233 add proper information indicating that the returned position may not meet the requirement (e.g., using the `<lev_conf>`
234 element). The `<resp_req>` element does not define a hard requirement. It provides guidance concerning how a WSP
235 SHOULD handle the request.

236 5.2. `<geo_info>` element

237 The `<geo_info>` element specifies two things: (1) what coordinate reference system should be used and (2) what
238 geolocation information the requestor wants. Note that the default coordinate reference system and the only one
239 which MUST be supported is [WGS-84], when coordinates are used. A `<geo_info>` element MUST contain one or
240 more of the following elements: `<CoordinateReferenceSystem>`, `<CivilData>`, `<shape>`, `<speed>`, `<alt>`,
241 `<direction>`, and `<Heading>`.

242 5.2.1. `<CoordinateReferenceSystem>` element

243 The `<CoordinateReferenceSystem>` element defines the coordinate reference system requested to be used in a
244 response. For the normative semantic definition of the `<CoordinateReferenceSystem>` element, see [MLPv3.1].
245 A `<CoordinateReferenceSystem>` element contains an `<Identifier>` element, which identifies a coordinate
246 reference system. The `<Identifier>` element contains three elements identifying the coordinate reference system.
247 The first element is `<code>`, which is a unique identifier for the coordinate reference system as defined by the authority
248 identified by the content of the `<codeSpace>` second element, which is also inside the `<Identifier>` element.
249 The third element inside the `<Identifier>` element is the `<edition>` element, which defines the version of the
250 coordinate reference system database managed by the authority defined by the `<codeSpace>` element.

251 If the `<CoordinateReferenceSystem>` element is used in a request and a WSP does not support the requested
252 coordinate reference system, the processing of the request fails and a second level `CrsNotSupported` status code

253 should be returned in addition to the top level status code. If there is no `<CoordinateReferenceSystem>` element
 254 in a request, a WSP MUST use [WGS-84].

255 5.2.2. Elements requesting specific geolocation information

256 A `<geo_info>` element may also contain one or more of the elements `<CivilData>`, `<shape>`, `<speed>`, `<alt>`,
 257 `<direction>`, and `<heading>` to indicate that a WSC wants to have exactly those elements.

258 A WSP SHOULD return only the requested geolocation information, but, if it does not support the parameters in
 259 requests, it MAY also return all the geolocation information it has related to a Principal and is allowed to share with the
 260 requesting WSC, taking into account permissions and privacy policies. If a WSP supports the parameters in requests,
 261 but does not have all or part of the requested elements available at a time, it is not, in general, an error condition. That
 262 information is just missing from a response or a notification. If a WSP does not support some requested data at all,
 263 it SHOULD indicate this with a proper second level status code and return that part of the requested data it has. The
 264 status codes `CivilDataNotSupport`, `ShapeNotSupported`, `SpeedNotSupported`, `AltitudeNotSupported`,
 265 `DirectionNotSupported`, and `HeadingNotSupported` are available for this purpose. Note that there may be
 266 multiple second level status codes in a response.

267 When any of the elements `<shape>`, `<speed>`, `<alt>`, `<direction>`, or `<Heading>` are used inside a `<geo_info>`
 268 element, that element MUST NOT have any content (e.g., if a WSC wants to have the altitude, it SHOULD use `<alt>`
 269 inside the `<geo_info>` element or it MAY NOT specify any info in general and just expect to get all the geolocation
 270 information available and allowed). The `<CivilData>` element MAY contain other elements indicating which parts
 271 of the civil data a WSC wants to have. These subelements are only used to indicate what data a WSC wants and MUST
 272 NOT contain anything. For example, if a WSC wants to know in which country and city a Principal is, it may use
 273 following:

```
274
275
276 <geo_info>
277   <CivilData>
278     <L/>
279     <C/>
280   </CivilData>
281 </geo_info>
282
283
```

284 5.2.3. The Schema of the `<geo_info>` Element

```
285
286
287 <xs:element name="geoinfo" minOccurs="0">
288   <xs:complexType>
289     <xs:sequence>
290       <xs:element name="CoordinateReferenceSystem" minOccurs="0">
291         <xs:complexType>
292           <xs:sequence>
293             <xs:element name="Identifier">
294               <xs:complexType>
295                 <xs:sequence>
296                   <xs:element name="code" type="xs:string"/>
297                   <xs:element name="codeSpace" type="xs:string"/>
298                   <xs:element name="edition" type="xs:string"/>
299                 </xs:sequence>
300               </xs:complexType>
301             </xs:element>
302           </xs:sequence>
303         </xs:complexType>
304       </xs:element>
305       <xs:element ref="CivilData" minOccurs="0"/>
306       <xs:element ref="shape" minOccurs="0"/>

```

```
307     <xs:element ref="speed" minOccurs="0"/>
308     <xs:element ref="alt" minOccurs="0"/>
309     <xs:element ref="direction" minOccurs="0"/>
310     <xs:element ref="Heading" minOccurs="0"/>
311     <xs:element ref="Extension" minOccurs="0"/>
312   </xs:sequence>
313 </xs:complexType>
314 </xs:element>
315
316
```

317 5.3. <loc_type> element

318 The <loc_type> element specifies the type of geolocation information a WSC wants to have. For the normative
319 semantic definition, see [MLPv3.1]. The <loc_type> element has a type attribute which can have the following
320 values:

- 321 • **CURRENT**, a WSC wants to know the current geolocation information of a Principal. To meet this requirement, a
322 WSP must not use geolocation information it knew before the time it received a request. It must obtain the current
323 location.
- 324 • **LAST**, a WSC wants to know the last geolocation information a WSP has.
- 325 • **CURRENT_OR_LAST**, a WSP should find out the current geolocation information, but, if it can not get it, it may
326 also return the latest information it has.
- 327 • **INITIAL**, this value is related to originating emergency calls and refers to the geolocation information from the
328 time an emergency call was set-up.

329 The default value is **CURRENT**. If a WSP can not return the correct type of geolocation information in a response
330 to a query, the processing of that <QueryItem> **MUST** fail and a second level **LocTypeNotAvailable** status code
331 **SHOULD** be used in addition to the top level status code. In a subscription request, the value **INITIAL** **MUST NOT** be
332 used. For notification subscriptions, usually, only the default value **CURRENT** makes sense and other values **SHOULD**
333 **NOT** be used.

334 5.4. <prio> element

335 This element specifies the priority of the request. For the normative definition, see [MLPv3.1]. The <prio> element
336 has a *type* attribute, which can have a value of **NORMAL** or **HIGH**. The default value is **NORMAL**. A WSP **SHOULD**
337 prioritize the requests it is handling concurrently based on the priorities requested. A WSP **MAY** decide, itself, to
338 whom it wants to provide high priority service. A WSP **MAY** not support different priorities at all and that is not an
339 error condition, so it **MUST** process the request otherwise normally but ignoring the <prio> element.

340 5.5. <AreaComparison> Element

341 The <AreaComparison> element allows a WSC to ask for the location of a Principal relative to some specified
342 location. The <AreaComparison> element contains a reference area in a form of <CivilData> or <shape>. A
343 WSP compares this reference area to the location of the specified Principal and returns **true**, if the Principal is within
344 the reference area, **false**, if the Principal is not within the reference area and **unknown**, if a WSP is not able to
345 resolve the case. This way, the actual location of the Principal is not revealed as, by default, the location is not
346 returned. The algorithm used to determine whether a Principal is within the specified area or not is out of the scope of
347 this specification.

348 The true/false/unknown comparison result is returned in a <ComparisonResult> element as the first child
349 element of the <Data> element in a response. A WSC may put more than one <AreaComparison> element in
350 a request. In that case, each <AreaComparison> element **MUST** have an *itemID* attribute and each *itemID*

351 attribute MUST have a different value compared to other itemID attributes within the same message. In the
352 response message, when multiple <AreaComparison> elements were used in the request, there MUST be one
353 <ComparisonResult> element in the response for each <AreaComparison> element in the request. When
354 there are multiple <ComparisonResult> elements in one response, each of them MUST have an itemIDRef
355 attribute and the value of an itemIDRef attribute MUST be the value of the itemID attribute of the corresponding
356 <AreaComparison> element.

357 A WSP might not support the format used for defining the reference area, not recognize the given values
358 when <CivilData> is used, the specified reference area might be too small for a WSP to support it, or the
359 Principal's privacy preferences would preclude their location being pinpointed so exactly. In these cases, the
360 processing of the <AreaComparison> element MUST fail and a suitable second level status code SHOULD
361 be used in addition to the top level status code. If the actual location is requested simultaneously, process-
362 ing of that part SHOULD NOT fail if the area comparison fails. The following status codes are defined for
363 the mentioned purposes: ShapeNotSupported, SpecifiedShapeNotSupported, CivilDataNotSupported,
364 SpecifiedCivilDataNotRecognised, and TooSmallArea.

365 A WSP might not support the <AreaComparison> element at all. In that case, it MUST ignore the whole element,
366 SHOULD use the second level status code AreaComparisonNotSupported in the response, and MUST process the
367 request as a normal location query otherwise.

368 A normal query without the <AreaComparison> element inside the <Select> element is expected to return normal
369 location data as specified by other parameters and processing rules. With the <AreaComparison> element, it is still
370 possible to request the normal location data, but a WSC MUST set the returnLocation attribute to true inside the
371 <AreaComparison> element to indicate that, in addition to the relative location, it wants to get the actual location,
372 also. If the returnLocation attribute is set to false, the actual location data MUST NOT be returned, even if
373 there are parameters related to a normal query inside the <Select> element. When the returnLocation attribute
374 is set to true, after processing the relative position request, the WSP MUST process the normal location query as the
375 <AreaComparison> element would not exist, e.g., even though for the area comparison a WSC submitted an area as
376 <CivilData>, the location should be returned in the format specified in the <geo_info> parameter. Note that, if
377 no other parameters than the <AreaComparison> element has been provided inside the <Select> element and the
378 returnLocation attribute is set to true, it is also still a valid query for the actual location as all normal parameters
379 are covered by default values and processing rules. Of course, the policies set by the Principal and/or the WSP might
380 not allow returning the actual location and only the relative location information is given, if allowed.

381 The schema for the <AreaComparison> element:

```
382  
383  
384 <xs:element name="AreaComparison" minOccurs="0" maxOccurs="unbounded">  
385   <xs:complexType>  
386     <xs:choice>  
387       <xs:element ref="CivilData"/>  
388       <xs:element ref="shape"/>  
389       <xs:element ref="Extension"/>  
390     </xs:choice>  
391     <xs:attribute name="itemID" type="IDType"/>  
392     <xs:attribute name="returnLocation" type="xs:boolean" default="false"/>  
393   </xs:complexType>  
394 </xs:element>  
395
```

396 The schema for the <ComparisonResult> element:

```
397  
398  
399 <xs:simpleType name="ComparisonResultType">  
400   <xs:restriction base="xs:string">  
401     <xs:enumeration value="true"/>  
402     <xs:enumeration value="false"/>  
403
```



```
403     <xs:enumeration value="unknown" />
404   </xs:restriction>
405 </xs:simpleType>
406 <xs:element name="ComparisonResult">
407   <xs:complexType>
408     <xs:simpleContent>
409       <xs:extension base="ComparisonResultType">
410         <xs:attribute name="ItemIDRef" type="IDReferenceType" />
411       </xs:extension>
412     </xs:simpleContent>
413   </xs:complexType>
414 </xs:element>
415
```

416 5.6. Common Error Handling

417 In addition to the error cases described related to different parameters and mentioned in [\[LibertyDST\]](#), there are some
418 more geolocation-specific cases. The following second level status codes are defined for those cases:

- 419 • `AbsentSubscriber`, in case of a network-based positioning, the user (Principal) is currently not reachable.
- 420 • `PositionMethodFailure`, a WSP was not able to obtain the position of a Principal.
- 421 • `DisallowedByLocalRegulations`, the request is disallowed by local regulatory requirements.

422 5.7. Modifications

423 A WSP providing ID-SIS-GL service usually obtains the geolocation information by other means, but in special
424 applications it MAY also support updating the geolocation information through the ID-SIS-GL web service interface it
425 provides. When modifications are done using `<Modify>`, the requesting WSC MUST use only one `<Modification>`
426 element. The `<Select>` element MUST NOT be used and the `<NewData>` element MUST contain all current content
427 for the `<pd>` element. It is not allowed to modify only one part. All available information must be from the same
428 moment of time.

6. TriggerType

There may be several different reasons triggering notifications. A WSC MUST define in a subscription request what should trigger notifications unless basic change notifications are used. This is done using the <Trigger> element in subscription requests. The ID-SIS-GL supports the following types of notifications:

- Notifications are sent when the geolocation information of a Principal changes.
- Notifications are sent periodically.
- Notifications are sent based on an event (e.g., when a mobile device a Principal is using and which is used for positioning a Principal is connected to a network).
- Notifications are sent, when the Principal enters or leaves a specified area.

A WSP may not support notifications at all or it may support only certain type of notifications. Notification may be delayed or not sent at all due to privacy reasons.

6.1. Change Notifications

The change notifications are the basic notifications supported by [LibertyDST]. The <Select> element is used normally to indicate what data a WSC is interested in. A WSC MAY also define the granularity (ie., how small of a change should trigger a notification). This change is related only to the position (e.g., it is not possible to request notifications when the direction has changed by 90 degrees). The change granularity is specified by the <Granularity> element, which specifies the granularity in meters.

```
<xs:element name="Granularity" type="xs:positiveInteger" minOccurs="0" />
```

A WSP MAY NOT support the granularity requested by a WSC. In that case the processing of the subscription request MUST fail and the second level RequestedGranularityNotSupported status code SHOULD be used in addition to the top level status code. The minimum granularity supported by a WSP MAY be returned in the comment attribute of the second level <Status> element. When the minimum granularity is returned, the comment attribute MUST contain the text "Minimum granularity XXX meters." where the XXX is replaced with the right value. If a WSC does not put a <Trigger> element inside a subscription request, it is requesting a change notification without specifying any granularity. In this case, a WSP decides the default granularity to be used. A WSP MAY also implement a minimum filtering period during which a new notification is not sent, even if the position change has been big enough.

A WSP MAY support change notifications without supporting the <Granularity> element in subscription requests. If a WSP receives a request containing a <Granularity> element, although it does not support it, the processing of that request MUST fail and a second level GranularityNotSupported status code SHOULD be used in addition to the top level status code. If a WSP does not support change notifications in general and it receives a subscription request for such, the processing of that request MUST fail and the second level ChangeNotificationsNotSupported status code SHOULD be used in addition to the top level status code.

6.2. Periodic Notifications

For periodic notifications, a WSC MUST specify the interval in a subscription request. The interval is defined by an <Interval> element inside a <Trigger> element.

```
<xs:element name="Interval" type="xs:duration" minOccurs="0" />
```

471
472

473 If a WSC specifies a too short interval from the point of view of a WSP, a WSP MAY reject the subscription request.
474 If a WSP rejects a subscription request for this reason it SHOULD use a second level `TooShortInterval` status
475 code in addition to the top level status code. It MAY also return the minimum interval in the `comment` attribute of the
476 second level `<Status>` element. When the minimum interval is returned, the `comment` attribute MUST contain the
477 text "Minimum interval XXX ." where the XXX is replaced with the minimum value using the format specified for
478 `xs:duration`.

479 If a WSP supports notifications, but does not support periodic notifications, the processing of a subscription request
480 MUST fail and a second level `PeriodicNotificationsNotSupported` status code SHOULD be used in addition
481 to the top level status code.

482 6.3. Notifications Based On Other Events

483 A WSP may also support notifications triggered by a defined event. The only event defined at the moment is that
484 of a mobile device becoming available to a network. If a WSC wants to get a geolocation notification when the
485 mobile device a Principal is using becomes available to a network, it MUST add an `<ms_event>` element inside
486 a `<Trigger>` element. See [\[MLPv3.1\]](#) for the normative semantic definitions of the `<ms_event>` element. The
487 `<ms_event>` element MUST have a type attribute for which only the value `MS_AVAIL` is defined in this specification
488 as inherited from [\[MLPv3.1\]](#).

489
490
491
492
493
494
495
496
497

```
<xs:element name="ms_action" minOccurs="0">  
  <xs:complexType>  
    <xs:attribute name="type" type="xs:string" use="required"/>  
  </xs:complexType>  
</xs:element>
```

498 If a WSP does not support this type of notification, the processing of the subscription request MUST fail and a second
499 level `EventNotificationNotSupported` status code SHOULD be used in addition to the top level status code.

500 6.4. Area-Based Notifications

501 A WSC might want to know when a Principal either enters, leaves, or is within a specified area. There is a new element
502 defined for this purpose: `<ChangeArea>`. The actual area is specified either using a `<shape>` element, which contains
503 a geometric shape defining an area, or by using a `<CivilData>` element, which contains different elements of an
504 address to the level needed to define an area (e.g., country and city). The `<ChangeArea>` element also contains a
505 mandatory `event` attribute to specify what kind of an event related to the specified area should trigger a notification.
506 The following values are defined for the `event` attribute: `ENTERING`, `LEAVING`, and `WITHIN`.

507
508
509
510
511
512
513
514
515
516
517
518
519
520

```
<xs:element name="ChangeArea" minOccurs="0" maxOccurs="unbounded">  
  <xs:complexType>  
    <xs:choice>  
      <xs:element ref="CivilData"/>  
      <xs:element ref="shape"/>  
      <xs:element ref="Extension"/>  
    </xs:choice>  
    <xs:attribute name="event" type="xs:string" use="required"/>  
  </xs:complexType>  
</xs:element>
```

521 If a WSP does not support area notifications and it receives a subscription request for such, the processing of that
 522 subscription request MUST fail and the second level `AreaNotificationsNotSupported` status code SHOULD
 523 be used in addition to the top level status code. A WSP might not support the format used for defining the area,
 524 not recognize the given values when `<CivilData>` is used, or the specified area might be too small for a WSP
 525 to support it. In these cases, the processing of that subscription request MUST fail and a suitable second level
 526 status code SHOULD be used in addition to the top level status code. The following status codes are defined
 527 for the mentioned purposes: `ShapeNotSupported`, `SpecifiedShapeNotSupported`, `CivilDataNotSupported`,
 528 `SpecifiedCivilDataNotRecognised`, and `TooSmallArea`.

529 Note that a WSP may send an area notification much later than the actual event happened (e.g., it might not have been
 530 able to locate a Principal at the time when she entered or left a specified area). In this case, the notification is sent
 531 when a WSP becomes aware of that fact. A WSP also may not know about the whole event at all (e.g., a WSC has
 532 subscribed to get a notification when a Principal enters certain area, but a WSP may not be able to get the geolocation
 533 information of a Principal while the Principal was inside that specified area).

534 6.5. The Schema for the TriggerType

```
535
536
537 <xs:complexType name="TriggerType">
538   <xs:sequence>
539     <xs:element name="Granularity" type="xs:positiveInteger" minOccurs="0"/>
540     <xs:element name="Interval" type="xs:duration" minOccurs="0"/>
541     <xs:element name="ms_action" minOccurs="0">
542       <xs:complexType>
543         <xs:attribute name="type" type="xs:string" use="required"/>
544       </xs:complexType>
545     </xs:element>
546     <xs:element name="ChangeArea" minOccurs="0" maxOccurs="unbounded">
547       <xs:complexType>
548         <xs:choice>
549           <xs:element ref="CivilData"/>
550           <xs:element ref="shape"/>
551           <xs:element ref="Extension"/>
552         </xs:choice>
553         <xs:attribute name="event" type="xs:string" use="required"/>
554       </xs:complexType>
555     </xs:element>
556     <xs:element ref="Extension" minOccurs="0"/>
557   </xs:sequence>
558 </xs:complexType>
559 <xs:element name="TriggerReason" type="xs:string"/>
560
561
```

562 6.6. The reason for a notification

563 As with one request, a WSC may subscribe to multiple different types of notifications. A notification message must
 564 indicate what the reason for the notification was. This is indicated with a `<TriggerReason>` element inside the
 565 `<Data>` element of a notification. The following values are defined for the `<TriggerReason>`:

- 566 • PERIODIC
- 567 • CHANGE
- 568 • MS-AVAIL
- 569 • ENTERING

570 • LEAVING

571 • WITHIN

572 A WSP MAY support notification types not defined in this specification and there MAY be other reason codes for
573 those notifications.

574 The <TriggerReason> MUST be before the <pd> element inside the <Data> element in a notification message.

7. Geolocation Information

Responses and notifications contain geolocation information inside <Data> elements provided by [LibertyDST]. The geolocation information is always inside a <pd> element. Most of the content of the <pd> element are from [MLPv3.1]. A <pd> element has the ACC attribute, which is defined in [LibertyDST]. In ID-SIS-GL, it is used to describe how the geolocation information has been obtained (e.g., is it calculated by a network or a device? entered by a human?). When the geolocation data is calculated by a network or a device (e.g., GPS), the value urn:liberty:dst:acc:challenge SHOULD be used to indicate this. When the data is entered by a human, a WSP does not know whether that data was calculated or entered or it does not want or is not allowed to release that information. The ACC MUST NOT be used or the value urn:liberty:dst:acc:unknown MUST be used in such cases. All the elements inside the <pd> element as well as the ACC attribute are optional as a <pd> element MAY have a different content depending, for example, on the request parameters, access rights and privacy policies, and the results of the positioning methods at each time.

The schema for the <pd> element:

```
588
589
590 <xs:element name="pd">
591   <xs:complexType>
592     <xs:sequence>
593       <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
594       <xs:element ref="CivilData" minOccurs="0"/>
595       <xs:element ref="shape" minOccurs="0"/>
596       <xs:sequence minOccurs="0">
597         <xs:element ref="alt" type="xs:float"/>
598         <xs:element ref="alt_acc" type="xs:float" minOccurs="0"/>
599       </xs:sequence>
600       <xs:element ref="speed" type="xs:float" minOccurs="0"/>
601       <xs:element ref="direction" type="xs:float" minOccurs="0"/>
602       <xs:element ref="heading" type="xs:float" minOccurs="0"/>
603       <xs:element name="lev_conf" type="xs:float" minOccurs="0"/>
604       <xs:element ref="Extension" minOccurs="0"/>
605     </xs:sequence>
606     <xs:attribute ref="ACC"/>
607   </xs:complexType>
608 </xs:element>
609
610
```

7.1. <time> element

The value of the <time> element defines the time when the information contained in a <pd> element was valid (e.g., calculated by a network). See [MLPv3.1] for a normative semantic definition for this element, but please note that the type definition in ID-SIS-GL is different as the xs:dateTime is used. A WSP SHOULD always include a <time> element in a response or a notification. It is especially important when a WSP is not returning the current geolocation information, but the last one it has related to a Principal.

7.2. <CivilData> element

The <CivilData> element contains the position of a Principal in the format of a street address. The same format used by the <Address> element in the [LibertyIDPP] is used as the base with the exception that only those common attributes needed for the localization are used. One new subelement containing the Mobile Network Code is added compared to the <Address> element in the [LibertyIDPP]. See [LibertyIDPP] for the definitions of the other elements inside the <CivilData> element. Please note that the Mobile Network Code values are country-specific, so it SHOULD be used only together with the country code.

```
624
625
626 <xs:element name="CivilData" type="CivilDataType"/>
```

```

627 <xs:complexType name="CivilDataType">
628   <xs:sequence>
629     <xs:element name="PostalAddress" type="xs:string" minOccurs="0"/>
630     <xs:element name="LPostalAddress" type="LocalizedString" minOccurs="0"
631       maxOccurs="unbounded"/>
632     <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
633     <xs:element name="L" type="xs:string" minOccurs="0"/>
634     <xs:element name="LL" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
635     <xs:element name="St" type="xs:string" minOccurs="0"/>
636     <xs:element name="LSt" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
637     <xs:element name="C" type="xs:string" minOccurs="0"/>
638     <xs:element name="MNC" type="xs:string" minOccurs="0"/>
639     <xs:element ref="Extension" minOccurs="0"/>
640   </xs:sequence>
641 </xs:complexType>
642 <xs:complexType name="LocalizedString">
643   <xs:simpleContent>
644     <xs:extension base="xs:string">
645       <xs:attribute ref="xml:lang" use="required"/>
646       <xs:attribute ref="script"/>
647     </xs:extension>
648   </xs:simpleContent>
649 </xs:complexType>
650
651

```

652 7.3. <shape> element

653 The <shape> element contains the position of a Principal as a geometric shape. The shape is defined by a subelement,
654 which is inside a <shape> element. A Principal is somewhere in the area covered by that geometric shape. A WSP
655 chooses which shape to use based on the position method it is using. As many WSCs may not be able to utilize all
656 the different shapes properly, a WSP SHOULD try to return a shape which is easy for a WSC to use for its purposes
657 (e.g., a point). See [\[MLPv3.1\]](#) for the normative semantic definitions as well as for the informative introduction to the
658 different shapes.

```

659
660
661 <xs:element name="shape">
662   <xs:complexType>
663     <xs:choice minOccurs="0">
664       <xs:element ref="Point"/>
665       <xs:element ref="LineString"/>
666       <xs:element ref="Polygon"/>
667       <xs:element ref="Box"/>
668       <xs:element ref="CircularArea"/>
669       <xs:element ref="CircularArcArea"/>
670       <xs:element ref="EllipticalArea"/>
671       <xs:element ref="GeometryCollection"/>
672       <xs:element ref="MultiLineString"/>
673       <xs:element ref="MultiPoint"/>
674       <xs:element ref="MultiPolygon"/>
675       <xs:element ref="Extension"/>
676     </xs:choice>
677   </xs:complexType>
678 </xs:element>
679
680

```

681 7.4. <alt> and <alt_acc> elements

682 The <alt> element is used to indicate the altitude of a Principal in meters. See [\[MLPv3.1\]](#) for normative semantic
683 definitions of the <alt> element. The altitude is in relation to the ellipsoid of the coordinate reference system used.

684 When an <alt> element is included in a response or a notification, the accuracy of that altitude may be described by
685 an <alt_acc> element in the same response or notification. The accuracy of an altitude is also reported in meters.

686 7.5. <speed> element

687 When the speed of a Principal is returned in a response or a notification, it is reported in meters per second. See
688 [\[MLPv3.1\]](#) for normative semantic definitions of the <speed> element.

689 7.6. <direction> element

690 A <direction> element indicates the direction towards which a Principal is moving. It is reported in degrees (0-360,
691 0=360=True North). See [\[MLPv3.1\]](#) for normative semantic definitions of the <direction> element.

692 7.7. <Heading> element

693 The <heading> is different than the <direction>. The <heading> defines to which direction a Principal is facing,
694 so it may be defined also when a Principal is not moving. The <heading> is also reported in degrees (0-360,
695 0=360=True North).

696 7.8. <lev_conf> element

697 The <lev_conf> gives an estimate of the probability that a Principal is located at the area defined by the <shape>
698 element returned in the same response or notification. See [\[MLPv3.1\]](#) for normative semantic definitions of the
699 <lev_conf> element. When no <shape> element is included in a response or notification, but there is the
700 <CivilData> element, the <lev_conf> element estimates the probability that a Principal is located at the area
701 defined by the <CivilData> element.

702 7.9. Emergency service related elements

703 In addition to the position data inside a <pd> element, a response or a notification message MAY also contain one
704 or two additional elements when a message is sent to an emergency service. These two elements are the Emergency
705 Services Routing Digits <esrd> and the Emergency Services Routing Key <esrk>. See [\[MLPv3.1\]](#) for the definition
706 of the semantics of these elements. These elements are added directly under the <Data> element at the same level
707 as the <pd> element, when needed. When present, the order MUST be following: <esrd>, <esrk>, and <pd>. A
708 WSP MUST know, by other means, when to add these elements in a response or a notification message, as request
709 messages do not indicate, specifically, the need for these elements (ie., a WSP MUST know that a message is related
710 to an emergency service/call).

```
711  
712  
713 <xs:element name="esrd">  
714   <xs:complexType>  
715     <xs:simpleContent>  
716       <xs:extension base="xs:string">  
717         <xs:attribute name="type" type="xs:string" />  
718       </xs:extension>  
719     </xs:simpleContent>  
720   </xs:complexType>  
721 </xs:element>  
722 <xs:element name="esrk">  
723   <xs:complexType>  
724     <xs:simpleContent>  
725       <xs:extension base="xs:string">  
726         <xs:attribute name="type" type="xs:string" />  
727       </xs:extension>  
728     </xs:simpleContent>  
729   </xs:complexType>  
730 </xs:element>  
731  
732
```


8. Examples

733

734 The request message to query the location of a user with a number of parameters:

735

736

```
737 <gl:Query xmlns:gl="urn:liberty:id-sis-gl:2005-02" >
738   <gl:ResourceID>http://location.com/659gft565</gl:ResourceID>
739   <gl:QueryItem>
740     <gl:Select>
741       <gl:eqop>
742         <gl:resp_req type="LOW_DELAY"/>
743         <gl:hor_acc>1000</gl:hor_acc>
744       </gl:eqop>
745       <gl:loc_type type="CURRENT_OR_LAST"/>
746       <gl:prio type="HIGH"/>
747     </gl:Select>
748   </gl:QueryItem>
749 </gl:Query>
```

750

751

752 An example response:

753

754

```
755 <gl:QueryResponse xmlns:gl="urn:liberty:id-sis-gl:2005-02">
756   <gl:Status code="OK"/>
757   <gl:Data>
758     <gl:pd>
759       <gl:time>2002-06-23-13:44:53Z</gl:time>
760       <gl:shape>
761         <gl:CircularArea>
762           <gl:coord>
763             <gl:x>30 16 28.312N</gl:x>
764             <gl:y>45 15 33.431E</gl:y>
765           <gl:coord>
766             <gl:radius>240</gl:radius>
767         </gl:CircularArea>
768       </gl:shape>
769     </gl:pd>
770   </gl:Data>
771 </gl:QueryResponse>
```

772

773

774 Another response example for a case when it was not possible to attain the requested quality of positioning:

775

776

```
777 <gl:QueryResponse xmlns:gl="urn:liberty:id-sis-gl:2005-02">
778   <gl:Status code="Failed">
779     <gl:Status code="QoPNotAttainable"/>
780   </gl:Status>
781 </gl:QueryResponse>
```

782

783

784 In a basic query, no parameters need to be defined to get all of a Principal's current geolocation information that is available and allowed. A request message for this may be simply:

786

787

```
788 <gl:Query xmlns:gl="urn:liberty:id-sis-gl:2005-02" >
789   <gl:ResourceID>http://location.com/659gft565</gl:ResourceID>
790 </gl:Query>
```

791
792

793 In some cases, there is no need to specify the resource (see [\[LibertyDST\]](#)), so the request also may be simply:

794
795
796 `<gl:Query xmlns:gl="urn:liberty:id-sis-gl:2005-02" />`
797
798

9. Schema

799

800

801

802 <?xml version="1.0" encoding="UTF-8"?>

803 <xs:schema

804 targetNamespace="urn:liberty:id-sis-gl:2005-02"

805 xmlns="urn:liberty:id-sis-gl:2005-02"

806 xmlns:xs="http://www.w3.org/2001/XMLSchema"

807 elementFormDefault="qualified" version="1.0-10">

808 <xs:include schemaLocation="liberty-idwsf-dst-v2.0.xsd"/>

809 <xs:include schemaLocation="liberty-idwsf-dst-dt-v2.0.xsd"/>

810

811

812 <xs:complexType name="SortType">

813 <xs:complexContent>

814 <xs:restriction base="EmptyType"/>

815 </xs:complexContent>

816 </xs:complexType>

817 <xs:complexType name="TypeType">

818 <xs:complexContent>

819 <xs:restriction base="EmptyType"/>

820 </xs:complexContent>

821 </xs:complexType>

822 <xs:complexType name="SelectType">

823 <xs:sequence>

824 <xs:element name="AreaComparison" minOccurs="0" maxOccurs="unbounded">

825 <xs:complexType>

826 <xs:choice>

827 <xs:element ref="CivilData"/>

828 <xs:element ref="shape"/>

829 <xs:element ref="Extension"/>

830 </xs:choice>

831 <xs:attribute name="itemID" type="IDType"/>

832 <xs:attribute name="returnLocation" type="xs:boolean" default="false"/>

833 </xs:complexType>

834 </xs:element>

835 <xs:element name="eqop" minOccurs="0">

836 <xs:complexType>

837 <xs:sequence>

838 <xs:element name="resp_req" minOccurs="0">

839 <xs:complexType>

840 <xs:attribute name="type" type="xs:string" default="DELAY_TOL"/>

841 </xs:complexType>

842 </xs:element>

843 <xs:choice minOccurs="0">

844 <xs:element name="ll_acc" type="xs:float"/>

845 <xs:element name="hor_acc" type="xs:float"/>

846 </xs:choice>

847 <xs:element ref="alt_acc" minOccurs="0"/>

848 <xs:element name="max_loc_age" type="xs:integer" minOccurs="0"/>

849 <xs:element ref="Extension" minOccurs="0"/>

850 </xs:sequence>

851 </xs:complexType>

852 </xs:element>

853 <xs:element name="geoinfo" minOccurs="0">

854 <xs:complexType>

855 <xs:sequence>

856 <xs:element name="CoordinateReferenceSystem" minOccurs="0">

857 <xs:complexType>

858 <xs:sequence>

859 <xs:element name="Identifier">

860 <xs:complexType>

861 <xs:sequence>

862 <xs:element name="code" type="xs:string"/>

863 <xs:element name="codeSpace" type="xs:string"/>

864 <xs:element name="edition" type="xs:string"/>

```

865         </xs:sequence>
866     </xs:complexType>
867 </xs:element>
868 </xs:sequence>
869 </xs:complexType>
870 </xs:element>
871 <xs:element ref="CivilData" minOccurs="0"/>
872 <xs:element ref="shape" minOccurs="0"/>
873 <xs:element ref="speed" minOccurs="0"/>
874 <xs:element ref="alt" minOccurs="0"/>
875 <xs:element ref="direction" minOccurs="0"/>
876 <xs:element ref="Heading" minOccurs="0"/>
877 <xs:element ref="Extension" minOccurs="0"/>
878 </xs:sequence>
879 </xs:complexType>
880 </xs:element>
881 <xs:element name="loc_type" minOccurs="0">
882 <xs:complexType>
883 <xs:attribute name="type" type="xs:string" default="CURRENT"/>
884 </xs:complexType>
885 </xs:element>
886 <xs:element name="prio" minOccurs="0">
887 <xs:complexType>
888 <xs:attribute name="type" type="xs:string" default="NORMAL"/>
889 </xs:complexType>
890 </xs:element>
891 <xs:element ref="Extension" minOccurs="0"/>
892 </xs:sequence>
893 </xs:complexType>
894 <xs:simpleType name="ComparisonResultType">
895 <xs:restriction base="xs:string">
896 <xs:enumeration value="true"/>
897 <xs:enumeration value="false"/>
898 <xs:enumeration value="unknown"/>
899 </xs:restriction>
900 </xs:simpleType>
901 <xs:element name="ComparisonResult">
902 <xs:complexType>
903 <xs:simpleContent>
904 <xs:extension base="ComparisonResultType">
905 <xs:attribute name="ItemIDRef" type="IDReferenceType"/>
906 </xs:extension>
907 </xs:simpleContent>
908 </xs:complexType>
909 </xs:element>
910 <xs:complexType name="TriggerType">
911 <xs:sequence>
912 <xs:element name="Granularity" type="xs:positiveInteger" minOccurs="0"/>
913 <xs:element name="Interval" type="xs:duration" minOccurs="0"/>
914 <xs:element name="ms_action" minOccurs="0">
915 <xs:complexType>
916 <xs:attribute name="type" type="xs:string" use="required"/>
917 </xs:complexType>
918 </xs:element>
919 <xs:element name="ChangeArea" minOccurs="0" maxOccurs="unbounded">
920 <xs:complexType>
921 <xs:choice>
922 <xs:element ref="CivilData"/>
923 <xs:element ref="shape"/>
924 <xs:element ref="Extension"/>
925 </xs:choice>
926 <xs:attribute name="event" type="xs:string" use="required"/>
927 </xs:complexType>
928 </xs:element>
929 <xs:element ref="Extension" minOccurs="0"/>
930 </xs:sequence>
931 </xs:complexType>
    
```

```

932 <xs:element name="TriggerReason" type="xs:string"/>
933 <xs:element name="pd">
934   <xs:complexType>
935     <xs:sequence>
936       <xs:element name="time" type="xs:dateTime" minOccurs="0"/>
937       <xs:element ref="CivilData" minOccurs="0"/>
938       <xs:element ref="shape" minOccurs="0"/>
939       <xs:sequence minOccurs="0">
940         <xs:element ref="alt" type="xs:float"/>
941         <xs:element ref="alt_acc" type="xs:float" minOccurs="0"/>
942       </xs:sequence>
943       <xs:element ref="speed" type="xs:float" minOccurs="0"/>
944       <xs:element ref="direction" type="xs:float" minOccurs="0"/>
945       <xs:element ref="Heading" type="xs:float" minOccurs="0"/>
946       <xs:element name="lev_conf" type="xs:float" minOccurs="0"/>
947       <xs:element ref="Extension" minOccurs="0"/>
948     </xs:sequence>
949     <xs:attribute ref="ACC"/>
950   </xs:complexType>
951 </xs:element>
952 <xs:element name="shape">
953   <xs:complexType>
954     <xs:choice minOccurs="0">
955       <xs:element ref="Point"/>
956       <xs:element ref="LineString"/>
957       <xs:element ref="Polygon"/>
958       <xs:element ref="Box"/>
959       <xs:element ref="CircularArea"/>
960       <xs:element ref="CircularArcArea"/>
961       <xs:element ref="EllipticalArea"/>
962       <xs:element ref="GeometryCollection"/>
963       <xs:element ref="MultiLineString"/>
964       <xs:element ref="MultiPoint"/>
965       <xs:element ref="MultiPolygon"/>
966       <xs:element ref="Extension"/>
967     </xs:choice>
968   </xs:complexType>
969 </xs:element>
970 <xs:element name="Box">
971   <xs:complexType>
972     <xs:sequence>
973       <xs:element ref="coord"/>
974       <xs:element ref="coord"/>
975     </xs:sequence>
976     <xs:attribute name="gid" type="xs:ID"/>
977     <xs:attribute name="srsName" type="xs:string"/>
978   </xs:complexType>
979 </xs:element>
980 <xs:element name="CircularArcArea">
981   <xs:complexType>
982     <xs:sequence>
983       <xs:element ref="coord"/>
984       <xs:element ref="inRadius"/>
985       <xs:element ref="outRadius"/>
986       <xs:element ref="startAngle"/>
987       <xs:element ref="stopAngle"/>
988       <xs:element ref="angularUnit" minOccurs="0"/>
989       <xs:element ref="distanceUnit" minOccurs="0"/>
990     </xs:sequence>
991     <xs:attribute name="gid" type="xs:ID"/>
992     <xs:attribute name="srsName" type="xs:string"/>
993   </xs:complexType>
994 </xs:element>
995 <xs:element name="CircularArea">
996   <xs:complexType>
997     <xs:sequence>
998       <xs:element ref="coord"/>

```

```

999         <xs:element ref="radius"/>
1000         <xs:element ref="distanceUnit" minOccurs="0"/>
1001     </xs:sequence>
1002     <xs:attribute name="gid" type="xs:ID"/>
1003     <xs:attribute name="srsName" type="xs:string"/>
1004 </xs:complexType>
1005 </xs:element>
1006 <xs:element name="EllipticalArea">
1007     <xs:complexType>
1008         <xs:sequence>
1009             <xs:element ref="coord"/>
1010             <xs:element ref="angle"/>
1011             <xs:element ref="semiMajor"/>
1012             <xs:element ref="semiMinor"/>
1013             <xs:element ref="angularUnit"/>
1014             <xs:element ref="distanceUnit" minOccurs="0"/>
1015         </xs:sequence>
1016         <xs:attribute name="gid" type="xs:ID"/>
1017         <xs:attribute name="srsName" type="xs:string"/>
1018     </xs:complexType>
1019 </xs:element>
1020 <xs:element name="GeometryCollection">
1021     <xs:complexType>
1022         <xs:sequence>
1023             <xs:element ref="shape" maxOccurs="unbounded"/>
1024         </xs:sequence>
1025         <xs:attribute name="gid" type="xs:ID"/>
1026         <xs:attribute name="srsName" type="xs:string"/>
1027     </xs:complexType>
1028 </xs:element>
1029 <xs:element name="LineString">
1030     <xs:complexType>
1031         <xs:sequence>
1032             <xs:element ref="coord"/>
1033             <xs:element ref="coord" maxOccurs="unbounded"/>
1034         </xs:sequence>
1035         <xs:attribute name="gid" type="xs:ID"/>
1036         <xs:attribute name="srsName" type="xs:string"/>
1037     </xs:complexType>
1038 </xs:element>
1039 <xs:element name="LinearRing">
1040     <xs:complexType>
1041         <xs:sequence>
1042             <xs:element ref="coord" minOccurs="3" maxOccurs="unbounded"/>
1043         </xs:sequence>
1044         <xs:attribute name="gid" type="xs:ID"/>
1045         <xs:attribute name="srsName" type="xs:string"/>
1046     </xs:complexType>
1047 </xs:element>
1048 <xs:element name="MultiLineString">
1049     <xs:complexType>
1050         <xs:sequence>
1051             <xs:element ref="LineString" maxOccurs="unbounded"/>
1052         </xs:sequence>
1053         <xs:attribute name="gid" type="xs:ID"/>
1054         <xs:attribute name="srsName" type="xs:string"/>
1055     </xs:complexType>
1056 </xs:element>
1057 <xs:element name="MultiPoint">
1058     <xs:complexType>
1059         <xs:sequence>
1060             <xs:element ref="Point" maxOccurs="unbounded"/>
1061         </xs:sequence>
1062         <xs:attribute name="gid" type="xs:ID"/>
1063         <xs:attribute name="srsName" type="xs:string"/>
1064     </xs:complexType>
1065 </xs:element>
    
```

```
1066 <xs:element name="MultiPolygon">
1067   <xs:complexType>
1068     <xs:choice maxOccurs="unbounded">
1069       <xs:element ref="Polygon"/>
1070       <xs:element ref="Box"/>
1071       <xs:element ref="CircularArea"/>
1072       <xs:element ref="CircularArcArea"/>
1073       <xs:element ref="EllipticalArea"/>
1074     </xs:choice>
1075     <xs:attribute name="gid" type="xs:ID"/>
1076     <xs:attribute name="srsName" type="xs:string"/>
1077   </xs:complexType>
1078 </xs:element>
1079 <xs:element name="Point">
1080   <xs:complexType>
1081     <xs:sequence>
1082       <xs:element ref="coord"/>
1083     </xs:sequence>
1084     <xs:attribute name="gid" type="xs:ID"/>
1085     <xs:attribute name="srsName" type="xs:string"/>
1086   </xs:complexType>
1087 </xs:element>
1088 <xs:element name="Polygon">
1089   <xs:complexType>
1090     <xs:sequence>
1091       <xs:element ref="outerBoundaryIs"/>
1092       <xs:element ref="innerBoundaryIs" minOccurs="0" maxOccurs="unbounded"/>
1093     </xs:sequence>
1094     <xs:attribute name="gid" type="xs:ID"/>
1095     <xs:attribute name="srsName" type="xs:string"/>
1096   </xs:complexType>
1097 </xs:element>
1098 <xs:element name="X" type="xs:string"/>
1099 <xs:element name="Y" type="xs:string"/>
1100 <xs:element name="Z" type="xs:string"/>
1101 <xs:element name="angle" type="xs:string"/>
1102 <xs:element name="angularUnit" type="xs:string"/>
1103 <xs:element name="coord">
1104   <xs:complexType>
1105     <xs:sequence>
1106       <xs:element ref="X"/>
1107       <xs:element ref="Y" minOccurs="0"/>
1108       <xs:element ref="Z" minOccurs="0"/>
1109     </xs:sequence>
1110   </xs:complexType>
1111 </xs:element>
1112 <xs:element name="distanceUnit" type="xs:string"/>
1113 <xs:element name="inRadius" type="xs:string"/>
1114 <xs:element name="innerBoundaryIs">
1115   <xs:complexType>
1116     <xs:sequence>
1117       <xs:element ref="LinearRing"/>
1118     </xs:sequence>
1119   </xs:complexType>
1120 </xs:element>
1121 <xs:element name="outRadius" type="xs:string"/>
1122 <xs:element name="outerBoundaryIs">
1123   <xs:complexType>
1124     <xs:sequence>
1125       <xs:element ref="LinearRing"/>
1126     </xs:sequence>
1127   </xs:complexType>
1128 </xs:element>
1129 <xs:element name="radius" type="xs:string"/>
1130 <xs:element name="semiMajor" type="xs:string"/>
1131 <xs:element name="semiMinor" type="xs:string"/>
1132 <xs:element name="startAngle" type="xs:string"/>
```

```
1133 <xs:element name="stopAngle" type="xs:string"/>
1134 <xs:element name="alt" type="xs:float"/>
1135 <xs:element name="alt_acc" type="xs:float"/>
1136 <xs:element name="speed" type="xs:float"/>
1137 <xs:element name="direction" type="xs:float"/>
1138 <xs:element name="Heading" type="xs:float"/>
1139 <xs:element name="CivilData" type="CivilDataType"/>
1140 <xs:complexType name="CivilDataType">
1141   <xs:sequence>
1142     <xs:element name="PostalAddress" type="xs:string" minOccurs="0"/>
1143     <xs:element name="LPostalAddress" type="LocalizedString" minOccurs="0"
1144       maxOccurs="unbounded"/>
1145     <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
1146     <xs:element name="L" type="xs:string" minOccurs="0"/>
1147     <xs:element name="LL" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
1148     <xs:element name="St" type="xs:string" minOccurs="0"/>
1149     <xs:element name="LSt" type="LocalizedString" minOccurs="0" maxOccurs="unbounded"/>
1150     <xs:element name="C" type="xs:string" minOccurs="0"/>
1151     <xs:element name="MNC" type="xs:string" minOccurs="0"/>
1152     <xs:element ref="Extension" minOccurs="0"/>
1153   </xs:sequence>
1154 </xs:complexType>
1155 <xs:complexType name="LocalizedString">
1156   <xs:simpleContent>
1157     <xs:extension base="xs:string">
1158       <xs:attribute ref="xml:lang" use="required"/>
1159       <xs:attribute ref="script"/>
1160     </xs:extension>
1161   </xs:simpleContent>
1162 </xs:complexType>
1163 <xs:element name="esrd">
1164   <xs:complexType>
1165     <xs:simpleContent>
1166       <xs:extension base="xs:string">
1167         <xs:attribute name="type" type="xs:string"/>
1168       </xs:extension>
1169     </xs:simpleContent>
1170   </xs:complexType>
1171 </xs:element>
1172 <xs:element name="esrk">
1173   <xs:complexType>
1174     <xs:simpleContent>
1175       <xs:extension base="xs:string">
1176         <xs:attribute name="type" type="xs:string"/>
1177       </xs:extension>
1178     </xs:simpleContent>
1179   </xs:complexType>
1180 </xs:element>
1181 </xs:schema>
1182
1183
```


10. Abstract WSDL

```
1184
1185
1186
1187 <?xml version="1.0" encoding="UTF-8"?>
1188 <definitions name="id-sis-gl_2005-02_interface "
1189     targetNamespace="urn:liberty:id-sis-gl:2005-02:wsdl:interface"
1190     xmlns:typens="urn:liberty:id-sis-gl:2005-02:wsdl:interface "
1191     xmlns="http://schemas.xmlsoap.org/wsdl/"
1192     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1193     xmlns:gl="urn:liberty:id-sis-gl:2005-02">
1194
1195
1196     <!-- Abstract WSDL for Liberty ID-SIS Geolocation Interface Specification -->
1197     <types>
1198         <xsd:schema>
1199             <xsd:import schemaLocation="liberty-idsis-gl.xsd"/>
1200         </xsd:schema>
1201     </types>
1202     <message name="Query">
1203         <part name="body" element="gl:Query"/>
1204     </message>
1205     <message name="QueryResponse">
1206         <part name="body" element="gl:QueryResponse"/>
1207     </message>
1208     <message name="Modify">
1209         <part name="body" element="gl:Modify"/>
1210     </message>
1211     <message name="ModifyResponse">
1212         <part name="body" element="gl:ModifyResponse"/>
1213     </message>
1214     <message name="Subscribe">
1215         <part name="body" element="gl:Subscribe"/>
1216     </message>
1217     <message name="SubscribeResponse">
1218         <part name="body" element="gl:SubscribeResponse"/>
1219     </message>
1220     <message name="QuerySubscriptions">
1221         <part name="body" element="gl:QuerySubscriptions"/>
1222     </message>
1223     <message name="Subscriptions">
1224         <part name="body" element="gl:Subscriptions"/>
1225     </message>
1226     <portType name="GeolocPort">
1227         <operation name="GeolocQuery">
1228             <input message="typens:Query"/>
1229             <output message="typens:QueryResponse"/>
1230         </operation>
1231         <operation name="GeolocModify">
1232             <input message="typens:Modify"/>
1233             <output message="typens:ModifyResponse"/>
1234         </operation>
1235         <operation name="GeolocSubscribe">
1236             <input message="typens:Subscribe"/>
1237             <output message="typens:SubscribeResponse"/>
1238         </operation>
1239         <operation name="GeolocQuerySubscriptions">
1240             <input message="typens:QuerySubscriptions"/>
1241             <output message="typens:Subscriptions"/>
1242         </operation>
1243     </portType>
1244 </definitions>
1245
1246
```

References

Normative

- 1247
1248
- 1249 [LibertyDisco] Beatty, John, Hodges, Jeff, Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification,"
1250 Version 2.0-02, Liberty Alliance Project (24 Nov 2004). <http://www.projectliberty.org/specs>
- 1251 [LibertyDST] "Liberty ID-WSF Data Services Template Specification," Version 2.0-06, Liberty Alliance Project (22
1252 November, 2004). <http://www.projectliberty.org/specs> Kainulainen, Jukka, Ranganathan, Aravindan, eds.
- 1253 [LibertyGlossary] Hodges, Jeff, eds. "Liberty Technical Glossary," Version 1.3-errata-v1.0, Liberty Alliance Project
1254 (12 Aug 2004). <http://www.projectliberty.org/specs>
- 1255 [LibertyIDPP] Kellomaki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.0, Liberty
1256 Alliance Project (12 November 2003). <http://www.projectliberty.org/specs>
- 1257 [LibertyReg] Kemp, John, eds. "Liberty Enumeration Registry Governance," Version 1.0, Liberty Alliance Project (12
1258 November 2003). <http://www.projectliberty.org/specs>
- 1259 [MLPv3.1] "Mobile Location Protocol (MLP) Candidate Version 3.1," Open Mobile Alliance (16 March 2004).
1260 http://member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/OMA-LIF-MLP-
1261 [V3_1-20040316-C.zip](http://member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/OMA-LIF-MLP-V3_1-20040316-C.zip)
- 1262 [RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet
1263 Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt> [March 1997].
- 1264 [Schema1] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (May
1265 2002). "XML Schema Part 1: Structures," Recommendation, World Wide Web Consortium
1266 <http://www.w3.org/TR/xmlschema-1/>
- 1267 [WGS-84] "WGS 84 Implementation Manual, v2.4," (EUROCONTROL), European Organization for the Safety of Air
1268 Navigation, (IfEN), Institute of Geodesy and Navigation, eds. World Geodetic System 1984 (12 February,
1269 1998). <http://www.wgs84.com/files/wgsman24.pdf>
- 1270 [XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C.M., Maler, Eve, eds. (Oct 2000). "Extensible
1271 Markup Language (XML) 1.0 (Second Edition)," Recommendation, World Wide Web Consortium
1272 <http://www.w3.org/TR/2000/REC-xml-20001006>
- 1273 [XPath] Clark, J., DeRose, S., eds. (16 November 1999). "XML Path Language (XPath) Version 1.0,"
1274 Recommendation, W3C <http://www.w3.org/TR/xpath> [August 2003].

Informative

- 1275
- 1276 [LibertyIDGLGuide] Graham, Corina, Castellanos, David, Kainulainen, Jukka, eds. "Liberty ID-SIS Geolocation Ser-
1277 vice Implementation Guidelines," Version 1.0-15, (23 February, 2005). <http://www.projectliberty.org/specs>