



# Liberty ID-SIS Directory Access Protocol

Version: 1.0

**Editors:**

Sampo Kellomäki, Symlabs, Inc.

**Contributors:**

Guillermo Lorenzo, Symlabs, Inc.

Rob Lockhart, IEEE-ISTO

**Abstract:**

The Liberty ID-SIS Directory Access Protocol (ID-SIS-DAP) specifies a web service offering directory information. ID-SIS-DAP is an instance of a data-oriented identity web service, based on the Liberty ID-WSF Data Services Template (DST). ID-SIS-DAP is characterized by the ability to query and update attribute data and incorporates, from other specifications, mechanisms for access control and for conveying data validation information and usage directives. Readers of this document should be familiar with LDAP, SOAP, SAML, and XML. Readers may also wish to familiarize themselves with the Liberty ID-SIS Personal Profile (ID-SIS-PP) and ID-SIS Contact Book (ID-SIS-CB), which provide similar function, but with standardized schemata.

**Filename:** liberty-id-sis-dap-v1.0.pdf

---

**Notice**

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

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

15 Copyright © 2005-2007 Ericsson; NeuStar; Symlabs, Inc.; Telefónica Móviles; and Vodafone Group Plc. All rights  
16 reserved.

17  
18 Liberty Alliance Project  
19 Licensing Administrator  
20 c/o IEEE-ISTO  
21 445 Hoes Lane  
22 Piscataway, NJ 08855-1331, USA  
23 info@projectliberty.org  
24

---

## Contents

25		
26	1. Introduction .....	4
27	1.1. Notational Conventions .....	4
28	1.2. Derivation of ID-SIS-DAP from DST and WSF .....	4
29	1.3. Compliance .....	4
30	1.4. Namespaces .....	5
31	2. Definition of Parameters (checklists) .....	6
32	2.1. DST-Mandated Parameters .....	6
33	2.2. Subscription Parameters .....	9
34	3. Mapping to LDAP .....	11
35	4. Processing Rules and Other Considerations .....	12
36	4.1. Query Is Not Required to Report Same Data to Repeated Queries .....	12
37	4.2. Simulation (dry-run) is Required .....	12
38	4.3. Use of Usage Directives .....	12
39	4.4. Granularity of Metadata .....	12
40	4.5. Subscription to Change in an Attribute .....	12
41	5. DST 1.1 Support .....	14
42	6. Examples .....	15
43	6.1. ID-SIS-DAP Query Request .....	15
44	6.2. ID-SIS-DAP QueryResponse .....	16
45	7. Schema for ID-SIS-DAP .....	18
46	7.1. Summary of ID-SIS-DAP .....	18
47	7.2. Formal XML Schema for ID-SIS-DAP .....	20
48	8. WSDL for ID-SIS-DAP .....	26
49	References .....	28

## 1. Introduction

The ID-SIS Directory Access Protocol (ID-SIS-DAP) enables Liberty identity web services that pass directory data with arbitrary schema over a web services interface.

This document is a Liberty ID Services Interface Specification that normatively specifies the ID-SIS-DAP Service.

In case of disagreement between the present document and any guidelines or XML schema descriptions, this document is prescriptive. Any published errata is hereby incorporated to this document by reference and as such is normative.

### 1.1. Notational Conventions

When capitalized, the key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]. When these words are not capitalized, they are meant in their natural-language sense.

N.B. LDAP [RFC2251] and colloquial directory terminology is used where ever possible. In LDAP, a typical container is an *entry*, which contains *attributes* which can be multivalued. Contrast this with an attribute of an XML element, which is always single-valued. To avoid confusion, the latter will always be referred to as an "XML attribute." A directory entry is identified by a *distinguished name*. XML elements are sometimes identified by XPaths [XPATH], but this specification does not use XPath. Directory data is often described by a *schema*, see [RFC2252], but XML protocols are often described by a schema as well [Schema1-2]. The latter will always be termed an "XML schema."

For better readability, XML schemata are described using notation defined in Section 1.3 "Schema Grammar Formalism" of [LibertyDST].

### 1.2. Derivation of ID-SIS-DAP from DST and WSF

The ID-SIS-DAP service is an instance of the Liberty ID-WSF Data Services Template [LibertyDST] and all stipulations therein are hereby incorporated unless expressly waived or modified in this document. However, a special provision is made that either version 1.1 or 2.1 of DST MAY be used. An implementation MUST support DST 2.1 [LibertyDST]. Support for DST 1.1 [LibertyDST11] will naturally not be as rich in features, but useful, never-the-less, in the interim while sites are migrating to DST 2.1. This document is written from a DST 2.1 perspective. The DST 1.1 specifics are isolated in Section 5.

A service that consults the ID-SIS-DAP service SHOULD use the Liberty architectural framework, see [LibertyProtSchema] and [LibertyIDWSFGuide10]. The Liberty architectural framework is relied upon to ensure that a service acts on behalf of the Principal or that the Principal has consented to sharing the data. A service that consults an ID-SIS-DAP-compliant service MUST adhere to the specifications that comprise the Liberty architectural framework, see [LibertyProtSchema] and [LibertyIDWSFOverview]. A service MUST be able to demonstrate adherence to the specifications.

A special provision is made that a service MAY support ID-SIS-DAP using ID-WSF 1.1. A service MUST support ID-WSF 2.0. The choice of ID-WSF version is orthogonal to choice of DST version and an implementation MAY support any or all four combinations. The combination of DST 2.1 with ID-WSF 2.0 MUST be supported.

### 1.3. Compliance

This specification defines an interface to which an *implementation* and an *instance* (deployment) of ID-SIS-DAP service MUST conform. For an AP to be ID-SIS-DAP-compliant, it MUST adhere to all aspects of the specification.

A minimally-compliant ID-SIS-DAP implementation MAY choose not to support optional features of this specification. Such an implementation may be labeled as an "ID-SIS-DAP implementation" provided that publicly-available

91 documentation about the implementation clearly discloses which optional parts of the schema and which features are  
92 not supported. All other features and schema are assumed to be supported. A deployment of an implementation that is  
93 not capable of supporting the full schema SHOULD only register the discovery option keywords that accurately reflect  
94 its capabilities.

95 An implementation that supports all of the schema and features specified in this document MAY be labeled as a  
96 "full ID-SIS-DAP implementation." An implementation that is deficient in any feature or part of the schema MUST  
97 NOT be labeled as a "full ID-SIS-DAP implementation." A "full ID-SIS-DAP implementation" deployment MAY  
98 administratively or, via configuration, restrict the schema and features.

99 An ID-SIS-DAP implementation MUST publicly disclose which DST and ID-WSF versions it supports. Labeling of  
100 an implementation that does not yet support DST 2.1 and ID-WSF 2.0 must be qualified by designator "(interim)."

101 A deployment that supports all of the schema and features specified in this document MAY be labeled as a "full  
102 ID-SIS-DAP deployment" or a "full ID-SIS-DAP service." To meet "full ID-SIS-DAP deployment" status, all of the  
103 schema and features MUST be supported for all Principals wishing to use them, barring a policy decision excluding  
104 specific Principal(s).

105 A deployment that only supports some subset of ID-SIS-DAP may still be labeled as an "ID-SIS-DAP deployment" or  
106 an "ID-SIS-DAP service" provided that the deployment publicly discloses the subset that it supports.

107 An ID-SIS-DAP deployment or instance MUST publicly disclose which DST and ID-WSF versions it supports.  
108 Labeling of a deployment or implementation that does not yet support DST 2.1 and ID-WSF 2.0 must be qualified by  
109 designator "(interim)."

## 110 1.4. Namespaces

111 The ID-SIS-DAP namespace is abbreviated as "dap:" in this document. If the namespace has been omitted at any  
112 place in this document, "dap:" should be considered the default namespace. The namespace URI is also used as the  
113 ServiceType designator.

114 **Table 1. Referenced XML namespaces**

Prefix	URI	Description
dap:	urn:liberty:id-sis-dap:2006-08:dst-2.1	See <a href="#">Section 2.1</a> , checklist 1
dst:	urn:liberty:dst:2006-08	<a href="#">[LibertyDST]</a>
subs:	urn:liberty:ssos:2006-08	<a href="#">[LibertySUBS]</a>
xml:	http://www.w3.org/TR/REC-xml	XML Definition <a href="#">[XML]</a> (for xml:lang)
xs:	http://www.w3.org/2002/XMLSchema	XML Schema Definition <a href="#">[Schema1-2]</a>

## 2. Definition of Parameters (checklists)

### 2.1. DST-Mandated Parameters

In Section 10 of [LibertyDST], a checklist is provided for service specifications. This section corresponds, item for item, to that checklist.

1. ID-SIS-DAP uses ServiceType URN urn:liberty:id-sis-dap:2006-08:dst-2.1

2. For service schema, see Section 7

3. Following object types are supported

entry                    Object named by *distinguished name*, often referred to as dn and containing a directory entry as defined in [RFC2251].

\_Subscription            A subscription object, as defined in [LibertyDST]. Furthermore the subscriptions themselves should follow the semantics of [LDAPPSearch].

4. AppDataType is defined as follows

```
%AppDataType: redef(liberty-idwsf-dst-v2.1.xsd)
dap:LDIF | dst:Subscription
;
```

The <LDIF> element represents objects of type "entry" with contents as specified in [RFC2849].

The <Subscription> elements represent objects of type "\_Subscription" with schema as specified in [LibertyDST].

5. Definition of SelectType:

```
%SelectType:
dn?                    -> %xs:string
filter?                -> %xs:string
@scope?                -> %xs:integer default(0)
@sizelimit?            -> %xs:integer default(0)
@timelimit?            -> %xs:integer default(0)
@attributes?           -> %xs:string
@typesonly?            -> %xs:boolean default(false)
@derefaliases?         -> %xs:integer default(0)
;
```

a. If objectType is "entry" then, with reference to Section 4.5.1 "Search Request" of [RFC2251], the <dn> is mapped to baseObject and all other elements and XML attributes are mapped to correspondingly-named fields of SearchRequest production. The attributes single-valued XML attribute is converted to an AttributeDescriptionList by splitting the string on comma characters and forming an AttributeDescription from each resulting substring.

b. The Search filter supplied in <Select> SHOULD be passed verbatim to the underlying directory system or repository. Full [RFC2254] language MUST be accepted by the ID-SIS-DAP service, though it need not ensure that the underlying directory understands the filters. Extensions to filter language MAY be accepted.

c. If <dn> is omitted, or incomplete, the ID-SIS-DAP server MUST determine the correct *distinguished name* from the identity information conveyed by the ID-WSF layer.

- 158 d. If `<dn>` is provided, the ID-SIS-DAP server MAY further determine the actual *distinguished name* from the  
159 identity information conveyed by the ID-WSF layer. In this case, the `<dn>` could act as the *base suffix*, while  
160 the identity information would provide the *naming attribute*.
- 161 e. If `<Select>` is used in `<DeleteItem>` or `<ModifyItem>` relating to `objectType "entry"`, `<Filter>` element  
162 MUST NOT be specified and NONE of the XML attributes MUST be specified.
- 163 f. In case of omission of an XML attribute, the specified default value MUST be used. Usually, these mean  
164 that the limitation does not apply or, in case of `attributes`, that all attributes, except metadata attributes,  
165 should be returned.
- 166 g. An implementation that is not LDAP-based MAY interpret the fields in any other way that achieves the  
167 semantics inherent in [\[RFC2251\]](#).
- 168 h. A specification that profiles ID-SIS-DAP MAY specify additional `objectTypes` and MUST specify  
169 which query language applies to each of them. For example, a relational database profile could specify  
170 `objectType "relational"` with query language SQL.

171 6. The `TestOpType` is defined to be equal to `SelectType` with all the filter language provisions for the various  
172 `objectTypes`. If a filter is not by its nature a test, then it is considered *true* if the result of its evaluation against  
173 underlying backend is nonempty. For example, if an LDAP search filter does not return any entries, then it  
174 evaluates to *false*.

175 7. Contents of `SortType` string is a dollar-separated (ASCII 0x24) list of *sort keys* in descending order of  
176 importance, where each sort key is a comma-separated list of three elements:

- 177 a. Order: "a" == ascending (reverse order false, the default), "d" == descending (reverse order true),
- 178 b. Ordering rule (empty means default for the attribute),
- 179 c. Attribute name.

180 These elements are interpreted in the sense of [\[RFC2891\]](#), Section 1.1 "Request Control."

181 Example:

```
182 a,,cn$d,,fn$d,,sn
```

185 An implementation that does not support sorting MAY ignore the sort specification. An implementation that  
186 partially supports sorting, SHOULD make best effort to satisfy the sort criteria, but need not adhere to it literally.  
187 Such implementations MAY ignore LDAP control-criticality specifications regarding the sort control.

188 Sorting is not supported for subscriptions.

189 8. All DST methods are defined and the default method support policy applies.

190 9. All DST-specified discovery option keywords are supported with the semantics defined in [\[LibertyDST\]](#).

191 All discovery option keywords specified in [\[LibertySUBS\]](#) are supported.

192 ID-SIS-DAP also supports the following additional discovery option keywords to indicate the supported ID-WSF  
193 framework and DST version(s):

```
194 urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf1.1:dst1.1  
195 urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf1.1:dst2.1  
196 urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf2.0:dst1.1  
197 urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf2.0:dst2.1
```

198  
199

200 10. Element uniqueness mechanisms for objects of type "\_Subscription" are specified in [LibertySUBS] (i.e.,  
201 subscriptionID).

202 For objects of type "entry," *distinguished name* provides a unique identifier.

203 11. The default policy for protocol extension applies.

204 For objects of type "entry," no special extension mechanism is needed. ID-SIS-DAP does not specify any schema  
205 so all entries might as well belong to LDAP `objectClass` "extensibleObject."

206 While ID-SIS-DAP as protocol does not mandate any schema, or mechanism for schema validation, the  
207 underlying backend MAY, in fact, be schema-aware and perform schema validation in the way usual for directory  
208 servers. The agreement about the schema between WSC and WSP is out-of-band.

209 12. Query features (default feature support policy applies):

210 a. Testing **MUST** be supported.

211 b. **<ResultQuery>** **MUST** be supported.

212 c. Sorting is **OPTIONAL** (typically, the backend would have to implement [RFC2891]). If sorting is not  
213 supported for `objectType` "entry," discovery option keyword `urn:liberty:dst:noSorting` **MUST** be  
214 registered.

215 Sorting need not be supported for `objectType` "\_Subscription" and the lack of such support is **NOT**  
216 reflected in discovery option keyword registrations.

217 d. Pagination of results is **OPTIONAL** (typically, the backend would have to implement [RFC2696]  
218 or [LDAPVLV]). If pagination is not supported for `objectType` "entry," discovery option keyword  
219 `urn:liberty:dst:noPagination` **MUST** be registered.

220 Pagination need not be supported for `objectType` "\_Subscription" and the lack of such support is **NOT**  
221 reflected in discovery option keyword registrations.

222 e. Support static sets in pagination is **OPTIONAL** (the backend probably would have to support [LDAPVLV]).  
223 If static sets are not supported, discovery option keyword `urn:liberty:dst:noStatic` **MUST** be  
224 registered.

225 f. Multiple **<Query>** elements **MUST** be supported. No transactional semantic is attached to this construct.

226 g. Multiple **<QueryItem>** elements **MUST** be supported. No transactional semantic is attached to this  
227 construct.

228 h. Multiple **<TestItem>** elements **MUST** be supported. No transactional semantic is attached to this construct.

229 i. `changedSince` **MUST** be supported in **<ResultQuery>** and **<QueryItem>**, but the granularity of the  
230 change detection **MAY** be one directory entry. **<ChangeFormat>** `CurrentElements` **MUST** be supported  
231 and is interpreted as returning the entire changed directory entry, possibly projected using the attribute  
232 list supplied in the query. **<ChangeFormat>** specifications `ChangedElements` and `All` **MUST NOT** be  
233 used. If **<ChangeFormat>** is omitted, `CurrentElements` is assumed. Since there is only one possible  
234 **<ChangeFormat>**, we **RECOMMENDED** that **<ChangeFormat>** is always omitted.

235 j. `includeCommonAttributes` **MUST** be supported.

236 13. Create features:

237 a. Multiple **<Create>** elements **MUST** be supported. No transactional semantic is attached to this construct.

238 14. Delete features:



239 a. Multiple **<Delete>** elements MUST be supported. No transactional semantic is attached to this construct.

240 15. Modify features:

241 a. Multiple **<Modify>** elements MUST be supported. No transactional semantic is attached to this construct.

242 b. Multiple **<ModifyItem>** elements MUST be supported. No transactional semantic is attached to this  
243 construct.

244 c. Multiple **<ModifyItem>** elements may fail independently. No atomic modify or transactional semantic  
245 needs to be supported.

246 d. `notChangedSince` MUST be supported, but only at directory entry granularity.

## 247 2.2. Subscription Parameters

248 In Section 7 of [[LibertySUBS](#)], a checklist is provided for service specifications. This section corresponds, item for  
249 item, to that checklist.

250 1. Standard names are used. See [Section 7](#).

251 2. The **<Subscription>** elements represent objects of type "\_Subscription" in `AppDataType` with schema as  
252 specified in [[LibertySUBS](#)].

253 3. If `objectType` is "\_Subscription," then the **<dn>** string is an XPath expression [[XPATH](#)] pointing to the  
254 subscription object and NO other elements or XML attributes MUST BE specified. Default restriction on XPaths  
255 for subscriptions applies.

256 4. `TriggerType` is not used. The only way to trigger notifications is by change of data.

257 5. `AggregationType` is not used. The underlying backend MAY use any aggregation policy it chooses.

258 6. Extension mechanisms for objects of type "\_Subscription" are specified in [[LibertySUBS](#)].

259 7. How subscriptions can be established and manipulated:

260 a. Supporting `objectType` "\_Subscription" is OPTIONAL (typically, the backend would have  
261 to support [[LDAPPSearch](#)]). If subscription is not supported, the discovery option keyword  
262 `urn:liberty:dst:noSubscribe` MUST be registered.

263 b. Subscribing in **<Query>** MUST be supported if `objectType` "\_Subscription" is supported. However,  
264 subscriptions to objects of type "\_Subscription" need not be supported.

265 c. Multiple **<Subscription>** elements in **<Query>** MUST be supported if `objectType` "\_Subscription" is  
266 supported.

267 d. Subscribing in **<Create>** MUST be supported if `objectType` "\_Subscription" is supported. However,  
268 subscriptions to objects of type "\_Subscription" need not be supported.

269 e. Multiple **<Subscription>** elements in **<Create>** MUST be supported if `objectType` "\_Subscription" is  
270 supported.

271 f. Subscribing in **<Modify>** MUST be supported if `objectType` "\_Subscription" is supported. However,  
272 subscriptions to objects of type "\_Subscription" need not be supported.

273 g. Multiple **<Subscription>** elements in **<Modify>** MUST be supported if `objectType` "\_Subscription" is  
274 supported.

- 
- 275 8. `starts` XML attribute **MUST** be supported for specification of the start of a subscription if `objectType`  
276 `"_Subscription"` is supported.
- 277 9. If `objectType` `"_Subscription"` is supported, it **MUST** be possible to expire subscriptions by at least two  
278 methods:
- 279 a. by expiration of credentials,
- 280 b. by specification of the `expires` XML attribute, and omission of `expires` implies expiration when  
281 credentials expire.
- 282 Expiration will happen at the earlier of the occasions specified by (a) and (b).
- 283 10. Use of same value for the `expires` and `starts` XML attributes **MUST** be supported to request one notification  
284 message at a specified time if `objectType` `"_Subscription"` is supported.
- 285 11. Querying existing subscriptions **SHOULD** be supported.
- 286 12. Notifications **SHOULD** be acknowledged.

### 3. Mapping to LDAP

Generally, ID-WSF-conveyed identity will map to the LDAP *distinguished name* as described in the `SelectType` discussion, above. In results, the `dn` attribute SHOULD be substituted by a dummy value such as "liberty-identity-based-dn" if any identifiable information is available from it. Access controls keyed on the WSC's identification should be applied to other attribute data to achieve deployment-dependent privacy goals.

ID-SIS-DAP `<ResultQuery>`, `<QueryItem>`, and `<TestItem>` map to LDAP *Search* operation in a straight forward way. `<TestItem>` can, in some cases, be mapped to the LDAP *Compare* operation. `<Sort>` can be mapped to the LDAP control described in [RFC2891]. Pagination functionality (`count` and `offset` XML attributes) can be mapped to either [RFC2696] or [LDAPVLV]. The static set functionality (`setID` and `setReq` XML attributes) can be mapped to [LDAPVLV].

ID-SIS-DAP `<CreateItem>` maps straight to the LDAP *Add* operation. The LDIF format data in the `<LDIF>` container in the `<NewData>` container will specify the distinguished names of the new entries.

ID-SIS-DAP `<DeleteItem>` maps straight to the LDAP *Delete* operation.

LDAP *Bind* operation does not have direct mapping, but the identity information conveyed by the ID Web Services Framework should allow the ID-SIS-DAP implementation to perform a *Bind* to authenticate, correctly, the Principal to the directory server. Since *Bind* normally uses a password or SASL authentication mechanism and since it would be ill-advised for the ID-SIS-DAP implementation to know the credentials, it may be more convenient to use the LDAP control for proxy authentication [LDAPproxyauth]. Another alternative would be to develop a SASL mechanism that would somehow allow ID credentials that were received from the ID-WSF layer to be passed to the directory server.

There is no mapping for the LDAP *Unbind*, *Abandon*, or *Extended* operations. No requirement for such mapping is known.

There is no mapping for the LDAP *Modify RDN* (rename) operation.

ID-SIS-DAP `<ModifyItem>` maps to the LDAP *Modify* operation, but the mapping is not straight forward. The following rules describe the mapping.

1. The `<LDIF>` container in `<NewData>` MUST contain attribute data for one entry, however, this data may be empty.
2. The distinguished name in LDIF format data that is provided in the `<LDIF>` container in the `<NewData>` container MUST be ignored and MAY be entirely omitted.
3. Distinguished name for LDAP *Modify* operation MUST be taken from ID-WSF-conveyed identity combined with the `<Select>` element of the `<ModifyItem>`.
4. The LDIF format data is interpreted as descriptive of the changes to be applied as described in [RFC2849].
5. The *changeType* SHOULD be "Modify." However, other *changeTypes* are permitted for DST 1.1-based implementations.
6. Any number of *add*, *delete*, and *replace* specifications can be specified to effectuate the modification as described in [RFC2849].
7. `overrideAllowed` XML attribute MUST NOT be supplied in `<ModifyItem>` and all DST processing rules regarding it are void.

## 4. Processing Rules and Other Considerations

### 4.1. Query Is Not Required to Report Same Data to Repeated Queries

An ID-SIS-DAP instance is NOT REQUIRED to report the same results to two instances of the same query. An ID-SIS-DAP instance SHOULD report the same results to the same query made by the same client unless an update (Create, Delete, Modify, or out-of-band) has occurred in the interim. An ID-SIS-DAP instance MAY use the Interaction Service protocol [[LibertyInteract11](#)] or out-of-band means to determine the data to return.

Data to be returned in response to a query is determined by the ID-SIS-DAP provider as guided by its policies, the permissions the Principal has set, and the interaction with the Principal. Clients should use the data based on the data's semantic meaning as specified here and further qualified by the `acc` (Attribute Collection Context) XML attributes [[LibertyDST](#)] that may be present in the query response. A client SHOULD NOT attempt to use ID-SIS-DAP as a transparent data store as there may be multiple updates, permission, and policy reasons that impede the transparency.

### 4.2. Simulation (dry-run) is Required

The simulation method using the `ProcessingContext` header with value `"urn:liberty:sb:2003-08:ProcessingContext:Simulate,"` as specified in [[LibertySOAPBinding](#)], MUST be supported. If simulated operation succeeds, similar actual operation SHOULD have a high probability of succeeding within next 30 minutes.

This feature allows a WSC to test whether a modification is plausible prior to invoking the user interface to query data from the Principal, thus avoiding principal-supplied nonactionable data unnecessarily.

#### LDAP-specific guidance

It is recommend that the simulation is actually implemented as real create or modify operations with dummy data, possibly followed by deletion of the dummy data. While this is expensive, it gives the degree of guarantee that is the intent of explicitly invoking the simulation run.

The "real modify" approach may not be viable if the modify would destroy old data. In such cases, the implementation may need to use another substitute operation. Generally, the write-ability of the entry (or even the existence of it in the case of Modify) and the validity of the schema are what needs to be established - the dry-run operation has to be sufficiently realistic to establish this.

### 4.3. Use of Usage Directives

When querying a WSC, the query MUST include usage directives that indicate the reason for the request and the type of use to which the data will be put. If the ID-SIS-DAP WSP invokes the interaction service, it SHOULD display this information, usually after localization, to the Principal.

The actual format of the usage directives MUST be agreed out of band, e.g., in a business agreement between the WSC and the ID-SIS-DAP WSP.

### 4.4. Granularity of Metadata

Since it may be cumbersome to remember metadata on a "per attribute" basis when using an LDAP backend, the following relaxations are made:

1. XML attributes in a `localizedLeafAttributes` XML attribute group apply to the entire contents of the `<LDIF>` container, whether returned by `<Query>` or supplied as `<NewData>` to `<Create>` or `<Modify>`.
2. `modificationTime` MAY be computed and `notChangedSince` evaluated according to the last change to any attribute in the entry.

## 362 **4.5. Subscription to Change in an Attribute**

363 If subscription is established by specifying a specific list of attributes, then the notification is triggered only if one of  
364 the specified attributes changes. If no specific attributes are specified, then change to any attribute in the entry triggers  
365 the notification.

## 5. DST 1.1 Support

This list documents the limitations and modifications to this specification if DST 1.1 [[LibertyDST11](#)] is used. Mostly these reflect lesser functionality of the 1.1 version.

1. The service type and namespace **MUST** be changed from "urn:liberty:id-sis-dap:2006-08:dst-2.1" to "urn:liberty:id-sis-dap:2006-08:dst-1.1" to reflect that the DST 1.1-based service is a different version than the DST 2.1-based service (checklist 1).
2. **<ResourceID>** element **MUST** be omitted. Instead of the resource IDs, the identification of the principal **MUST** be done using a security mechanism that carries a SAML assertion (any version) whose statements have all the same subjects and that this subject identifies the resource to be accessed. Essentially, this adopts the ID-WSF 2.0 model for all cases. This implies that there **MUST** be a federation for the principal between the ID-SIS-DAP server and IdP that was used to start the session involving ID-SIS-DAP access.
3. The test feature is undefined (checklist 12.a)
4. The **<ResultQuery>** is undefined (checklist 12.b)
5. The optional pagination feature (checklist 12.d) is undefined and the corresponding discovery option keyword **MUST** be registered.
6. The optional subscription feature is undefined and the corresponding discovery option keyword **MUST** be registered.
7. The concept of object type does not exist. All objects **MUST** be of type "entry."
8. The **<Create>** and **<Delete>** methods do not exist. Their function is accomplished using **<Modify>** and features of LDIF format.
9. The ID-SIS-DAP schema is modified as follows
  - a. The target namespace and dap namespace are changed from "urn:liberty:id-sis-dap:2006-08:dst-2.1" to "urn:liberty:id-sis-dap:2006-08:dst-1.1."
  - b. The include statement for "liberty-idwsf-dst-dt-v2.1.xsd" is changed to include "liberty-idwsf-dst-dt-v1.1.xsd" instead.
  - c. The include statement for "liberty-idwsf-dst-v2.1.xsd" is changed to include "liberty-idwsf-dst-v1.1.xsd" instead.

## 6. Examples

The following examples illustrate some ID-SIS-DAP protocol exchanges and the ID-WSF SOAP binding around them. These examples are not normative.

### 6.1. ID-SIS-DAP Query Request

This query queries for the specific attribute JAVAPROFILE from the LDAP entry associated with the Name ID PEX7\_Y-G\_hMaysCNJmaQj as specified in the assertion used as credential for TLS:Bearer security mechanism. This message would appear as the body of an https request.

```
393 <soap:Envelope
394   xmlns:lib="urn:liberty:iff:2003-08"
395   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
396   <soap:Header>
397     <wsa:MessageID
398       xmlns:wsa="http://www.w3.org/2005/03/addressing"
399       id="#MID">uuid:6g5n02-nzY8vJYwK2uCW</wsa:MessageID>
400     <wsa:To
401       xmlns:wsa="http://www.w3.org/2005/03/addressing">
402       https://g-sp.liberty-iop.org:8843/DAP-PSBEARER</wsa:To>
403     <wsa:Action xmlns:wsa="http://www.w3.org/2005/03/addressing"/>
404     <wsa:ReplyTo xmlns:wsa="http://www.w3.org/2005/03/addressing">
405       <wsa:Address>http://www.w3.org/2005/03/addressing/role/anonymous</wsa:Address>
406     </wsa:ReplyTo>
407     <wsse:Security
408       xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200
409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453
454       <sa:Assertion
455         xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
456         ID="CRED1WqsZ7VozwEo6TBpkIMV"
457         IssueInstant="2005-10-03T16:58:33Z"
458         Version="2.0">
459         <sa:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
460           https://g-ds.liberty-iop.org:8681/idp.xml</sa:Issuer>
461         <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
462           <ds:SignedInfo>
463             <ds:CanonicalizationMethod
464               Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
465             <ds:SignatureMethod
466               Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
467             <ds:Reference URI="#CRED1WqsZ7VozwEo6TBpkIMV">
468               <ds:Transforms>
469                 <ds:Transform
470                   Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
471                 <ds:Transform
472                   Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
473               </ds:Transforms>
474               <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
475               <ds:DigestValue>BA5GsDSlCmzvCTkNoBORaSCP9kI=</ds:DigestValue>
476             </ds:Reference>
477           </ds:SignedInfo>
478           <ds:SignatureValue>
479             Y+kMa0ToVrbG8smohquiGyeMrKM5Dw4aizPszOQ6hDWODvpH/q0eIow7PIFzvoN+7XUQbkZ
480             aA0J4KXEyzeRtOrt5XrhOzg6d2BCnKSHZXHwCE7d6qa5NJGtQlszOjMtwTcMByNKMWdcuksQdryGsklNkED6ZPEFpn
481             xK4aKs57Rs=
482           </ds:SignatureValue>
483         </ds:Signature>
484         <sa:Subject>
485           <sa:NameID
486             Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
487             NameQualifier="https://g-ds.liberty-iop.org:8681/idp.xml">PEX7_Y-G_hMaysCNJmaQj
488           </sa:NameID>
489           <sa:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer" />
490         </sa:Subject>

```

```
454     <sa:Conditions
455         NotBefore="2005-10-03T16:53:33Z"
456         NotOnOrAfter="2005-10-03T17:08:33Z">
457         <sa:AudienceRestriction>
458             <sa:Audience>https://g-wsc.liberty-iop.org:8643/sp.xml</sa:Audience>
459         </sa:AudienceRestriction>
460     </sa:Conditions>
461     <sa:AuthnStatement AuthnInstant="2005-10-03T16:58:33Z">
462         <sa:AuthnContext>
463             <sa:AuthnContextClassRef>
464                 urn:oasis:names:tc:SAML:2.0:ac:classes:Password
465             </sa:AuthnContextClassRef>
466         </sa:AuthnContext>
467     </sa:AuthnStatement>
468 </sa:Assertion>
469 </wsse:Security>
470 <sb2:Timestamp xmlns:sb2="urn:liberty:sb:2004-12">
471     <wsu:Created
472         xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecur
473 ity-utility-1.0.xsd">
474         2005-10-03T16:58:33Z
475     </wsu:Created>
476 </sb2:Timestamp>
477 <sb2:Sender xmlns:sb2="urn:liberty:sb:2004-12"
478     providerID="https://g-wsc.liberty-iop.org:8643/sp.xml"
479     soap:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
480 </soap:Header>
481 <soap:Body id="bdy">
482     <dap:Query
483         xmlns:dap="urn:liberty:id-sis-dap:2006-08:dst-2.1"
484         id="id:CoS9jNRrwxecrm_TEFqU"
485         itemID="itm:S9fdkoCcS8_PoRAhvRlt">
486         <dap:QueryItem
487             id="id:0nli3-cRcCl5WolnEfaf"
488             itemID="itm:UsMupiJjpt1CN4PkRzyQ"
489             objectType="entry">
490             <dap:Select attributes="JAVAPROFILE"/>
491         </dap:QueryItem>
492     </dap:Query>
493 </soap:Body>
494 </soap:Envelope>
495
496
```

## 497 6.2. ID-SIS-DAP QueryResponse

498 This is a response to the query in the previous example. This illustrates how the distinguished name is blanked out with  
499 value liberty-identity-based-dn to avoid leaking privacy-sensitive information that may have been present in  
500 the dn. The requested attribute has value J2ME. This message would appear as the body of an https response.

```
501 <soap:Envelope
502     xmlns:lib="urn:liberty:iff:2003-08"
503     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
504     <soap:Header>
505         <wsa:MessageID
506             xmlns:wsa="http://www.w3.org/2005/03/addressing"
507             id="#MID">uuid:eg4ZFqQi3PmHSS EUdg0Q</wsa:MessageID>
508         <wsa:RelatesTo
509             xmlns:wsa="http://www.w3.org/2005/03/addressing">uuid:6g5n02-nzY8vJYwK2uCW
510         </wsa:RelatesTo>
511         <wsa:To xmlns:wsa="http://www.w3.org/2005/03/addressing">
512             http://www.w3.org/2005/03/addressing/role/anonymous</wsa:To>
513         <wsa:Action xmlns:wsa="http://www.w3.org/2005/03/addressing"/>
514         <wsa:ReplyTo xmlns:wsa="http://www.w3.org/2005/03/addressing">
515             <wsa:Address>http://www.w3.org/2005/03/addressing/role/anonymous</wsa:Address>
```



```
516     </wsa:ReplyTo>
517     <sb2:Timestamp xmlns:sb2="urn:liberty:sb:2004-12">
518         <wsu:Created
519             xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-w
520 ssecurity-utility-1.0.xsd">
521             2005-10-03T16:58:34Z</wsu:Created>
522         </sb2:Timestamp>
523     <sb2:Sender
524         xmlns:sb2="urn:liberty:sb:2004-12"
525         providerID="https://g-sp.liberty-iop.org:8843/sp.xml"
526         soap:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
527 </soap:Header>
528 <soap:Body>
529     <dap:QueryResponse
530         xmlns:dap="urn:liberty:id-sis-dap:2006-08:dst-2.1"
531         id="dapid:aM7j4DhGRwZ69dsQPWvh"
532         itemIDRef="itm:S9fdkoCcS8_PoRAhvRlt">
533         <dap:Status code="OK"/>
534         <dap:Data id="dapdataid:kc430k3KmG3Ju79KKFc6">
535             dn: liberty-identity-based-dn JAVAPROFILE: J2ME
536         </dap:Data>
537     </dap:QueryResponse>
538 </soap:Body>
539 </soap:Envelope>
540
541
```

## 542 7. Schema for ID-SIS-DAP

### 543 7.1. Summary of ID-SIS-DAP

544 The ID-SIS-DAP schema may be summarized as follows.

```

545 target(dap,      urn:liberty:id-sis-dap:2006-08:dst-2.1)
546 import(dst,      urn:liberty:dst:2006-08,      liberty-idwsf-dst-v2.1.xsd)
547 import(subs,     urn:liberty:ssos:2006-08,     liberty-idwsf-subsv1.0.xsd)
548 import(subsref, urn:liberty:ssos:2006-08:ref,  liberty-idwsf-subsv1.0.xsd)
549 import(lu,       urn:liberty:util:2006-08,     liberty-idwsf-utility-v2.0.xsd)
550
551 #sec(methods)
552 Create      -> %subsref:CreateType
553 CreateResponse -> %subsref:CreateResponseType
554 Query       -> %subsref:QueryType
555 QueryResponse -> %subsref:QueryResponseType
556 Modify      -> %subsref:ModifyType
557 ModifyResponse -> %subsref:ModifyResponseType
558 Delete      -> %subsref>DeleteType
559 DeleteResponse -> %subsref>DeleteResponseType
560 Notify      -> %subsref:NotifyType
561 NotifyResponse -> %subsref:NotifyResponseType
562 #endsec(methods)
563
564 # =====
565
566 #sec(redefs)
567 %SelectType:
568   dn?          -> %xs:string
569   filter?      -> %xs:string
570   @scope?      -> %xs:integer default(0)
571   @sizelimit?  -> %xs:integer default(0)
572   @timelimit?  -> %xs:integer default(0)
573   @attributes? -> %xs:string
574   @typesonly?  -> %xs:boolean default(false)
575   @derefaliases? -> %xs:integer default(0)
576   ;
577
578 %TestOpType:   base(dap:SelectType) ;
579 %SortType:     base(xs:string) ;
580 %TriggerType:  base(xs:string) ;
581 %AggregationType: base(xs:string) ;
582
583 %AppDataType:
584   dap:LDIF
585   | dap:Subscription
586   ;
587
588 LDIF: base(xs:string)
589   &@dst:localizedLeafAttributes
590   ;
591 #endsec(redefs)
592
593 # =====
594
595 #sec(create)
596 %CreateType:   base(dst:RequestType)
597   dap:Subscription*
598   dap:CreateItem+
599   dap:ResultQuery*
600   ;
601
602 CreateItem      -> %dap:CreateItemType
603 %CreateItemType:
604   dap:NewData?

```

```
605   &@dst:CreateItemAttributeGroup
606   ;
607
608 NewData          -> %dap:AppDataType
609
610 %CreateResponseType: base(dap:DataResponseType) ;
611 %DataResponseType:  base(dst:DataResponseBaseType)
612   dap:ItemData*
613   ;
614 #endsec(create)
615
616 # =====
617
618 #sec(query)
619 %QueryType:        base(dst:RequestType)
620   dap:TestItem*
621   dap:QueryItem*
622   dap:Subscription*
623   ;
624
625 TestItem         -> %dap:TestItemType
626 %TestItemType:    base(dst:TestItemBaseType)
627   TestOp?        -> %dap:TestOpType
628   ;
629
630 QueryItem        -> %dap:QueryItemType
631 %QueryItemType:   base(dap:ResultQueryType)
632   &@dst:PaginationAttributeGroup
633   ;
634 #endsec(query)
635
636 #sec(queryresp)
637 %QueryResponseType: base(dst:DataResponseBaseType)
638   dst:TestResult*
639   dap:Data*
640   ;
641
642 Data             -> %dap:DataType
643 %DataType:        base(dap:ItemDataType)
644   &@dst:PaginationResponseAttributeGroup
645   ;
646 #endsec(queryresp)
647
648 # =====
649
650 #sec(mod)
651 %ModifyType:      base(dst:RequestType)
652   dap:Subscription*
653   dap:ModifyItem+
654   dap:ResultQuery*
655   ;
656
657 ModifyItem       -> %dap:ModifyItemType
658 %ModifyItemType:
659   dap:Select?
660   dap:NewData?
661   &@dst:ModifyItemAttributeGroup
662   ;
663
664 %ModifyResponseType: base(dap:DataResponseType) ;
665 #endsec(mod)
666
667 # =====
668
669 #sec(del)
670 %DeleteType:      base(dst:RequestType)
671   dap:DeleteItem+
```

```

672 ;
673
674 DeleteItem      -> %dap:DeleteItemType
675 %DeleteItemType: base(dst:DeleteItemBaseType)
676   dap:Select?
677 ;
678
679 %DeleteResponseType: base(lu:ResponseType) ;
680 #endsec(del)
681
682 # =====
683
684 #sec(resqry)
685 Select          -> %dap:SelectType
686
687 ResultQuery     -> %dap:ResultQueryType
688 %ResultQueryType: base(dst:ResultQueryBaseType)
689   dap:Select?
690   Sort?        -> %dap:SortType
691 ;
692
693 ItemData        -> %dap:ItemDataType
694 %ItemDataType:  base(dap:AppDataType)
695   &@dst:ItemDataAttributeGroup
696 ;
697 #endsec(resqry)
698
699 #sec(subscr)
700 Subscription     -> %dap:SubscriptionType
701 %SubscriptionType: base(subs:SubscriptionType)
702   dap:ResultQuery*
703   Aggregation?  -> %dap:AggregationType
704   Trigger?      -> %dap:TriggerType
705 ;
706 #endsec(subscr)
707
708 #sec(notif)
709 %NotifyType:    base(dst:RequestType)
710   dap:Notification*
711   &@subs:NotifyAttributeGroup
712 ;
713
714 Notification     -> %dap:NotificationType
715 %NotificationType: base(subs:NotificationType)
716   dap:ItemData*
717 ;
718
719 %NotifyResponseType: base(subs:NotifyResponseType) ;
720 #endsec(notif)
721
722 #EOF
723

```

## 7.2. Formal XML Schema for ID-SIS-DAP

The formal ID-SIS-DAP schema follows.

```

726 <?xml version="1.0" encoding="UTF-8"?>
727 <xs:schema
728   targetNamespace="urn:liberty:id-sis-dap:2006-08:dst-2.1"
729   xmlns:dap="urn:liberty:id-sis-dap:2006-08:dst-2.1"
730   xmlns:dst="urn:liberty:dst:2006-08"
731   xmlns:subs="urn:liberty:ssos:2006-08"
732   xmlns:subsref="urn:liberty:ssos:2006-08:ref"
733   xmlns:lu="urn:liberty:util:2006-08"

```

```

734     xmlns:xs="http://www.w3.org/2001/XMLSchema"
735     elementFormDefault="qualified"
736     attributeFormDefault="unqualified">
737     <xs:import namespace="urn:liberty:dst:2006-08"
738       schemaLocation="liberty-idwsf-dst-v2.1.xsd"/>
739     <xs:import namespace="urn:liberty:ssos:2006-08"
740       schemaLocation="liberty-idwsf-subsv1.0.xsd"/>
741     <xs:import namespace="urn:liberty:ssos:2006-08:ref"
742       schemaLocation="liberty-idwsf-subsv1.0.xsd"/>
743     <xs:import namespace="urn:liberty:util:2006-08"
744       schemaLocation="liberty-idwsf-utility-v2.0.xsd"/>
745     <!--sec(methods)-->
746     <xs:element name="Create" type="subref:CreateType"/>
747     <xs:element name="CreateResponse" type="subref:CreateResponseType"/>
748     <xs:element name="Query" type="subref:QueryType"/>
749     <xs:element name="QueryResponse" type="subref:QueryResponseType"/>
750     <xs:element name="Modify" type="subref:ModifyType"/>
751     <xs:element name="ModifyResponse" type="subref:ModifyResponseType"/>
752     <xs:element name="Delete" type="subref>DeleteType"/>
753     <xs:element name="DeleteResponse" type="subref>DeleteResponseType"/>
754     <xs:element name="Notify" type="subref:NotifyType"/>
755     <xs:element name="NotifyResponse" type="subref:NotifyResponseType"/>
756     <!--endsec(methods)-->
757     <!--sec(redefs)-->
758     <xs:complexType name="SelectType">
759       <xs:sequence>
760         <xs:element name="dn" minOccurs="0" maxOccurs="1" type="xs:string"/>
761         <xs:element name="filter" minOccurs="0" maxOccurs="1" type="xs:string"/>
762       </xs:sequence>
763       <xs:attribute name="scope" use="optional" type="xs:integer" default="0"/>
764       <xs:attribute name="sizelimit" use="optional" type="xs:integer" default="0"/>
765       <xs:attribute name="timelimit" use="optional" type="xs:integer" default="0"/>
766       <xs:attribute name="attributes" use="optional" type="xs:string"/>
767       <xs:attribute name="typesonly" use="optional" type="xs:boolean" default="false"/>
768       <xs:attribute name="dereferaliases" use="optional" type="xs:integer" default="0"/>
769     </xs:complexType>
770     <xs:complexType name="TestOpType">
771       <xs:complexContent>
772         <xs:extension base="dap:SelectType"/>
773       </xs:complexContent>
774     </xs:complexType>
775     <xs:complexType name="SortType">
776       <xs:simpleContent>
777         <xs:extension base="xs:string"/>
778       </xs:simpleContent>
779     </xs:complexType>
780     <xs:complexType name="TriggerType">
781       <xs:simpleContent>
782         <xs:extension base="xs:string"/>
783       </xs:simpleContent>
784     </xs:complexType>
785     <xs:complexType name="AggregationType">
786       <xs:simpleContent>
787         <xs:extension base="xs:string"/>
788       </xs:simpleContent>
789     </xs:complexType>
790     <xs:complexType name="AppDataType">
791       <xs:choice>
792         <xs:element ref="dap:LDIF"/>
793         <xs:element ref="dap:Subscription"/>
794       </xs:choice>
795     </xs:complexType>
796     <xs:element name="LDIF">
797       <xs:complexType>
798         <xs:simpleContent>
799           <xs:extension base="xs:string">
800             <xs:attributeGroup ref="dst:localizedLeafAttributes"/>

```

```

801     </xs:extension>
802     </xs:simpleContent>
803     </xs:complexType>
804 </xs:element>
805 <!--endsec(redefs)-->
806 <!--sec(create)-->
807     <xs:complexType name="CreateType">
808         <xs:complexContent>
809             <xs:extension base="dst:RequestType">
810                 <xs:sequence>
811                     <xs:element ref="dap:Subscription" minOccurs="0" maxOccurs="unbounded"/>
812                     <xs:element ref="dap:CreateItem" minOccurs="1" maxOccurs="unbounded"/>
813                     <xs:element ref="dap:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
814                 </xs:sequence>
815             </xs:extension>
816         </xs:complexContent>
817     </xs:complexType>
818     <xs:element name="CreateItem" type="dap:CreateItemType"/>
819     <xs:complexType name="CreateItemType">
820         <xs:sequence>
821             <xs:element ref="dap:NewData" minOccurs="0" maxOccurs="1"/>
822         </xs:sequence>
823         <xs:attributeGroup ref="dst:CreateItemAttributeGroup"/>
824     </xs:complexType>
825     <xs:element name="NewData" type="dap:AppDataType"/>
826     <xs:complexType name="CreateResponseType">
827         <xs:complexContent>
828             <xs:extension base="dap:DataResponseType"/>
829         </xs:complexContent>
830     </xs:complexType>
831     <xs:complexType name="DataResponseType">
832         <xs:complexContent>
833             <xs:extension base="dst:DataResponseBaseType">
834                 <xs:sequence>
835                     <xs:element ref="dap:ItemData" minOccurs="0" maxOccurs="unbounded"/>
836                 </xs:sequence>
837             </xs:extension>
838         </xs:complexContent>
839     </xs:complexType>
840 <!--endsec(create)-->
841 <!--sec(query)-->
842     <xs:complexType name="QueryType">
843         <xs:complexContent>
844             <xs:extension base="dst:RequestType">
845                 <xs:sequence>
846                     <xs:element ref="dap:TestItem" minOccurs="0" maxOccurs="unbounded"/>
847                     <xs:element ref="dap:QueryItem" minOccurs="0" maxOccurs="unbounded"/>
848                     <xs:element ref="dap:Subscription" minOccurs="0" maxOccurs="unbounded"/>
849                 </xs:sequence>
850             </xs:extension>
851         </xs:complexContent>
852     </xs:complexType>
853     <xs:element name="TestItem" type="dap:TestItemType"/>
854     <xs:complexType name="TestItemType">
855         <xs:complexContent>
856             <xs:extension base="dst:TestItemBaseType">
857                 <xs:sequence>
858                     <xs:element name="TestOp" minOccurs="0" maxOccurs="1" type="dap:TestOpType"/>
859                 </xs:sequence>
860             </xs:extension>
861         </xs:complexContent>
862     </xs:complexType>
863     <xs:element name="QueryItem" type="dap:QueryItemType"/>
864     <xs:complexType name="QueryItemType">
865         <xs:complexContent>
866             <xs:extension base="dap:ResultQueryType">
867                 <xs:attributeGroup ref="dst:PaginationAttributeGroup"/>

```

```

868     </xs:extension>
869   </xs:complexContent>
870 </xs:complexType>
871 <!--endsec(query)-->
872 <!--sec(queryresp)-->
873 <xs:complexType name="QueryResponseType">
874   <xs:complexContent>
875     <xs:extension base="dst:DataResponseBaseType">
876       <xs:sequence>
877         <xs:element ref="dst:TestResult" minOccurs="0" maxOccurs="unbounded"/>
878         <xs:element ref="dap:Data" minOccurs="0" maxOccurs="unbounded"/>
879       </xs:sequence>
880     </xs:extension>
881   </xs:complexContent>
882 </xs:complexType>
883 <xs:element name="Data" type="dap:DataType"/>
884 <xs:complexType name="DataType">
885   <xs:complexContent>
886     <xs:extension base="dap:ItemDataType">
887       <xs:attributeGroup ref="dst:PaginationResponseAttributeGroup"/>
888     </xs:extension>
889   </xs:complexContent>
890 </xs:complexType>
891 <!--endsec(queryresp)-->
892 <!--sec(mod)-->
893 <xs:complexType name="ModifyType">
894   <xs:complexContent>
895     <xs:extension base="dst:RequestType">
896       <xs:sequence>
897         <xs:element ref="dap:Subscription" minOccurs="0" maxOccurs="unbounded"/>
898         <xs:element ref="dap:ModifyItem" minOccurs="1" maxOccurs="unbounded"/>
899         <xs:element ref="dap:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
900       </xs:sequence>
901     </xs:extension>
902   </xs:complexContent>
903 </xs:complexType>
904 <xs:element name="ModifyItem" type="dap:ModifyItemType"/>
905 <xs:complexType name="ModifyItemType">
906   <xs:sequence>
907     <xs:element ref="dap:Select" minOccurs="0" maxOccurs="1"/>
908     <xs:element ref="dap:NewData" minOccurs="0" maxOccurs="1"/>
909   </xs:sequence>
910   <xs:attributeGroup ref="dst:ModifyItemAttributeGroup"/>
911 </xs:complexType>
912 <xs:complexType name="ModifyResponseType">
913   <xs:complexContent>
914     <xs:extension base="dap:DataResponseType"/>
915   </xs:complexContent>
916 </xs:complexType>
917 <!--endsec(mod)-->
918 <!--sec(del)-->
919 <xs:complexType name="DeleteType">
920   <xs:complexContent>
921     <xs:extension base="dst:RequestType">
922       <xs:sequence>
923         <xs:element ref="dap>DeleteItem" minOccurs="1" maxOccurs="unbounded"/>
924       </xs:sequence>
925     </xs:extension>
926   </xs:complexContent>
927 </xs:complexType>
928 <xs:element name="DeleteItem" type="dap>DeleteItemType"/>
929 <xs:complexType name="DeleteItemType">
930   <xs:complexContent>
931     <xs:extension base="dst>DeleteItemBaseType">
932       <xs:sequence>
933         <xs:element ref="dap>Select" minOccurs="0" maxOccurs="1"/>
934       </xs:sequence>

```

```

935     </xs:extension>
936   </xs:complexContent>
937 </xs:complexType>
938 <xs:complexType name="DeleteResponseType">
939   <xs:complexContent>
940     <xs:extension base="lu:ResponseType" />
941   </xs:complexContent>
942 </xs:complexType>
943 <!--endsec(del)-->
944 <!--sec(resqry)-->
945 <xs:element name="Select" type="dap:SelectType" />
946 <xs:element name="ResultQuery" type="dap:ResultQueryType" />
947 <xs:complexType name="ResultQueryType">
948   <xs:complexContent>
949     <xs:extension base="dst:ResultQueryBaseType">
950       <xs:sequence>
951         <xs:element ref="dap:Select" minOccurs="0" maxOccurs="1" />
952         <xs:element name="Sort" minOccurs="0" maxOccurs="1" type="dap:SortType" />
953       </xs:sequence>
954     </xs:extension>
955   </xs:complexContent>
956 </xs:complexType>
957 <xs:element name="ItemData" type="dap:ItemDataType" />
958 <xs:complexType name="ItemDataType">
959   <xs:complexContent>
960     <xs:extension base="dap:AppDataType">
961       <xs:attributeGroup ref="dst:ItemDataAttributeGroup" />
962     </xs:extension>
963   </xs:complexContent>
964 </xs:complexType>
965 <!--endsec(resqry)-->
966 <!--sec(subscr)-->
967 <xs:element name="Subscription" type="dap:SubscriptionType" />
968 <xs:complexType name="SubscriptionType">
969   <xs:complexContent>
970     <xs:extension base="subs:SubscriptionType">
971       <xs:sequence>
972         <xs:element ref="dap:ResultQuery" minOccurs="0" maxOccurs="unbounded" />
973         <xs:element name="Aggregation" minOccurs="0" maxOccurs="1" type="dap:AggregationType" />
974         <xs:element name="Trigger" minOccurs="0" maxOccurs="1" type="dap:TriggerType" />
975       </xs:sequence>
976     </xs:extension>
977   </xs:complexContent>
978 </xs:complexType>
979 <!--endsec(subscr)-->
980 <!--sec(notif)-->
981 <xs:complexType name="NotifyType">
982   <xs:complexContent>
983     <xs:extension base="dst:RequestType">
984       <xs:sequence>
985         <xs:element ref="dap:Notification" minOccurs="0" maxOccurs="unbounded" />
986       </xs:sequence>
987       <xs:attributeGroup ref="subs:NotifyAttributeGroup" />
988     </xs:extension>
989   </xs:complexContent>
990 </xs:complexType>
991 <xs:element name="Notification" type="dap:NotificationType" />
992 <xs:complexType name="NotificationType">
993   <xs:complexContent>
994     <xs:extension base="subs:NotificationType">
995       <xs:sequence>
996         <xs:element ref="dap:ItemData" minOccurs="0" maxOccurs="unbounded" />
997       </xs:sequence>
998     </xs:extension>
999   </xs:complexContent>
1000 </xs:complexType>
1001 <xs:complexType name="NotifyResponseType">

```



```
1002     <xs:complexContent>
1003         <xs:extension base="subs:NotifyResponseType" />
1004     </xs:complexContent>
1005 </xs:complexType>
1006 <!--endsec(notif)-->
1007 </xs:schema>
1008
1009
```

## 8. WSDL for ID-SIS-DAP

The Abstract Web Services Description Language (WSDL) declaration for the ID-SIS-DAP follows. The declaration states what is derived from [\[LibertyDST\]](#), namely that ID-SIS-DAP is characterized by `Query` and `Modify` operations cast to the namespace of ID-SIS-DAP.

```
1014 <?xml version="1.0" encoding="UTF-8"?>
1015 <definitions
1016     xmlns:typens="urn:liberty:id-sis-dap:2006-08:dst-2.1:wSDL:interface"
1017     xmlns="http://schemas.xmlsoap.org/wSDL/"
1018     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1019     xmlns:dap="urn:liberty:id-sis-dap:2006-08:dst-2.1"
1020     targetNamespace="urn:liberty:id-sis-dap:2006-08:dst-2.1:wSDL:interface"
1021     name="id-sis-dap_2006-08_interface">
1022   <types>
1023     <xsd:schema>
1024       <xsd:import
1025         namespace="urn:liberty:id-sis-dap:2006-08:dst-2.1"
1026         schemaLocation="liberty-id-sis-dap-v1.0.xsd"/>
1027     </xsd:schema>
1028   </types>
1029
1030   <message name="Query">
1031     <part name="body" element="dap:Query" />
1032   </message>
1033   <message name="QueryResponse">
1034     <part name="body" element="dap:QueryResponse" />
1035   </message>
1036
1037   <message name="Create">
1038     <part name="body" element="dap:Create" />
1039   </message>
1040   <message name="CreateResponse">
1041     <part name="body" element="dap:CreateResponse" />
1042   </message>
1043
1044   <message name="Delete">
1045     <part name="body" element="dap:Delete" />
1046   </message>
1047   <message name="DeleteResponse">
1048     <part name="body" element="dap:DeleteResponse" />
1049   </message>
1050
1051   <message name="Modify">
1052     <part name="body" element="dap:Modify" />
1053   </message>
1054   <message name="ModifyResponse">
1055     <part name="body" element="dap:ModifyResponse" />
1056   </message>
1057
1058   <message name="Notify">
1059     <part name="body" element="dap:Notify" />
1060   </message>
1061   <message name="NotifyResponse">
1062     <part name="body" element="dap:NotifyResponse" />
1063   </message>
1064
1065   <portType name="IDDAPPort">
1066
1067     <operation name="IDDAPQuery">
1068       <input message="typens:Query" />
1069       <output message="typens:QueryResponse" />
1070     </operation>
1071
1072     <operation name="IDDAPCreate">
1073       <input message="typens:Create" />
```

```
1074         <output message="typens:CreateResponse" />
1075     </operation>
1076
1077     <operation name="IDDAPDelete">
1078         <input message="typens:Delete" />
1079         <output message="typens:DeleteResponse" />
1080     </operation>
1081
1082     <operation name="IDDAPModify">
1083         <input message="typens:Modify" />
1084         <output message="typens:ModifyResponse" />
1085     </operation>
1086
1087     <operation name="IDDAPNotify">
1088         <input message="typens:Notify" />
1089         <output message="typens:NotifyResponse" />
1090     </operation>
1091
1092 </portType>
1093 </definitions>
1094
1095
```

---

# References

## Normative

- 1096
- 1098 [LibertyDisco] Cahill, Conor, Hodges, Jeff, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0-  
1099 errata-v1.0, Liberty Alliance Project (29 November, 2006). <http://www.projectliberty.org/specs>
- 1100 [LibertyDST11] Kainulainen, Jukka, Ranganathan, Aravindan, eds. "Liberty ID-WSF Data Services Template Speci-  
1101 fication," Version 1.1, Liberty Alliance Project (14 December 2004). <http://www.projectliberty.org/specs/>
- 1102 [LibertyDST] Kellomäki, Sampo, Kainulainen, Jukka, eds. "Liberty ID-WSF Data Services Template," Version 2.1,  
1103 Liberty Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 1104 [LibertyIDPP] Kellomäki, Sampo, Lockhart, Rob, eds. "Liberty ID-SIS Personal Profile Service Specification,"  
1105 Version 1.1, Liberty Alliance Project (29 September, 2005). <http://www.projectliberty.org/specs>
- 1106 [LibertyInteract11] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 1.1, Liberty  
1107 Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 1108 [LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version  
1109 1.2-errata-v3.0, Liberty Alliance Project (14 December 2004). <http://www.projectliberty.org/specs>
- 1110 [LibertyReg] Kemp, John, eds. "Liberty Enumeration Registry Governance," Version 1.1, Liberty Alliance Project (14  
1111 December, 2004). <http://www.projectliberty.org/specs>
- 1112 [LibertySOAPBinding] Hodges, Jeff, Kemp, John, Aarts, Robert, Whitehead, Greg, Madsen, Paul, eds. "Liberty  
1113 ID-WSF SOAP Binding Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007).  
1114 <http://www.projectliberty.org/specs>
- 1115 [LibertySUBS] Kellomäki, Sampo, eds. "Liberty ID-WSF Subscriptions and Notifications," Version 1.0, Liberty  
1116 Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 1117 [RFC2119] S. Bradner "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet  
1118 Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt>
- 1119 [RFC2251] "Lightweight Directory Access Protocol (v3)," M. Wahl T. Howes S. Kille (December 1997). RFC 2251,  
1120 Internet Engineering Task Force <http://www.ietf.org/rfc/rfc2251.txt>
- 1121 [RFC2252] "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions," M. Wahl A.  
1122 Coulbeck T. Howes S. Kille (December 1997). RFC 2252, Internet Engineering Task Force  
1123 <http://www.ietf.org/rfc/rfc2252.txt>
- 1124 [RFC2254] "The String Representation of LDAP Search Filters," T. Howes (December 1997). RFC 2254, Internet  
1125 Engineering Task Force <http://www.ietf.org/rfc/rfc2254.txt>
- 1126 [RFC2696] "LDAP Control Extension for Simple Paged Results Manipulation," C. Weider A. Herron A. Anantha T.  
1127 Howes (September 1997). RFC 2696, Internet Engineering Task Force <http://www.ietf.org/rfc/rfc2696.txt>
- 1128 [RFC2849] Good, G., eds. (June 2000). "The LDAP Data Interchange Format (LDIF) - Technical Specification,"  
1129 RFC 2828, Internet Engineering Task Force <http://www.ietf.org/rfc/rfc2849.txt>
- 1130 [RFC2891] Howes, T., Wahl, M., eds. (August 2000). "LDAP Control Extension for Server Side Sorting of Search  
1131 Results," RFC 2891, Internet Engineering Task Force <http://www.ietf.org/rfc/rfc2891.txt>

- 1132 [LDAPPSearch] Smith, M., Good, G., Howes, T., Weltman, R., eds. (06 August, 1998). "Persistent Search: A Simple  
1133 LDAP Change Notification Mechanism (expired, WIP)," Internet-Draft, The Internet Engineering Task Force  
1134 <http://www.ietf.org/proceedings/98aug/I-D/draft-ietf-ldapext-psearch-01.txt>
- 1135 [LDAPVLV] Boreham, D., Sermersheim, J., Kashi, A., eds. (November, 2002). "LDAP Extensions for Scrolling  
1136 View Browsing of Search Results (expired, WIP)," Internet-Draft, The Internet Engineering Task Force  
1137 <http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-ldapext-ldav3-vlv-09.txt>
- 1138 [LDAPproxyauth] Weltman, Rob, eds. (June, 2005). "LDAP Proxied Authorization Control (WIP)," Internet-Draft,  
1139 The Internet Engineering Task Force <http://www.ietf.org/internet-drafts/draft-weltman-ldav3-proxy-13.txt>
- 1140 [Schema1-2] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (28 October  
1141 2004). "XML Schema Part 1: Structures Second Edition," Recommendation, World Wide Web Consortium  
1142 <http://www.w3.org/TR/xmlschema-1/>
- 1143 [XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C. M., Maler, Eve, Yergeau, Francois, eds. (04 February 2004).  
1144 "Extensible Markup Language (XML) 1.0 (Third Edition)," Recommendation, World Wide Web Consortium  
1145 <http://www.w3.org/TR/2004/REC-xml-20040204>
- 1146 [XPath] Clark , J., DeRose , S., eds. (16 November 1999). "XML Path Language (XPath) Version 1.0 ,"  
1147 Recommendation, W3C <http://www.w3.org/TR/xpath> [August 2003].

## 1148 Informative

- 1149 [LibertyIDWSFGuide10] Weitzel, David, eds. (22 May 2005). "Liberty ID-WSF Implementation Guideline," Draft  
1150 v1.0-12, Liberty Alliance Project <http://www.projectliberty.org/specs/>
- 1151 [LibertyIDPPGuide] Kellomäki, Sampo, Lockhart, Rob, eds. "Liberty ID-SIS Personal Profile Service Implementation  
1152 Guidelines," Version 1.1, Liberty Alliance Project (29 September, 2005). <http://www.projectliberty.org/specs>
- 1153 [LibertyIDWSFOverview] Tourzan, Jonathan, Koga, Yuzo, eds. "Liberty ID-WSF Web Services Framework  
1154 Overview," Version 2.0, Liberty Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>