Liberty Alliance Project:

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

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

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

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

1. Introduction  .............................................................................................................. 4  
   1.1. Notational Conventions .................................................................................... 4  
   1.2. Derivation of ID-SIS-DAP from DST and WSF ................................................... 4  
   1.3. Compliance ....................................................................................................... 4  
   1.4. Namespaces ...................................................................................................... 5  
2. Definition of Parameters (checklists) ................................................................. 6  
   2.1. DST-Mandated Parameters ........................................................................... 6  
   2.2. Subscription Parameters ............................................................................... 9  
3. Mapping to LDAP .................................................................................................... 11  
4. Processing Rules and Other Considerations ...................................................... 12  
   4.1. Query Is Not Required to Report Same Data to Repeated Queries ................ 12  
   4.2. Simulation (dry-run) is Required ................................................................... 12  
   4.3. Use of Usage Directives ................................................................................ 12  
   4.4. Granularity of Metadata ............................................................................... 12  
   4.5. Subscription to Change in an Attribute ....................................................... 12  
5. DST 1.1 Support .................................................................................................... 14  
6. Examples ................................................................................................................... 15  
   6.1. ID-SIS-DAP Query Request ........................................................................... 15  
   6.2. ID-SIS-DAP QueryResponse ...................................................................... 16  
7. Schema for ID-SIS-DAP ....................................................................................... 18  
   7.1. Summary of ID-SIS-DAP ............................................................................ 18  
   7.2. Formal XML Schema for ID-SIS-DAP ...................................................... 20  
8. WSDL for ID-SIS-DAP .......................................................................................... 26  
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
documentation about the implementation clearly discloses which optional parts of the schema and which features are not supported. All other features and schema are assumed to be supported. A deployment of an implementation that is not capable of supporting the full schema SHOULD only register the discovery option keywords that accurately reflect its capabilities.

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

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

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

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

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

1.4. Namespaces

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

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dap:</td>
<td>urn:liberty:id-sis-dap:2006-08:dst-2.1</td>
<td>See Section 2.1, checklist 1</td>
</tr>
<tr>
<td>dst:</td>
<td>urn:liberty:dst:2006-08</td>
<td>[LibertyDST]</td>
</tr>
<tr>
<td>subs:</td>
<td>urn:liberty:ssos:2006-08</td>
<td>[LibertySUBS]</td>
</tr>
<tr>
<td>xml:</td>
<td><a href="http://www.w3.org/TR/REC-xml">http://www.w3.org/TR/REC-xml</a></td>
<td>XML Definition [XML] (for xml:lang)</td>
</tr>
</tbody>
</table>
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.
d. If <dn> is provided, the ID-SIS-DAP server MAY further determine the actual distinguished name from the identity information conveyed by the ID-WSF layer. In this case, the <dn> could act as the base suffix, while the identity information would provide the naming attribute.

e. If <Select> is used in <DeleteItem> or <ModifyItem> relating to objectType "entry," <Filter> element MUST NOT be specified and NONE of the XML attributes MUST be specified.

f. In case of omission of an XML attribute, the specified default value MUST be used. Usually, these mean that the limitation does not apply or, in case of attributes, that all attributes, except metadata attributes, should be returned.

g. An implementation that is not LDAP-based MAY interpret the fields in any other way that achieves the semantics inherent in [RFC2251].

h. A specification that profiles ID-SIS-DAP MAY specify additional objectTypes and MUST specify which query language applies to each of them. For example, a relational database profile could specify objectType "relational" with query language SQL.

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

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

   a. Order: "a" == ascending (reverse order false, the default), "d" == descending (reverse order true),

   b. Ordering rule (empty means default for the attribute),

   c. Attribute name.

These elements are interpreted in the sense of [RFC2891], Section 1.1 "Request Control."

Example:

   a\$cn\$d\$fn\$d\$sn

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

Sorting is not supported for subscriptions.

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

9. All DST-specified discovery option keywords are supported with the semantics defined in [LibertyDST].

   All discovery option keywords specified in [LibertySUBS] are supported.

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

   urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf1.1:dst1.1
   urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf1.1:dst2.1
   urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf2.0:dst1.1
   urn:liberty:ID-SIS-DAP:2006-08:framework:idwsf2.0:dst2.1
10. Element uniqueness mechanisms for objects of type "_Subscription" are specified in [LibertySUBS] (i.e., 
    subscriptionID).
    For objects of type "entry," *distinguished name* provides a unique identifier.

11. The default policy for protocol extension applies.
    For objects of type "entry," no special extension mechanism is needed. ID-SIS-DAP does not specify any schema
    so all entries might as well belong to LDAP objectClass "extensibleObject."
    While ID-SIS-DAP as protocol does not mandate any schema, or mechanism for schema validation, the
    underlying backend MAY, in fact, be schema-aware and perform schema validation in the way usual for directory
    servers. The agreement about the schema between WSC and WSP is out-of-band.

12. Query features (default feature support policy applies):
    a. Testing MUST be supported.
    b. `<ResultQuery>` MUST be supported.
    c. Sorting is OPTIONAL (typically, the backend would have to implement [RFC2891]). If sorting is not
       supported for objectType "entry," discovery option keyword urn:liberty:dst:noSorting MUST be
       registered.
       Sorting need not be supported for objectType "_Subscription" and the lack of such support is NOT
       reflected in discovery option keyword registrations.
    d. Pagination of results is OPTIONAL (typically, the backend would have to implement [RFC2696]
       or [LDAPVVL]). If pagination is not supported for objectType "entry," discovery option keyword
       urn:liberty:dst:noPagination MUST be registered.
       Pagination need not be supported for objectType "_Subscription" and the lack of such support is NOT
       reflected in discovery option keyword registrations.
    e. Support static sets in pagination is OPTIONAL (the backend probably would have to support [LDAPVVL]).
       If static sets are not supported, discovery option keyword urn:liberty:dst:noStatic MUST be
       registered.
    f. Multiple `<Query>` elements MUST be supported. No transactional semantic is attached to this construct.
    g. Multiple `<QueryItem>` elements MUST be supported. No transactional semantic is attached to this
       construct.
    h. Multiple `<TestItem>` elements MUST be supported. No transactional semantic is attached to this construct.
    i. `changedSince` MUST be supported in `<ResultQuery>` and `<QueryItem>`, but the granularity of the
       change detection MAY be one directory entry. `<ChangeFormat>` CurrentElements MUST be supported
       and is interpreted as returning the entire changed directory entry, possibly projected using the attribute
       list supplied in the query. `<ChangeFormat>` specifications ChangedElements and All MUST NOT be
       used. If `<ChangeFormat>` is omitted, CurrentElements is assumed. Since there is only one possible
       `<ChangeFormat>`, we RECOMMENDED that `<ChangeFormat>` is always omitted.
    j. `includeCommonAttributes` MUST be supported.

13. Create features:
    a. Multiple `<Create>` elements MUST be supported. No transactional semantic is attached to this construct.

14. Delete features:

Liberty Alliance Project
a. Multiple `<Delete>` elements MUST be supported. No transactional semantic is attached to this construct.

15. Modify features:

a. Multiple `<Modify>` elements MUST be supported. No transactional semantic is attached to this construct.
b. Multiple `<ModifyItem>` elements MUST be supported. No transactional semantic is attached to this construct.
c. Multiple `<ModifyItem>` elements may fail independently. No atomic modify or transactional semantic needs to be supported.
d. `notChangedSince` MUST be supported, but only at directory entry granularity.

2.2. Subscription Parameters

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

1. Standard names are used. See Section 7.

2. The `<Subscription>` elements represent objects of type "_Subscription" in AppDataType with schema as specified in [LibertySUBS].

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

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

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

6. Extension mechanisms for objects of type "_Subscription" are specified in [LibertySUBS].

7. How subscriptions can be established and manipulated:

   a. Supporting `objectType` "_Subscription" is OPTIONAL (typically, the backend would have to support [LDAPPSearch]). If subscription is not supported, the discovery option keyword `urn:liberty:dst:noSubscribe` MUST be registered.
   b. Subscribing in `<Query>` MUST be supported if `objectType` "_Subscription" is supported. However, subscriptions to objects of type "_Subscription" need not be supported.
   c. Multiple `<Subscription>` elements in `<Query>` MUST be supported if `objectType` "_Subscription" is supported.
   d. Subscribing in `<Create>` MUST be supported if `objectType` "_Subscription" is supported. However, subscriptions to objects of type "_Subscription" need not be supported.
   e. Multiple `<Subscription>` elements in `<Create>` MUST be supported if `objectType` "_Subscription" is supported.
   f. Subscribing in `<Modify>` MUST be supported if `objectType` "_Subscription" is supported. However, subscriptions to objects of type "_Subscription" need not be supported.
   g. Multiple `<Subscription>` elements in `<Modify>` MUST be supported if `objectType` "_Subscription" is supported.
8. The `starts` XML attribute MUST be supported for specification of the start of a subscription if `objectType` "_Subscription" is supported.

9. If `objectType` "_Subscription" is supported, it MUST be possible to expire subscriptions by at least two methods:
   a. by expiration of credentials,
   b. by specification of the `expires` XML attribute, and omission of `expires` implies expiration when credentials expire.

Expiration will happen at the earlier of the occasions specified by (a) and (b).

10. Use of same value for the `expires` and `starts` XML attributes MUST be supported to request one notification message at a specified time if `objectType` "_Subscription" is supported.

11. Querying existing subscriptions SHOULD be supported.

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 \texttt{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 \texttt{<ResultQuery>}, \texttt{<QueryItem>}, and \texttt{<TestItem>} map to LDAP \texttt{Search} operation in a straight forward way. \texttt{<TestItem>} can, in some cases, be mapped to the LDAP \texttt{Compare} operation. \texttt{<Sort>} can be mapped to the LDAP control described in [RFC2981]. Pagination functionality (\texttt{count} and \texttt{offset} XML attributes) can be mapped to either [RFC2696] or [LDAPVVL]. The static set functionality (\texttt{setID} and \texttt{setReq} XML attributes) can be mapped to [LDAPVVL].

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

ID-SIS-DAP \texttt{<DeleteItem>} maps straight to the LDAP \texttt{Delete} operation.

LDAP \texttt{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 \texttt{Bind} to authenticate, correctly, the Principal to the directory server. Since \texttt{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 \texttt{Unbind}, \texttt{Abandon}, or \texttt{Extended} operations. No requirement for such mapping is known.

There is no mapping for the LDAP \texttt{Modify RDN} (rename) operation.

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

1. The \texttt{<LDIF>} container in \texttt{<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 \texttt{<LDIF>} container in the \texttt{<NewData>} container MUST be ignored and MAY be entirely omitted.

3. Distinguished name for LDAP \texttt{Modify} operation MUST be taken from ID-WSF-conveyed identity combined with the \texttt{<Select>} element of the \texttt{<ModifyItem>}.

4. The LDIF format data is interpreted as descriptive of the changes to be applied as described in [RFC2849].

5. The \texttt{changeType} SHOULD be "Modify." However, other \texttt{changeTypes} are permitted for DST 1.1-based implementations.

6. Any number of \texttt{add}, \texttt{delete}, and \texttt{replace} specifications can be specified to effectuate the modification as described in [RFC2849].

7. \texttt{overrideAllowed} XML attribute MUST NOT be supplied in \texttt{<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.
4.5. Subscription to Change in an Attribute

If subscription is established by specifying a specific list of attributes, then the notification is triggered only if one of the specified attributes changes. If no specific attributes are specified, then change to any attribute in the entry triggers 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:Bearr security mechanism. This message would appear as the body of an https request.

```xml
<soap:Envelope
    xmlns:lib="urn:liberty:iff:2003-08"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    <soap:Header>
        <wsa:MessageID
            xmlns:wsa="http://www.w3.org/2005/03/addressing"
            id="#MID">uuid:6gSn02-nzY8vJYwKucW</wsa:MessageID>
        <wsa:To
            xmlns:wsa="http://www.w3.org/2005/03/addressing"
            https://g-sp.liberty-iop.org:8843/DAP-PSBEARER</wsa:To>
        <wsa:Action xmlns:wsa="http://www.w3.org/2005/03/addressing"/>
        <wsa:ReplyTo xmlns:wsa="http://www.w3.org/2005/03/addressing">
            http://www.w3.org/2005/03/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsse:Security
            xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
            <sa:Assertion
                xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
                ID="CRED1Wqsz7VozwEo6TBpkIMV"
                IssueInstant="2005-10-03T16:58:33Z"
                Version="2.0">
                <sa:Issuer
                    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity"
                <ds:Signature
                    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                    <ds:SignedInfo
                        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                        <ds:CanonicalizationMethod
                            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        <ds:SignatureMethod
                            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                        <ds:Reference URI="#CRED1Wqsz7VozwEo6TBpkIMV">
                            <ds:Transforms>
                                <ds:Transform
                                    Algorithm="http://www.w3.org/2001/09/xmldsig#enveloped-signature"/>
                                <ds:Transform
                                    Algorithm="http://www.w3.org/2001/09/xml-exc-c14n#"/>
                            </ds:Transforms>
                            <ds:DigestMethod Algorithm="http://www.w3.org/2003/09/xmldsig#sha1"/>
                            <ds:DigestValue>BA5GsDS1CmzvCTkNoBORaSCP9kI=</ds:DigestValue>
                        </ds:Reference>
                    </ds:SignedInfo>
                    <ds:SignatureValue>
                        Y+kMaOToVrbG8moohqiGyeMrKMS Dw4aizPsQZG6hDWDvph/q0eIow7pIFzveNo+7XUQbkEZ
                        aAOJ4KXEyyeRt0r5Xrh0zg6d2BCnKSH2XHwCE7d6qa5NJJGtQlszoJMtwwTcMByNKMKdcuksQdryGsk1NKED6ZPEFp
                        xK4aKs57Rs=
                    </ds:SignatureValue>
                </ds:Signature>
                <sa:Subject
                    xmlns:sa="urn:oasis:names:tc:SAML:2.0:assertion"
                    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
                    NameQualifier="https://g-ds.liberty-iop.org:8681/idp.xml">PEx7_Y-G_hMaysCNJmaQj</sa:Subject>
            </sa:Assertion>
        </wsse:Security>
    </soap:Header>
    <wsa:Body>
        <lib:QueryRequest
            xmlns:lib="urn:liberty:iff:2003-08"
            xmlns:libutility="urn:liberty:iff:utility:2003-08">
            <libutility:JCA
        </lib:QueryRequest>
    </wsa:Body>
</soap:Envelope>
```
6.2. ID-SIS-DAP QueryResponse

This is a response to the query in the previous example. This illustrates how the distinguished name is blanked out with value `liberty-identity-based-dn` to avoid leaking privacy-sensitive information that may have been present in the dn. The requested attribute has value `J2ME`. This message would appear as the body of an https response.
<wsa:ReplyTo>
  <sb2:Timestamp xmlns:sb2="urn:liberty:sb:2004-12">
    <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wss-utility-1.0.xsd">
      2005-10-03T16:58:34Z
    </wsu:Created>
  </sb2:Timestamp>
  <sb2:Sender>
  </sb2:Sender>
</soap:Header>
<soap:Body>
    id="dapid:aM7j4DhGRwZ69dsQPWvh"
    itemIDRef="itm:S9fdkoCcS8_PoRAhvRIt">
    <dap:Status code="OK"/>
    <dap:Data id="dapdataid:kc430k3KmG3Ju79KKFc6">
      dn: liberty-identity-based-dn JAVAPROFILE: J2ME
    </dap:Data>
  </dap:QueryResponse>
</soap:Body>
</soap:Envelope>
7. Schema for ID-SIS-DAP

7.1. Summary of ID-SIS-DAP

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

```xml
target(dap, urn:liberty:id-sis-dap:2006-08:dst-2.1)
import(dst, urn:liberty:dst:2006-08, liberty-idwsf-dst-v2.1.xsd)
import(subs, urn:liberty:ssos:2006-08, liberty-idwsf-subs-v1.0.xsd)
import(subsref, urn:liberty:ssos:2006-08:ref, liberty-idwsf-subs-ref-v1.0.xsd)
import(lu, urn:liberty:util:2006-08, liberty-idwsf-utility-v2.0.xsd)

#sec(methods)
Create -> %subsref:CreateType
CreateResponse -> %subsref:CreateResponseType
QueryResponse -> %subsref:QueryResponseType
Modify -> %subsref:ModifyType
ModifyResponse -> %subsref:ModifyResponseType
Delete -> %subsref:DeleteType
DeleteResponse -> %subsref:DeleteResponseType
Notify -> %subsref:NotifyType
NotifyResponse -> %subsref:NotifyResponseType

#endsec(methods)

# -----------------------------------------

#sec(redefs)
%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)
;
%TestOpType: base(dap:SelectType) ;
%SortType: base(xs:string) ;
%TriggerType: base(xs:string) ;
%AggregationType: base(xs:string) ;
%AppDataType:
dap:LDIF |
dap:Subscription
;
LDIF: base(xs:string)
|@dst:localizedLeafAttributes
;
#endsec(redefs)

# -----------------------------------------

#sec(create)
%CreateType: base(dst:RequestType)
dap:Subscription*
dap:CreateItem+
dap:ResultQuery*
;
CreateItem -> %dap:CreateItemType
%CreateItemType:
dap:NewData?
```
&dst:CreateItemAttributeGroup

NewData -> %dap:AppDataType

%CreateResponseType: base(dap:DataResponseType);
%DataResponseType: base(dst:DataResponseBaseType)
dap:ItemData*
;
#endsec(create)

# ==================================== ======

#sec(query)

%QueryType: base(dst:RequestType)
dap:TestItem*
dap:QueryItem*
dap:Subscription*
;
TestItem -> %dap:TestItemType
%TestItemType: base(dst:TestItemBaseType)
TestOp? -> %dap:TestOpType
;
QueryItem -> %dap:QueryItemType
%QueryItemType: base(dap:ResultQueryType)
&dst:PaginationAttributeGroup
;
#endsec(query)

#sec(queryresp)

%QueryResponseType: base(dst:DataResponseBaseType)
dst:TestResult*
dap:Data*
;
Data -> %dap:DataType
%DataType: base(dap:ItemDataType)
&dst:PaginationResponseAttributeGroup
;
#endsec(queryresp)

# ========================================== 

#sec(mod)

%ModifyType: base(dst:RequestType)
dap:Subscription*
dap:ModifyItem+
dap:ResultQuery*
;
ModifyItem -> %dap:ModifyItemType
%ModifyItemType:
dap:Select?
dap:NewData?
&dst:ModifyItemAttributeGroup
;
%ModifyResponseType: base(dap:DataResponseType);
#endsec(mod)

# ========================================== 

#sec(del)

%DeleteType: base(dst:RequestType)
dap:DeleteItem+

dap:DeleteItem+
7.2. Formal XML Schema for ID-SIS-DAP

The formal ID-SIS-DAP schema follows.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:id-sis-dap:2006-08:dst-2.1"
    xmlns:dap="urn:liberty:id-sis-dap:2006-08:dst-2.1"
    xmlns:dst="urn:liberty:dst:2006-08"
    xmlns:subs="urn:liberty:ssos:2006-08"
    xmlns:subsref="urn:liberty:ssos:2006-08:ref"
    xmlns:lu="urn:liberty:util:2006-08"/>
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="urn:liberty:dst:2006-08"
    schemaLocation="liberty-idwsf-dst-v2.1.xsd"/>
  <xs:import namespace="urn:liberty:ssos:2006-08"
    schemaLocation="liberty-idwsf-subs-v1.0.xsd"/>
  <xs:import namespace="urn:liberty:ssos:2006-08:ref"
    schemaLocation="liberty-idwsf-subs-ref-v1.0.xsd"/>
  <xs:import namespace="urn:liberty:util:2006-08"
    schemaLocation="liberty-idwsf-utility-v2.0.xsd"/>
  <!--sec(methods)-->  
  <xs:element name="Create" type="subsref:CreateType"/>
  <xs:element name="CreateResponse" type="subsref:CreateResponseType"/>
  <xs:element name="Query" type="subsref:QueryType"/>
  <xs:element name="QueryResponse" type="subsref:QueryResponseType"/>
  <xs:element name="Modify" type="subsref:ModifyType"/>
  <xs:element name="ModifyResponse" type="subsref:ModifyResponseType"/>
  <xs:element name="Delete" type="subsref:DeleteType"/>
  <xs:element name="DeleteResponse" type="subsref:DeleteResponseType"/>
  <xs:element name="Notify" type="subsref:NotifyType"/>
  <xs:element name="NotifyResponse" type="subsref:NotifyResponseType"/>
  <!--endsec(methods)-->  
  <!--sec(redefs)-->  
  <xs:complexType name="SelectType">
    <xs:sequence>
      <xs:element name="dn" minOccurs="0" maxOccurs="1" type="xs:string"/>
      <xs:element name="filter" minOccurs="0" maxOccurs="1" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="scope" use="optional" type="xs:integer" default="0"/>
    <xs:attribute name="sizelimit" use="optional" type="xs:integer" default="0"/>
    <xs:attribute name="timelimit" use="optional" type="xs:integer" default="0"/>
    <xs:attribute name="attributes" use="optional" type="xs:string"/>
    <xs:attribute name="typesonly" use="optional" type="xs:boolean" default="false"/>
    <xs:attribute name="derefaliases" use="optional" type="xs:integer" default="0"/>
  </xs:complexType>
  <xs:complexType name="TestOpType">
    <xs:extension base="dap:SelectType"/>
  </xs:complexType>
  <xs:complexType name="SortType">
    <xs:simpleContent>
      <xs:extension base="xs:string"/>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="TriggerType">
    <xs:simpleContent>
      <xs:extension base="xs:string"/>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="AggregationType">
    <xs:simpleContent>
      <xs:extension base="xs:string"/>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="AppDataType">
    <xs:choice>
      <xs:element ref="dap:LDIF"/>
      <xs:element ref="dap:Subscription"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="AppDataType">
    <xs:extension base="xs:string">
      <xs:attributeGroup ref="dst:localizedLeafAttributes"/>
    </xs:extension>
  </xs:complexType>
  <!--endsec(redefs)-->  
</xs:schema>
<xs:extension>
  <xs:simpleContent>
    <xs:complexType>
      <xs:element name="CreateType">
        <xs:complexContent>
          <xs:extension base="dst:RequestType">
            <xs:sequence>
              <xs:element ref="dap:Subscription" minOccurs="0" maxOccurs="unbounded"/>
              <xs:element ref="dap:CreateItem" minOccurs="1" maxOccurs="unbounded"/>
              <xs:element ref="dap:ResultQuery" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:extension>

<xs:complexType name="CreateItemType">
  <xs:sequence>
    <xs:element ref="dap:NewData" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attributeGroup ref="dst:CreateItemAttributeGroup"/>
</xs:complexType>

<xs:element name="NewData" type="dap:AppDataType"/>

<xs:complexType name="CreateResponseType">
  <xs:complexContent>
    <xs:extension base="dap:DataResponseType"/>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="DataResponseType">
  <xs:complexContent>
    <xs:extension base="dst:DataResponseBaseType">
      <xs:sequence>
        <xs:element ref="dap:ItemData" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="TestItem" type="dap:TestItemType"/>

<xs:complexType name="TestItemType">
  <xs:complexContent>
    <xs:extension base="dap:ResultQueryType">
      <xs:attributeGroup ref="dst:PaginationAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="QueryItem" type="dap:QueryItemType"/>

<xs:complexType name="QueryItemType">
  <xs:complexContent>
    <xs:extension base="dst:ResultQueryType">
      <xs:attributeGroup ref="dst:PaginationAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:extension name="ModifyResponseType">
  <xs:complexContent>
    <xs:extension base="dap:DataResponseType"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ModifyItem" type="dap:ModifyItemType"/>
<xs:complexType name="ModifyItemType">
  <xs:complexContent>
    <xs:extension base="dst:ModifyItemBaseType"/>
    <xs:sequence>
      <xs:element ref="dap:Select" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="dap:NewData" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attributeGroup ref="dst:ModifyItemAttributeGroup"/>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ModifyResponseType">
  <xs:complexContent>
    <xs:extension base="dst:DataResponseBaseType"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="Data" type="dap:DataType"/>
<xs:complexType name="DataType">
  <xs:complexContent>
    <xs:extension base="dap:ItemDataType"/>
    <xs:attributeGroup ref="dst:PaginationResponseAttributeGroup"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ModifyItem" type="dap:ModifyItemType"/>
<xs:complexType name="ModifyItemType">
  <xs:complexContent>
    <xs:extension base="dst:ModifyItemBaseType"/>
    <xs:sequence>
      <xs:element ref="dap:Select" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="dap:NewData" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attributeGroup ref="dst:ModifyItemAttributeGroup"/>
  </xs:complexContent>
</xs:complexType>

<xs:element name="DeleteItem" type="dap:DeleteItemType"/>
<xs:complexType name="DeleteItemType">
  <xs:complexContent>
    <xs:extension base="dst:DeleteItemBaseType"/>
    <xs:sequence>
      <xs:element ref="dap:Select" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>
<xs:complexContent>
  <xs:extension base="subs:NotifyResponseType"/>
</xs:complexContent>
<!--endsec(notif)-->
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.

```xml
<definitions
    xmlns="http://schemas.xmlsoap.org/wSDL/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:dap="urn:liberty:id-sis-dap:2006-08:dst-2.1"
    targetNamespace="urn:liberty:id-sis-dap:2006-08:dst-2.1:wSDL:interface"
    name="id-sis-dap_2006-08_interface">
  <types>
    <xsd:schema>
      <xsd:import
        namespace="urn:liberty:id-sis-dap:2006-08:dst-2.1"
        schemaLocation="liberty-id-sis-dap-v1.0.xsd"/>
    </xsd:schema>
  </types>

  <message name="Query">
    <part name="body" element="dap:Query"/>
  </message>

  <message name="QueryResponse">
    <part name="body" element="dap:QueryResponse"/>
  </message>

  <message name="Create">
    <part name="body" element="dap:Create"/>
  </message>

  <message name="CreateResponse">
    <part name="body" element="dap:CreateResponse"/>
  </message>

  <message name="Delete">
    <part name="body" element="dap:Delete"/>
  </message>

  <message name="DeleteResponse">
    <part name="body" element="dap:DeleteResponse"/>
  </message>

  <message name="Modify">
    <part name="body" element="dap:Modify"/>
  </message>

  <message name="ModifyResponse">
    <part name="body" element="dap:ModifyResponse"/>
  </message>

  <message name="Notify">
    <part name="body" element="dap:Notify"/>
  </message>

  <message name="NotifyResponse">
    <part name="body" element="dap:NotifyResponse"/>
  </message>

  <portType name="IDDAPPort">
    <operation name="IDDAPQuery">
      <input message="typens:Query"/>
      <output message="typens:QueryResponse"/>
    </operation>

    <operation name="IDDAPCreate">
      <input message="typens:Create"/>
```
<output message="typens:CreateResponse"/>
</operation>

<operation name="IDDAPDelete">
  <input message="typens:Delete"/>
  <output message="typens:DeleteResponse"/>
</operation>

<operation name="IDDAPModify">
  <input message="typens:Modify"/>
  <output message="typens:ModifyResponse"/>
</operation>

<operation name="IDDAPNotify">
  <input message="typens:Notify"/>
  <output message="typens:NotifyResponse"/>
</operation>
</portType>
</definitions>
References

Normative


http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-ldapext-ldapv3-vlv-09.txt


http://www.w3.org/TR/xmlschema-1/

http://www.w3.org/TR/2004/REC-xml-20040204


Informative

