SIF-Approved Document #: SIF-013-1997
(SIF Contribution #: SIF-RL-9704-043-R5)

CONTRIBUTION TO SONET INTEROPERABILITY FORUM

SIF PROJECT:	SIF Remote Login Subgroup

TITLE: Directory Services and Network Address Resolution
for CIT/OS Systems

SOURCE: ATOS (former Sligos Marben)

CONTACTS:	Marc Degieux	ATOS (former Sligos Marben)
	Tel:	(33) 1 41 28 43 34
	E-mail:	mdegieux@clamart.marben.fr
	Jerome Moisand	Lucent Technologies
	Tel:	(508) 960 4637
	E-mail:	j.moisand@lucent.com

DATE: July 10, 1997

ABSTRACT

This contribution proposes the use of the LDAP API for fulfilling Network Address Resolution needs for applications located on a CIT or an OS system.

It describes a uniform architecture allowing to perform such resolution through the TARP protocol or through the X.500 DAP protocol. A detailed profile of the mandatory aspects of the involved software components is described.

NOTICE

This contribution has been prepared to assist SIF. This document is offered to SIF as a basis for discussion and is not a binding proposal on ATOS (former Sligos Marben) or any other company. The requirements presented in this document are subject to change in form and numerical value after more study. ATOS (former Sligos Marben) specifically reserves the right to add to, amend, or withdraw the statements contained herein.

This document has received the approval of the SONET Interoperability Forum (SIF).
August 28, 1997

1. Introduction

1.1 Background

This specification complements the SIF specification [SOFT-ARCH] for Management Systems like Craft Interface Terminal (CIT) systems or Operation Systems (OS).

This specification requires the use of the LDAP API (as defined in [RFC1823] and companion standards) on such systems as the standardized API when building applications making a direct use of Network Address Resolution Services, and details profiles for such a purpose.

Use of LDAP on another kind of SONET system (e.g. Network Element) is not precluded, although not part of this specification.

This specification has strong relationships with the T1M1 and SIF documents specifying the use of directory services for Telecommunication Management Networks (TMN), like [T1.245] and [TARP500], as well as [LDAP-SCHEMA].

1.2 References

The following documents are referenced in this specification :

[SOFT-ARCH]	SIF-DN-9706-045 : SIF OS/CIT Software Platform Architecture Specification
[GR-253]	Bellcore GR-253-CORE (1995) : Generic Requirements : SONET Transport Systems : Common Generic Criteria
[LDAP-SCHEMA]	SIF-RL-9706-046R2 : Lightweight Directory Access Protocol: ANSI T1.245 and SIF schema definition
[RFC1278]	IETF RFC 1278 : A string encoding of Presentation Address
[RFC1823]	IETF RFC 1823 : The LDAP Application Program Interface
[T1.245]	ANSI T1.245 1995 : Directory Services for TMN and SONET
[TARP500]	SIF-TA-9604-034-R1 : TARP/500 : the TARP and X.500 Directory Services Interworking Specifications

1.3 Acronyms and abbreviations

AET	Application Entity Title
API	Application Programming Interface
ASN.1	Abstract Syntax Notation 1

CIT	Craft Interface Terminal
AVA	Attribute Value Assertion
CMIP	Common Management Information Protocol
DAP	(X.500) Directory Access Protocol
DN	Distinguished Name
DSA	(X.500) Directory Server Agent
DUA	(X.500) Directory User Agent
IS/IS L1	Intermediate System to Intermediate System protocol (Level 1)
LDAP	Lightweight Directory Access Protocol
NARSE	Network Address Resolution Service Element
NE	Network Element
OS	Operations System
OSI	Open Systems Interconnection
RA	Registration Agent
RDBMS	Relational Data Base Management System
RDN	Relative Distinguished Name
RFC	Request For Comments
RRP	(ANSI T1.245) Registration Request Protocol
TL1	Transaction Language 1
TMN	Telecommunication Management Network

1.4 Revision History

ISSUE	**DATE**	**SUMMARY OF CHANGES**
First Draft	15-Apr-1997	Very early draft
Second Draft	22-May-1997	Reviewed following Baltimore SIF meeting (April 1997)
Third Draft	10-Jun-1997	Reviewed for Woodland Hills SIF meeting (June 1997)
Fourth Draft	16-Jun-1997	Reviewed following Woodland Hills SIF meeting (June 1997)
Fifth Draft	10-Jul-1997	Reviewed following RL subgroup phone conference (July 1997)
Voted document	28-Aug-1997	Voted during Ypsilanti SIF plenary meeting (August 1997)

2. Use of LDAP API ; software architecture

2.1 Overview

In order to provide directory services within a Management System (CIT, OS, etc.), the following scheme is defined :

Applications

LDAP API

underlying software

local storage
or TARP protocol
or X.500 DAP protocol

This design allows applications to take advantage of various directory services in a uniform way.

The source of information is expected to be one of the following :

- a fully distributed directory achieved having a TARP entity in each TMN system,
- a more centralized directory achieved having X.500 DSAs conformant to [T1.245] reached with the help of the DAP/RRP protocols.
- local storage (RDBMS, flat files, etc.) contained in the Management System.

Therefore, except in the case of local storage, directory queries using either DAP or TARP have to be performed, following the principles described as the “T5” function in [TARP500].

In the X.500 case, it is important to note that actual source of information (as stored in the DSA) may come from two origins (that may be combined) :

- Network Elements or Operation Systems entries populated by a DUA/RA pair (as described in [T1.245]) acting in the NE/OS itself.
- « TARP-oriented » entries populated by a TARP500 Gateway (T5GW function, as described in [TARP500]) on behalf on NE/OS implementing only the TARP protocol.

Some useful background may be found in [TARP500] that summarizes why these quite distinct schemes (T5 vs T5GW) might be needed, and how they are working.

2.2 Use of the LDAP API

The LDAP API is defined in [RFC 1823] and companion standards.

The LDAP API has the following characteristics :

- it is simple to understand and use (no need to understand ASN.1)
- its external definition is not dependent on operating systems.
- it is easy to find implementations running on UNIX as well as Windows NT
- its external definition allows to achieve "multi-thread safe" implementations
- the LDAP protocol and the related API are becoming standards for directory support within the Internet world
- the LDAP API allows to reach several LDAP directory servers, that may or may not be X.500-based. As an example, some RDBMS mappings exist.

For all these reasons, the LDAP API is required for SONET Management Systems (CIT, OS, etc.) for several purposes :

- Network Address Resolution Service Element (NARSE) : either X.500-based, or TARP-based or local storage-based: needs only a reduced subset of the LDAP API in order to search network address based on application-level criteria.
- general-purpose Database Query Service Element : either X.500-based, or local storage-based. May be easily extended to other kind of sources of information. Sophisticated queries may be performed on the source of information.

2.3 Software architecture

In order to use the LDAP API for the purposes described in the previous sections, and in order to be consistent with the protocols defined with the SONET world, the following architecture is required :

```
========
LDAP API
========
  !      ====================
  +------     LDAP to T5        => DAP/RRP (T1.245) or TARP
  !      ====================
  !
  !      =======================
  +------ LDAP to Local Storage  => query to local storage
         =======================
```

Underneath the LDAP API, one or several “mapping modules” will allow to map the requests performed through the API to an actual source of information. Such a source may be either reached through a networking protocol or be achieved through local storage.

Two mapping modules are identified up to now :
a) LDAP to T5 : directory queries are performed using either DAP or TARP protocols, following the principles described as the "T5" function in [TARP500]
b) LDAP to Local Storage : directory queries are performed using local storage as source of information (e.g. RDBMS, flat files, etc.)

It has to be emphasized that the communication between the LDAP API and the mapping modules is a LOCAL matter within the Management System (CIT, OS, etc). Therefore no assumption is made about using the LDAP protocol or any kind of Inter-Process Communication for such internal communication.

Only the T1.245 DAP/RRP protocols and/or the GR-253 TARP protocol are used on the physical links going out of the Management System ("T5" function as described in [TARP500]).

3. NARSE profiles : overview

3.1 Introduction

Applications on a Management System share a common requirement : to be able to perform Network Address Resolution based on some application-level naming scheme.

The intent of NARSE profiles is to specify the mechanisms that are required to fulfill such needs.

This doesn't preclude a software vendor to provide products having additional capabilities, but allows to define a common, minimal subset that can be relied on.

3.2 Naming scheme and Information Model

In order to unify the view of the Network Address Resolution mechanisms by applications located on a Management System, the X.500 schema described in [T1.245] is used in any case.

The Target Identifier (TID) resolution is also covered by the information model, following principles described in [TARP500].

The naming tree as described in [T1.245] typically looks like :

c=US	Country
o=PhoneInc	Organization
ou=SouthernDiv	Organization Unit (maybe several levels)
cn=Smallville	TMN system (e.g. tmnNE)
cn=ftam	application process
cn=responder	application entity

That naming tree is typically separated in two parts :
a) geographical/organizational information (ou=z1, ou=z2, o=yy, c=xx)
b) TMN systems information (tmnNE entries, and related applicationProcess/Entity entries)

3.3 Local Naming Context

The geographical/organizational information for the local area is described as a "Local Naming Context" (LNC) by [TARP500].

That LNC may be either provided automatically through the RRP protocol via the “NamePrefix” field (a Distinguished Name -DN- prefix) to an End System implementing the Registration Agent function described in [T1.245], or locally provisioned.

Another point is that the TARP protocol doesn’t need geographical/organizational information to perform a TID address resolution : it starts by looking within the current IS/IS L1 area, then optionally expands the search to neighbor L1 areas.

In order to unify NARSE mechanisms, but also to allow any kind of queries (in the local area, or outside the local area), the application needs to be able to :
a) perform requests based on a full DN (giving **explicitly** the organization/geographical information as the first part of the sequence of RDNs) : explicit form
b) perform requests based on a partial DN (assuming **implicit** organization/geographical information, therefore giving only the naming components (RDNs) starting from the tmnNE entries in the naming tree : implicit form.

Therefore, the protocol mapping is the following :
a) when using X.500/DAP, an implicit geographical/organizational information means the local area as described by the NamePrefix given through the RRP protocol. Only TMN systems information needs to be given as a sequence of RDNs in a directory request.
b) when using X.500/DAP, an explicit geographical/organizational information means that a DN will have to include the full sequence of RDNs to be used in a directory request.
c) when using TARP, the geographical/organizational information is always implicit, meaning that the address resolution will start in the current IS/IS L1 area, then optionally expand to neighbor areas. Only TMN systems information needs to be given as a sequence of RDNs in a directory request.

3.4 Resolutions to be achieved

3.4.1 Introduction

Network Address Resolution requirements for most applications is limited to translate an application-level name (either a TID or an AET) to an OSI presentation address.

In addition, if only TARP is usable as a directory protocol, the kind of directory requests that are feasible through an LDAP/T5 mapping module are very restricted.

On the other hand, more sophisticated queries might be needed by some applications, using more complex search criteria. But this is only feasible when using T1.245 DAP to access to a true X.500 DSA.

Therefore, two NARSE profiles are defined, allowing two levels of conformance :
a) NARSE “core” profile
b) NARSE “extended” profile

3.4.2 NARSE “Core” profile queries

The main query that may be performed is a translation of an application-level name (either a TID or an AET) to an OSI presentation address.

That query is defined by:
a) search with base object defined by the RDN components identifying the relevant application entity entry using **implicit** geographical/organizational information.
b) scope of the search : the base object itself
c) filter set to “(objectClass=applicationEntity)”

The RDN sequence needed when using an AET for identifying such a base object are described in the X.500 schema defined in [T1.245] and its LDAP counterpart [LDAP-SCHEMA].

Additional information for using a TID as an application-level name may be found in [TARP500].

The resulting OSI address will be found in the presentationAddress attribute of the entry returned. No more than one entry should be returned by such a query. The presentation address will be encoded following the principles described in [RFC 1278].

The reverse query may also be performed : a translation of an OSI presentation address to an application-level name (either a TID or an AET).

That query is defined by:
a) search with base object using **implicit** geographical/organizational information without any additional naming component.
b) scope of the search : whole subtree
c) filter set to “(& (objectClass=applicationEntity) (presentationAddress=XXX))”

The resulting AET will be given as the distinguished name of the entry returned. When speaking of a TID, following the principles described in [TARP500], the TID will be the last naming component of that distinguished name.

Only support for these two queries is required for the "core" profile.

When performing such queries (using **implicit** geographical/organizational information), the base object DN will possibly make use of the following name forms (defined in [T1.245] X.500 schema and its LDAP counterpart [LDAP-SCHEMA]) :

- tmnNENameForm
- applProcessNameForm
- applEntityNameForm
- apaeAliasNameForm (dereferencing being typically hidden to the user of the LDAP API)

Therefore only support for these name forms (and corresponding structure rules) is required for that "core" profile.

When performing such queries, only support for applicationEntity object class entries (cf. [T1.245] X.500 schema, and its LDAP counterpart [LDAP-SCHEMA]) retrieval is required. Therefore only support for the corresponding object class, attribute types and syntaxes is required.

3.4.3 NARSE "Extended" profile queries

That profile is a superset of the "core" profile, where the requests to be achieved are far less restricted.

A pretty good definition of possible needs may be found in annex I of the [T1.245] text. The following requests, given as examples, are defined based on that annex I :

- RQ1 : request the identity of an NE based on its Network Address
- RQ2 : request the identity of an NE based on technology, implementation, network, or operation-dependent naming.
- RQ3 : request the identity of NEs based on vendor or locality information, when known
- RQ4 : request the identity of an NE based on its MIB naming attribute
- RQ5 : request the value of managedElementId of an NE for use in management protocol exchanges
- RQ6 : request the identity of an NE based on its function and role
- RQ7 : request AE-titles of entities with which management associations may be established
- RQ8 : request the Presentation Address of application entities based on their AE-titles
- RQ9 : request the supported application contexts of an application entity

Such queries imply to be able to perform searches using :

a) various base objects (e.g. applicationEntity entry, tmnNE entry, the tmnNE direct superior entry, etc.)
b) scope being baseObject, singleLevel or wholeSubtree
c) filters with a combination of criteria. It has to be noticed that all the requests previously

described only need a “and” combination of Attribute Value Assertions (at most one level of filter recursity is needed).

Note that some of these queries are likely to return more than one entry.

In addition, queries may be performed using implicit or explicit geographical/organizational information.

When performing such queries for NARSE purposes, the base object DN will possibly make use of the following name forms (defined in [T1.245] X.500 schema and its LDAP counterpart [LDAP-SCHEMA]) :

- countryNameForm
- sOPNameForm
- locNameForm
- orgNameForm
- orgUnitNameForm
- orgUnitLocNameForm
- orgUnitLocOrgNameForm
- tmnNENameForm
- applProcessNameForm
- applEntityNameForm
- apaeAliasNameForm (alias dereferencing being typically hidden to the user of the LDAP API)

Therefore only support for these name forms (and corresponding structure rules) is required for that “extended” profile.

When performing such queries for NARSE purposes, only support for tmnNE or applicationEntity object class entries (cf. [T1.245] X.500 schema, and its LDAP counterpart [LDAP-SCHEMA]) retrieval is required. Therefore only support for the corresponding object class, attribute types and syntaxes is required for that “extended” profile.
Note: support for the sdhNEEntry auxiliary object class is not required.

4. NARSE profiles : use of the LDAP API

The following section describes a subset of the LDAP API (as specified in [RFC 1823]), describing the mandatory features to be provided by the LDAP API and the underlying software to provide the services needed for Network Address Resolution.

This description doesn’t preclude software vendors from providing more extended features, but allows to specify a subset on which applications may rely in any case.

Specifying such a subset might also help to reduce time to market of such software.

4.1 X.500 schema and LDAP counterpart

[T1.245] defines the X.500 schema to be used in a SONET environment.

[LDAP-SCHEMA] defines the LDAP counterpart to T1.245, therefore gives the necessary information for making queries using the LDAP API.

For NARSE purposes, support for only a subset of the schema is required. See sections 3.4.2 and 3.4.3 for the required name forms and object classes (and related attributes types and syntaxes).

4.2 LDAP API primitives

4.2.1 Synchronous and asynchronous flavors

The following section describes only the asynchronous flavor of LDAP API primitives.
But the equivalent synchronous primitives (same name, but with a « _s » suffix) are considered to be part of the NARSE profiles requirements.

4.2.2 ldap_open

This primitive is required to **locally** connect to the relevant LDAP mapping module.

```
LDAP  *ldap_open (char *hostname, int portno)
```

The hostname shall be the local host : “localhost”.

The port number is used for distinguishing between various LDAP mapping modules (the ones currently envisioned being the LDAP to T5 mapping module and the LDAP to Local Storage mapping module).

The port number for an LDAP mapping module may be different from the standard LDAP one (389), in order to avoid potential conflicts with other LDAP-based products.

4.2.3 ldap_bind and friends

Only the "simple" flavor is required :

```
ldap_simple_bind (LDAP *ld, char *dn, char *passwd)
```

Use of “dn” and “password” parameters are left for future use (e.g. authentication). NULL values allow to bind in an anonymous way.

4.2.4 ldap_unbind

```
ldap_unbind (LDAP *ld)
```

This primitive is required in order to unbind and close a connection.

4.2.5 ldap_search

This primitive is required in order to initiate a network address resolution.

ldap_search (LDAP *ld, char *base, int scope, char *filter, char *attrs[], int attrsonly)

Refer to section 3.4 for possible restrictions to the use of that primitive depending on the NARSE profile.

Also refer to section 4.2.9 for more details about the base object DN.

4.2.6 Parsing search entries

All the primitives helping to parse search entries are required :

ldap_first_entry(...)
ldap_next_entry(...)
ldap_count_entries(...)
ldap_first_attribute(...)
ldap_next_attribute(...)
ldap_get_values(...)
ldap_get_values_len(...)
ldap_count_values(...)
ldap_count_values_len(...)
ldap_value_free(...)
ldap_value_free_len(...)
ldap_get_dn(...)
ldap_explode_dn(...)
ldap_msgfree(...)

4.2.7 ldap_result

That primitive is required, in order to be able to obtain the result of a previous asynchronously initiated operation.

4.2.8 Other primitives

Other primitives (e.g. ldap_modify, ldap_modrdn, ldap_add, ldap_delete, ldap_abandon, etc.) are not required for NARSE profiles.

4.2.9 Implicit/explicit geographical/organizational information

The base object whose DN is given in the ldap_search primitive might include either implicit or explicit geographical/organizational information.

Explicit form is a full DN, containing the full sequence of naming components (RDNs), therefore including naming components like organizationalUnit, organization, locality, country.

Implicit form is a DN containing only the RDNs related to the TMN system entries and leaf entries (e.g. applicationProcess, applicationEntity).

In order to distinguish the two forms, a “fake” naming component is used as the end of the DN string when using the implicit form :

“npx=.” (meaning Naming Prefix = local area)

No other value than “.” is currently allowed for that “fake” naming component.

Example of explicit form identifying an FTAM application on a given NE:

“cn=ftam, cn=Smallville, ou=SouthernDiv, o=PhoneInc, c=US”

Example of implicit form:

“cn=ftam, cn=Smallville, npx=.”

Refer to section 4.3 for more details about the processing of implicit form by the LDAP/T5 mapping module.

4.2.10 Miscellaneous

Direct access to fields in the LDAP structure defined by the LDAP API (e.g. through an “ld” pointer) is discouraged (such direct access will be deprecated in LDAP v3).

As a consequence, only support for default behaviors as described by such fields is required (e.g. always dereference aliases).

4.3 LDAP mapping modules

Such mapping modules may be viewed as an implementation matter, but some of their characteristics might imply some additional semantics or some limitation when using the LDAP API to connect with them.

This section details such “externally visible” consequences.

4.3.1 NARSE profiles : LDAP to T5 mapping module

That mapping module allows to process directory requests using either DAP (following T1.245 principles) or TARP protocol.

These protocols shall be used following the principles described as the “T5 function” in the [TARP500] document, in order to be able to perform directory requests using one protocol or another.

In some cases, one of these two protocols (DAP and TARP) might not be usable. Consequences of the use of one protocol or another are described in the following sections.

4.3.2 NARSE profiles : LDAP to T5 mapping module using DAP/RRP

Such a mapping module may be used for any kind of request (“core” profile as well as “extended” profile) described in the previous sections.

It has to be emphasized that using the DAP protocol, an X.500 DSA server will be queried that may contain information describing NEs that are T1.245 conformant (i.e. including an X.500 DUA function), but also possibly NEs that "TARP-oriented" (thanks to the TARP500 gateway mechanism).

Implicit geographical/organizational information refers to the NamePrefix distributed through the RRP protocol described in T1.245, or locally provisioned. But use of explicit geographical/organizational information in base object DNs is also allowed.

Since the geographical/organization information is known thanks to the RRP protocol, the Distinguished Names in resulting entries will always use the explicit form when the resolution has been performed using the DAP protocol.

Therefore looking at a resulting DN to an query using a base object DN with implicit form, an application may easily extract the current Name Prefix (the Local Naming Context in other words).

4.3.3 NARSE profiles : LDAP to T5 mapping module using TARP

Thanks to the [TARP500] specification, the TID/NET resolution has to follow T1.245 schema principles, handling tmnNE entries and applicationEntity entries in a consistent way.

Only the ability to perform search requests conformant to the “core” profile is required. Other kinds of search request might be rejected if the DAP protocol cannot be used (no DSA available).

Only requests using **implicit** geographical/organizational information are processed, following TARP principles: try to solve the request in the current IS/IS L1 area, then optionally expand the flooding to adjacent areas.

Since the geographical/organization information is implicit (and fully unknown if the RRP protocol is not usable, e.g. in a TARP-only network), the Distinguished Names in resulting entries will always use the implicit form when the resolution has been performed using the TARP protocol.

4.3.4 NARSE profiles : LDAP to Local Storage mapping module

Such a mapping module is left for further study.

It is intended that both “core” profile and “extended” profile will be supported by such a mapping module.

5. Future extensions

This section is not part of the specification. It only gives a few hints for future directions.

5.1 LDAP v3 and related API

Moving to the version 3 of LDAP and the related upgrade of the LDAP API will have to be studied when the corresponding RFCs will be stable enough.

Thanks to the upward compatibility that should be enforced between the two versions of the LDAP APIs, such a move should be feasible in a smooth way.

5.2 Query Service Element

This function should be achieved using the LDAP API and LDAP mapping modules in the same way as for the NARSE function, but using the API far more extensively, removing most of the limitations. One could see the NARSE function as a subset of the Query Service Element.

5.3 Security mechanisms

A more robust approach than anonymous bind or use of a simple password for covering security issues is left for further study, but will most probably need to be defined.

Strong authentication might notably be studied in order to improve security when establishing associations. That leads to two possible extensions :

a) use of an X.500 directory to store public key certificates as well as certificate revocation lists (CRLs). Such pieces of information might then need to be accessed through the LDAP API and then be typically used for establishing an association using some management protocol (e.g. TL1 or CMIP).

b) use of strong authentication during the association establishment with the directory server itself, through the DAP protocol.

6. Requirements for Conforming Implementations

The following table describes the specific items that implementations claiming conformance to these requirements must support.

Optional requirements are tagged with an “O” mark ; mandatory requirements are tagged with an “M” mark.

Such requirements may be strengthened for a given platform type (e.g. in a document like [SOFT-ARCH]), only the minimal requirements shared by every platform are defined here.

No	Description	M/O	Reference
R1	NARSE “core” profile	M	2.3 3.1 3.2 3.3 3.4.1 3.4.2 4.x
R2	NARSE “extended” profile	O	2.3 3.1 3.2 3.3 3.4.1 3.4.3 4.x
R3	LDAP/T5 mapping module : T1.245 DAP/RRP support	O	[T1.245]
R4	LDAP/T5 mapping module : GR-253 TARP support	O	[GR-253]
R5	LDAP/T5 mapping module : full “T5” support	O	[T1.245] [GR-253] [TARP500]

Notes :

- R3, R4, R5 are mutually exclusive. Support for one of them is mandatory.
- R2 implies either R3 or R5