# Liberty ID-SIS Contact Book Service Implementation Guidelines

Version: 1.0-06

**Editors:**
Guillaume Lambert, France Télécom

**Contributors:**
Shin Adachi, NTT
Rajeev Angal, Sun
Salima Fazal Karim, France Télécom
Sean Franklin, American Express
Ariel Gordon, France Télécom
Jukka Kainulainen, Nokia
Sampo Kellomäki, France Télécom
Kurt Kolok, IEEE-ISTO
Rob Lockhart, IEEE-ISTO

**Abstract:**

This document provides implementation guidelines supplemental to the Liberty ID-SIS Contact Book (ID-SIS-CB) service specification. It is also the general guideline for the Liberty ID-SIS Contact Book. The reader is expected to be familiar with the Liberty ID-WSF Web Services Framework Overview, XML, SAML, SOAP, vCard, and the Liberty ID-SIS Personal and Employee Profiles. The Liberty ID-SIS Contact Book is a web service hosted by an application provider and usually discovered via a discovery service. It offers the ability to manage a contact directory. The contact format is based upon the vCard Specifications and may contain several types of information such as telephone numbers and postal and email addresses. An extension mechanism allows other arbitrary data to be included. An ID-SIS-CB service also stores information regarding the Principal him or herself but is not intended to replace the Liberty ID-SIS Personal or Employee Profiles. An ID-SIS-CB service is an instance of a data-oriented identity web service. An ID-SIS-CB service, like all data services, is characterized by an ability to query and update attribute data. It incorporates mechanisms from other specifications for access control and for conveying data validation information and usage directives.

**Filename:** draft-liberty-id-sis-contactbook-guidelines-v1.0-06.pdf

**Notice**

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

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



Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08855-1331, USA
info@projectliberty.org

# Contents

1. Introduction . . . . . 4
2. Overview of the Conceptual Data Model (a.k.a. "Generic vCard") . . . . . 6
3. Discovery and Queries . . . . . 7
4. Managing the CB . . . . . 8
5. Mapping the CDM . . . . . 10
6. Cultural Portability . . . . . 11
References . . . . . 12

# 1. Introduction

The Liberty ID-SIS Contact Book Service Specification [LibertyCB] defines a Liberty identity service that supports information regarding the Principal and his or her contacts. The information regarding the Principal is stored in a unique card called "My Card" or "Self Card." This card is not intended to substitute for the Liberty ID-SIS Personal Profile Service [LibertyIDPP]. The Liberty ID-SIS Contact Book (ID-SIS-CB) service also allows the Principal to manage contacts for private and business acquaintances, friends, family members, and even for him or her self. The Principal can also create a Distribution List in order to simplify the management of his or her contacts.

This document provides a rationale and guidance for implementers of the ID-SIS-CB. A companion document, Liberty ID-SIS Contact Book Service Specification, [LibertyCB], normatively describes the ID-SIS-CB. If there is a disagreement between the present document and [LibertyCB], the [LibertyCB] specification is prescriptive.

## 1.1. Document Audience

This document is intended for application developers and implementers. The reader is presumed to be familiar with XML, SAML, SOAP, vCard, and the Liberty ID-SIS Personal Profile, [LibertyIDPP] and [LibertyIDPPGuide]. The reader should be familiar with the Liberty ID-FF Architecture Overview ([LibertyIDFFOverview]) and the Liberty ID-WSF Web Services Framework Overview ([LibertyIDWSFOverview]).

## 1.2. Architectural Context of the ID-SIS-CB

The ID-SIS-CB service uses the general framework and the methods described in the Liberty ID-WSF Data Services Template Specification [LibertyDST]. The services that consult the ID-SIS-CB service use the Liberty architectural framework to prove that they are acting on behalf of the Principal or that the Principal has somehow consented to sharing the data, e.g., by means of a standing order or subscription. The identity services are further described in [LibertyIDWSFOverview].

The ID-SIS-CB service is based on the vCard standards [RFC2426] and [vCard21]. These standards are not [XML]-based contact formats. An abstract conceptual data model in XML has been created to specify the selection structure in query and modification operations. Mappings with multiple vCard formats and [LibertyIDPP] have been provided in the ID-SIS-CB specification.

### 1.2.1. ID-SIS-CB as an Interface

The data accessible through the ID-SIS-CB often comes from back-end systems that may serve other purposes as well. For example, an enterprise hosting an ID-SIS-CB service for its employees may choose to use their customer address database. Such sharing of back-end systems is considered normal practice and may cause one service to update data in another "out-of-band." Out-of-band updates are expressly allowed, but are considered out of scope for the purposes of the ID-SIS-CB specification.

The ID-SIS-CB service specification, at the formal and conceptual levels, specifies an [XML] document. However, this does not mean that data is necessarily stored as an XML document. The data could just as well be computed on the fly or fetched from a directory ([LDAP]) or relational database (SQL) server and formatted into XML only for the purpose of speaking Liberty protocols. When this document specifies behavior against a conceptual XML document, the implementation has to behave as if the document existed, but does not necessarily have to implement it in concrete terms.

The ID-SIS-CB also allows the data to be returned in other formats than the conceptual data model. The ID-SIS-CB service specification explains how to handle the vCard 2.1 [vCard21], vCard 3.0 [RFC2426], vCard Jabber [JEP0054], and vCard RDF [vCardRDF] formats. Other formats may be supported for particular purposes but the implementer has to provide the discovery options keywords and the mappings for those formats with the XML conceptual data model. Since other such supported formats imply a proprietary implementation of the ID-SIS-CB service, there is no reason that other web service clients (WSC) must understand the discovery options keywords associated to those formats.

### 1.2.2. Participants and Compliance Testing

The ID-SIS-CB is provided by an attribute provider (AP) [LibertyIDWSFGuide], sometimes referred to as an ID-SIS-CB provider. The AP is a Liberty ID-WSF web service that hosts the ID-SIS-CB. The ID-SIS-CB is queried or updated by a client which is usually a service provider (SP) [LibertyIDWSFOverview] acting on behalf of the Principal [LibertyIDWSFGuide]. The client is sometimes referred to as a WSC. The [LibertyIDWSFGuide] describes the means by which the Principal may delegate to the SP a right to invoke his or her ID-SIS-CB service, i.e., a service assertion. Before the SP can access the ID-SIS-CB, it usually, but not necessarily, has to discover which AP hosts the ID-SIS-CB for the Principal. This is accomplished using a discovery service [LibertyDisco] that issues the service assertions.

ID-SIS-CB compliance testing addresses both implementations and instances. ID-SIS-CB specifies an interface to which an implementation and an instance (deployment) of an ID-SIS-CB service conform. The implementation may be a software product offered by a vendor. Typically such a product, if configured and operated correctly, will provide an ID-SIS-CB service instance. For an AP instance to be ID-SIS-CB-compliant, it must use correctly an ID-SIS-CB-compliant implementation.

## 1.3. XML Document Instantiations

An ID-SIS-CB service may respond to a query with an XML instantiation of the conceptual data model schema. The XML documents that are specified by the [LibertyCB] XML schema are the most general serial representations of the information. The expression "most general" means that a document could fully instantiate that schema if all data has been provisioned and no permissions filtering occurs. After filtering, the transmitted content may no longer conform to this schema. Thus, implementers may need to adjust this schema before using it to implement services.

When queries that point to interior elements of the conceptual XML document are applied, the returned data contains the queried element, its contents, and the higher level containers to avoid possible confusions if several elements are queried. If the query asks for one or more whole contacts to be returned, the format in which the contacts are returned inside the SOAP answer may be specified as long as the ID-SIS-CB supports the requested format. The supported formats are specified in the discovery option keywords.

A potential confusion is that, as requests to an ID-SIS-CB service are actually SOAP documents, there is one schema for the SOAP layer and another for the document that is returned inside the SOAP body. The [LibertyCB] Service Specification does not define the SOAP schemas.

## 1.4. Extension Mechanisms

The ID-SIS-CB schema has an extension container element which permits arbitrary schema extension. If an implementation supports schema extension, it must register the appropriate discovery option keyword.

# 2. Overview of the Conceptual Data Model (a.k.a. "Generic vCard")

## 2.1. Relationship between ID-SIS-CB and vCard

The Liberty ID-SIS-CB is based on vCard as specified in [vCard21] and [RFC2426]. All vCard types and their semantics, but not necessarily format, are hereby incorporated. The conceptual data model or "generic vCard" stands for an XML representation of vCard attributes and types. This means in particular that vCard categories are supported. (However, their use is not encouraged because of the several possible interpretations of this attribute. The ID-SIS-CB instead allows distribution lists to group together individual contact cards.)

The XML representation is used to specify [XPATH] query language and modification operations. Mappings to other vCard formats are provided in this specification. The purpose is to enable multiple vCard formats to coexist without endorsing any specific format. It is expected that new future vCard formats may be accommodated this way.

## 2.2. Structure of the Conceptual Data Model

The conceptual data model supports all attributes and types specified by [vCard21] and [RFC2426] except "`BEGIN`," "`END`," "`PROFILE`," "`NAME`," and "`SOURCE`" which are meaningless in an XML representation.

The jabber extension "`JABBERID`" and "`DESC`" are also supported. Additionally, the types "`FAVORITES`," "`LISTMEMBER`," "`SELF`," and "`CARDID`" have been added for ID-SIS-CB-specific purposes.

The type "`SELF`" is used to indicate the "Self Card" of the ID-SIS-CB owner, "`FAVORITE`" indicates the membership of a favorite list, and "`LISTMEMBER`" gives the membership of a distribution list.

"`CARDID`" is the identifier of the vCard.

# 3. Discovery and Queries

## 3.1. Rationale

The ID-SIS-CB is intended to satisfy the most possible needs for a contact directory in a Liberty environment. However, an implementer may choose to deploy only a few functionalities of the ID-SIS-CB. The ID-SIS-CB that a service provider sees is apt to be incomplete because:

- not all information about the contacts needs to be provisioned,
- not all of the vCard format needs to be implemented,
- an AP's policy forbids the SP from having some information, and/or
- the permissions that the Principal sets forbid the SP from accessing parts of the information.

The capacity of an ID-SIS-CB service implementation is expressed by the discovery option keywords. Option keywords are used to discover the existence or support for particular containers or groups of containers in a way that is meaningful to applications. See [LibertyDisco] for the generic definition of the Discovery Service and the processing rules for discovering by keyword.

## 3.2. Supported XPATH Expressions

Effectively, the minimal set of supported "`Modify`" [XPATH] expressions (i.e., "slashed paths") defines the minimal granularity of updates that needs to be supported. If a client needs to update an attribute at a finer granularity than defined here, then it should first query the element and then execute a "`Modify`" with queried values and the value it wants to change. It is recognized that this approach has inherent problems:

1. Other update(s) between the "`Query`" and "`Modify`": client should deal with this race condition in an implementation-dependent way, e.g., make a second query to verify that the update succeeded, ignore the possibility of a race condition.
2. "`Query`" may return incomplete data due to permissions. Presumably, under these circumstances, the corresponding "`Modify`" will fail for similar reasons. Updates to the containers listed above should be atomic whenever feasible. For example, if the underlying database technology is [LDAP], it is advisable to model each of the above-listed containers as an entry so that the directory server provides atomicity of update.

# 4. Managing the CB

## 4.1. ID-SIS-CB Overview

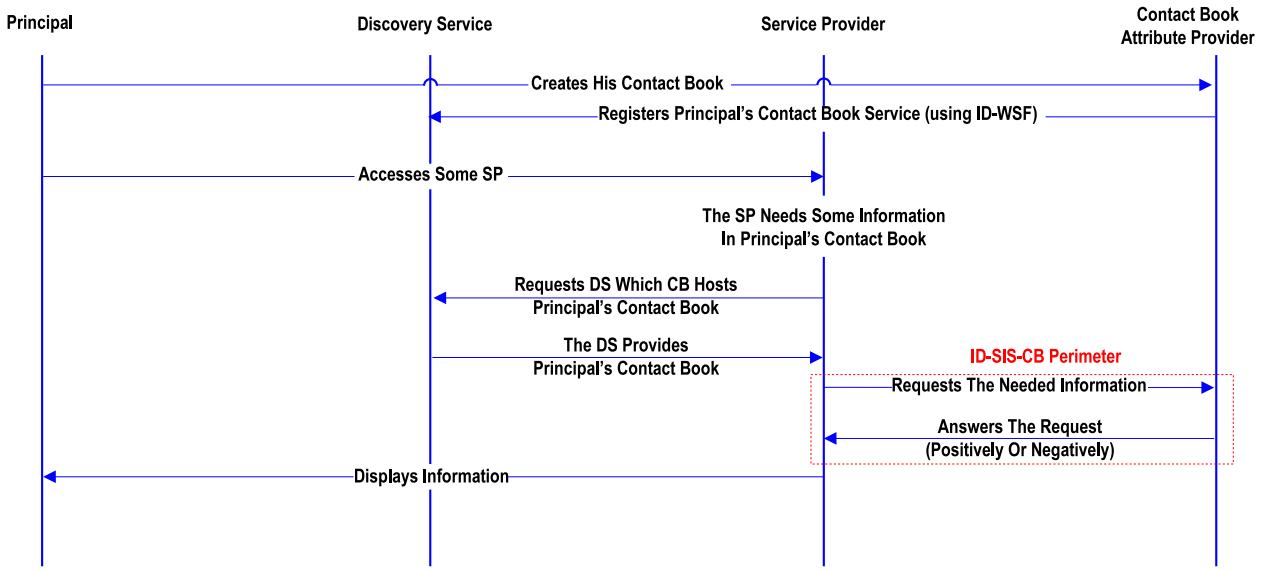


**Figure 1. ID-SIS-CB Overview**

## 4.2. ID-SIS-CB Interrogation

ID-SIS-CB is an AP as defined in [LibertyIDWSFGuide]. It can be invoked using the classical [LibertyDST] mechanisms.

The [XPATH] expression has to use the conceptual data model. As defined in the technical specification, when a whole card is requested, the returned format may be specified. When only a few vCard attributes are requested, the answer is mandatorily in the "generic vCard" format.

Examples of the SOAP bodies of an ID-SIS-CB "`query`" and "`response`":

In the following request, we want to obtain all vCards (`/cdm:vCard`) where the contact's nickname is "Jane" (`[/cdm:vCard/cdm:NICKNAME="Jane"]`) in the vCard 3.0 format (`<Select format="urn:liberty:cb:format:RFC2426">`).

```
<Query
  xmlns="urn:liberty:id-sis-cb:2005-02"
  id="my ID">
  <ResourceID>My Resource ID</ResourceID>
  <QueryItem changedSince="2004-03-03T16:49:09Z"
     id="my ID"
     includeCommonAttributes="false"
     itemID="my Item ID">
     <Select format="urn:liberty:cb:format:RFC2426">
        /cdm:vCard[/cdm:vCard/cdm:NICKNAME="Jane"]
     </Select>
  </QueryItem>
</Query>
```

The vCard is returned in raw vCard 3.0 format in the field <cb:CharData>

```
 <QueryResponse
   xmlns="urn:liberty:id-sis-cb:2005-02"
   id="my ID">
   <Status xmlns: ... >
   </Status>
   <Data itemIDRef="">
      <cb:Card cb:format="urn:liberty:cb:format:RFC2426">
         <cb:CharData>
BEGIN:vCard
FN:Jane Doe
N:Doe;Jane
NICKNAME:Jane
TEL;WORK:123456789
EMAIL;TYPE=internet:jane@example.com
END:vCard
         </cb:CharData>
      </cb:Card>
   </Data>
</QueryResponse>
```

# 5. Mapping the CDM

## 5.1. Mapping with vCard 2.1, vCard 3.0, vCard Jabber, and vCard RDF

The mappings provided by the [LibertyCB] specification are used to make the link between the conceptual data model and the different vCard formats used in the industry. Those mappings are almost intuitive but play an important role in the implementation. Even if the response format requested is not the "generic vCard," the ID-SIS-CB is queried by an [XPATH] expression upon the conceptual data model. It is important to have a standard and precise mapping to request of the ID-SIS-CB in order to ensure that different ID-SIS-CB implementations do not interpret the same [XPATH] query differently.

## 5.2. Mapping with Liberty ID-SIS Personal Profile

This mapping is provided to update the Principal's "Self Card" from his Liberty ID-SIS Personal Profile [LibertyIDPP]. Although the [LibertyIDPP] Personal Profile and the [LibertyCB] conceptual data model are very different, it can be very useful for a Principal to update his "Self Card" from his Personal Profile. The reverse update is, apparently, not true. This mapping does not aim to be a perfect mapping. It identifies what containers in the Personal Profile provide the same information that vCard attributes provide.

# 6. Cultural Portability

An Internet environment is the underlying assumption for these system designs. End users may venture to web sites outside their own culture and interact with other users and businesses in foreign countries. This calls for a common language. A large part of the world, but not the entire world, has standardized on the use of the Latin alphabet (character set) with some variations.

As noted, the Liberty ID-SIS-CB conceptual data model relies on the vCard semantics.

The default character set of vCard is ASCII. It can be overridden for an individual property value by using the "`CHARSET`" property parameter. This property parameter may be used on any property. However, the use of this parameter on some properties may not make sense. Any character set registered with the Internet Assigned Numbers Authority (IANA - `http://www.iana.org/`) may be specified by this property parameter.

The default language is "`en-US`" (US English). It may be overridden for an individual property value by using the "`LANGUAGE`" property parameter. The values for this property are strings consistent with [RFC1766]'s "Tags for the Identification of Languages." This property parameter may be used on any property. However, the use of this parameter on some properties, e.g., "`PHOTO`," "`LOGO`," "`SOUND`," "`TEL`," may not make sense.

The Liberty Framework provides a different character set property (`UTF-8`) at the transport wrapper level. During transport, the vCard data stream is encoded in `UTF-8`. Since `UTF-8` can represent all the characters of the other sets available in vCard, all languages supported by vCard may be supported.

# References

## Informative

[JEP0054] Saint-Andre, Peter, eds. (26 March, 2003). "Jabber Enhancement Proposal 0054: vcard-temp," v1.1, Jabber Software Foundation *http://www.jabber.org/jeps/jep-0054.html*

[LDAP] Wahl, M., Howes, T., Kille, S., eds. (December 1997). "Lightweight Directory Access Protocol (Version 3), ," RFC2251, Internet Engineering Task Force *http://www.rfc-editor.org/rfc/rfc2251.txt [August 2003].*

[LibertyCB] Kellomäki, Sampo, eds. "Liberty ID-SIS Contact Book Service Specification," Version 1.0-11, Liberty Alliance Project (16 May, 2005). *http://www.projectliberty.org/specs*

[LibertyDST] "Liberty ID-WSF Data Services Template Specification," Version 2.0-06, Liberty Alliance Project (22 November, 2004). *http://www.projectliberty.org/specs* Kainulainen, Jukka, Ranganathan, Aravindan, eds.

[LibertyDisco] Beatty, John, Hodges, Jeff, Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0-02, Liberty Alliance Project (24 Nov 2004). *http://www.projectliberty.org/specs*

[LibertyIDFFOverview] Wason, Thomas, eds. "Liberty ID-FF Architecture Overview," Version 1.2-errata-v1.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[LibertyIDPP] Kellomäki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.1, Liberty Alliance Project (25 May, 2005). *http://www.projectliberty.org/specs*

[LibertyIDPPGuide] Kellomäki, Sampo, Wason, Thomas, eds. "Liberty ID-SIS Personal Profile Service Implementation Guidelines," Version 1.1, Liberty Alliance Project (25 May, 2005). *http://www.projectliberty.org/specs*

[LibertyIDWSFGuide] Weitzel, David, eds. (13 January 2005). "Liberty ID-WSF Impelmentation Guideline," Draft version 2.0-02, Liberty Alliance Project *http://www.projectliberty.org/specs/*

[LibertyIDWSFOverview] Tourzan, Jonathan, Koga, Yuzo, eds. "Liberty ID-WSF Web Services Framework Overview," Version 1.0-errata-v1.0, Liberty Alliance Project (12 September 2004). *http://www.projectliberty.org/specs*

[RFC1766] Alvestrand, H., eds. (March 1995). "Tags for the Identification of Languages," RFC 1766, Internet Engineering Task Force *http://www.ietf.org/rfc/rfc1766.txt [March 1995].*

[RFC2426] Dawson, F., Howes, T., eds. (September 1998). "vCard MIME Directory Profile," RFC 2426, The Internet Engineering Task Force *http://www.ietf.org/rfc/rfc2426.txt [September 1998].*

[vCard21] Consortium, Versit, eds. (18 September, 1996). "vCard, The Electronic Business Card," Version 2.1, Internet Mail Consortium *http://www.imc.org/pdi/vcard-21.txt*

[vCardRDF] Iannella, Renato, eds. (22 February, 2001). "Representing vCard Objects in RDF/XML," W3C NOTE-vcard-rdf-20010222, World Wide Web Consortium *http://www.w3.org/TR/vcard-rdf*

[XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C.M., Maler, Eve, eds. (Oct 2000). "Extensible Markup Language (XML) 1.0 (Second Edition)," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/2000/REC-xml-20001006*

[XPATH] Clark , J., DeRose , S., eds. (16 November 1999). "XML Path Language (XPath) Version 1.0 ," Recommendation, W3C *http://www.w3.org/TR/xpath [August 2003].*