IDS Working Group
INTERNET-DRAFT

Al Grimstad
Rick Huber
AT&T
Sri Sataluri
Lucent Technologies
Steve Kille
Isode Ltd.
Mark Wahl
Critical Angle Inc.

November 26, 1997

Naming Plan for Internet Directory-Enabled Applications
Filename: draft-ietf-ids-dirnaming-03.txt

Status of this Memo

Abstract

Application of the conventional X.500 approach to naming has heretofore, in the experience of the authors, proven to be an obstacle to the wide deployment of directory-enabled applications on the Internet. We propose a new directory naming plan that leverages the strengths of the most popular and successful Internet naming schemes for naming objects in a hierarchical directory. This plan can, we believe, facilitate the creation of an Internet White Pages

Service (IWPS) and other directory-enabled applications by overcoming the problems encountered by those using the conventional X.500 approach to naming.

1.0 Executive Summary

Application of the conventional X.500 approach to naming has heretofore, in the experience of the authors, proven to be an obstacle to the wide deployment of directory-enabled applications on the Internet. The required registration infrastructure is either non-existent or largely ignored. The infrastructure that does exist is cumbersome to use and tends to produce counterproductive results. The attributes used for naming have been confusing for users and inflexible to managers and operators of directory servers.

This paper describes an alternative directory naming plan for the construction of an Internet directory infrastructure to support directory-enabled applications.

The plan has the following two main features. First, it bases the root and upper portions of the name hierarchy on the existing infrastructure of names from the Domain Name System (DNS). This component of the plan makes use of the ideas described in the companion paper to this plan, "Using Domains in LDAP Distinguished Names" [1]. And second, it provides a number of options for the assignment of names to directory leaf objects such as person objects, including an option that allows the reuse of existing Internet identifiers for people.

Here, in summary, is our proposal.

The upper portions of the hierarchical directory tree should be constructed using the components of registered DNS names using the domain component attribute "dc". The directory name for the organization having the domain name "acme.com" will then be, e.g.,

    dc=acme, dc=com

Organizations can add additional directory structure, for example to support implementation of access control lists or partitioning of their directory information, by using registered subdomains of DNS names, e.g., the subdomain "corporate.acme.com" can be used as the basis for the directory name

    dc=corporate, dc=acme, dc=com

For naming directory leaf objects such as persons, groups, server applications and certification authorities in a hierarchical

directory, we propose the use of either the "uid" (user identifier) or the "cn" (common name) attribute for the relative distinguished name. This plan does not constrain how these two attributes are used.

One approach to their use, for example, is to employ the uid attribute as the RDN when reusing an existing store of identifiers and the cn attribute as the RDN when creating new identifiers specifically for the directory. A convenient existing identification scheme for person objects is the RFC822 mailbox identifier. So an RDN for person employing this store of identifiers would be, e.g.,

uid=John.Smith@acme.com

For leaf objects not conveniently identified with such a scheme, the "cn" attribute is used, e.g.,

cn=Reading Room

Directory distinguished names will thus have the following structure, e.g.,

uid=John.Smith@acme.com, dc=acme, dc=com
uid=Mary.Jones@acme.com, dc=corporate, dc=acme, dc=com
uid=J.Smith@worldnet.att.net, dc=legal, dc=acme, dc=com
cn=Reading Room, dc=physics, dc=national-lab, dc=edu

2.0 The Problem

The X.500 Directory model [2] can be used to create a world-wide distributed directory. The Internet X.500 Directory Pilot has been operational for several years and has grown to a size of about 1.5 million entries of varying quality. The rate of growth of the pilot is far lower than the rate of growth of the Internet during the pilot period.

There are a substantial number of contributing factors that have inhibited the growth of this pilot. The common X.500 approach to naming, while not the preponderant problem, has contributed in several ways to limit the growth of an Internet White Pages Service based on X.500.

The conventional way to construct names in the X.500 community is documented as an informative (i.e., not officially standardized) Annex B to X.521. The relative distinguished name (RDN) of a user consists of a common name (cn) attribute. This is meant to be what -- in the user's particular society -- is customarily understood to be the name of that user. The distinguished name of a user is the combination of the name of some general object, such as an

organization or a geographical unit, with the common name. There are two main problems with this style of name construction.

First, the common name attribute, while seeming to be user-friendly, cannot be used generally as an RDN in practice. In any significant set of users to be named under the same Directory Information Tree (DIT) node there will be collisions on common name. There is no way to overcome this other than either by forcing uniqueness on common names, something they do not possess, or by using an additional attribute to prevent collisions. This additional attribute normally needs to be unique in a much larger context to have any practical value. The end result is a RDN that is very long and unpopular with users.

Second, and more serious, X.500 has not been able to use any significant number of pre-existing names. Since X.500 naming models typically use organization names as part of the hierarchy [2, 3], organization names must be registered. As organization names are frequently tied to trademarks and are used in sales and promotions, registration can be a difficult and acrimonious process.

The North American Directory Forum (NADF, now the North Atlantic Directory Forum but still the NADF) proposed to avoid the problem of registration by using names that were already registered in the "civil naming infrastructure" [4][5]. Directory distinguished names would be based on an organization's legal name as recognized by some governmental agency (county clerk, state secretary of state, etc.) or other registering entity such as ANSI.

This scheme has the significant advantage of keeping directory service providers out of disputes about the right to use a particular name, but it leads to rather obscure names. Among these obscurities, the legal name almost invariably takes a form that is less familiar and longer than what users typically associate with the organization. For example, in the US a large proportion of legal organization names end with the text ", Inc." as in "Acme, Inc." Moreover, in the case of the US, the civil naming infrastructure does not operate nationally, so the organization names it provides must be located under state and regional DIT nodes, making them difficult to find while browsing the directory. NADF proposes a way to algorithmically derive multi-attribute RDNs which would allow placement of entries or aliases in more convenient places in the DIT, but these derived names are cumbersome and unpopular. For example, suppose Nadir is an organization that is registered in New Jersey civil naming infrastructure under the name "Nadir Networks, Inc." Its civil distinguished name (DN) would then be

o="Nadir Networks, Inc.", st=New Jersey, c=US

while its derived name which is unambiguous under c=US directly is

o="Nadir Networks, Inc." + st=New Jersey, c=US

More generally, the requirement for registration of organizations in X.500 naming has led to the establishment of national registration authorities whose function is mainly limited to assignment of X.500 organization names. Because of the very limited attraction of X.500, interest in registering an organization with one of these national authorities has been minimal. Finally, multi-national organizations are frustrated by a lack of an international registration authority.

3.0 Requirements

A directory naming plan must provide for names of directory objects that are unambiguous (identify only one directory object) within some context (namespace), at a minimum within one isolated directory server.

A directory object is simply a set of attribute values. The association between a real-world object and a directory object is made by directory-enabled applications and is, in the general case, one to many.

The following additional naming characteristics are requirements that this naming plan seeks to satisfy:

a) hierarchical

The Internet, consisting of a very large number of objects and management domains, requires hierarchical names. Such names permit delegation in the name assignment process and partitioning of directory information among directory servers.

b) friendly to loose coupling of directory servers

One purpose of this naming plan is to define a naming pattern that will facilitate one form or another of loose coupling of potentially autonomous directory servers into a larger system.

A name in such a loosely-coupled system should unambiguously identify one real-world object. The real-world object may, however, be represented differently (i.e. by different directory objects having different attributes) in different (e.g. independently managed) servers in the loosely-coupled system. The plan does not attempt to produce names to overcome this likely scenario. That is, it does not attempt to produce a single namespace for all directory objects. (This issue is considered in more detail in Section 5.1.)

c) readily usable by LDAP clients and servers

As of this writing, a substantial number of the Lightweight Directory Access Protocol (LDAP) [6][7] implementations are currently available or soon will be. The names specified by this naming plan should be readily usable by these implementations and applications based on them.

d) friendly to re-use of existing Internet name registries

As described in Section 2 above, creation of new global name registries has been highly problematic. Therefore, a fundamental requirement this plan addresses is to enable the reuse of existing Internet name registries such as DNS names and RFC822 mailbox identifiers when constructing directory names.

e) minimally user-friendly

Although we expect that user interfaces of directory-enabled applications will avoid exposing users to DNs, it is unlikely that users can be totally insulated from them. For this reason, the naming plan should permit use of familiar information in name construction. Minimally, a user should be capable of recognizing the information encoded in his/her own DN. Names that are totally opaque to users cannot meet this requirement.

4.0 Name Construction

The paper assumes familiarity with the terminology and concepts behind the terms distinguished name (DN) and relative distinguished name (RDN) [2][8][9].

We describe how DNs can be constructed using three attribute types, domainComponent (dc), userID (uid) and commonName (cn). They are each described in turn.

4.1 Domain Component (dc)

The domain component attribute is defined and registered in RFC1274 [3][10]. It is used in the construction of a DN from a domain name. Details of the construction algorithm is described in "Using Domains in LDAP Distinguished Names" [1].

An organization wishing to deploy a directory following this naming plan would proceed as follows. Consider an organization, for example "Acme, Inc.", having the registered domain name "acme.com". It would construct the DN

dc=acme, dc=com

from its domain name. It would then use this DN as the root of its subtree of directory information.

The DN itself can be used to identify a directory organization object that represents information about the organization. The directory schema required to enable this is described below in section 5.2.

The subordinates of the DN will be directory objects related to the organization. The domain component attribute can be used to name subdivisions of the organization such as organizational units and localities. Acme, for example, might use the domain names "corporate.acme.com" and "engineering.acme.com" to construct the names

dc=corporate, dc=acme, dc=com
dc=engineering, dc=acme, dc=com

under which to place its directory objects. The directory schema required to name organizationalUnit and locality objects in this way is described below in section 5.2.

Use of this attribute for the RDN of directory objects of class "domain" is also possible [1].

4.2 User ID (uid)

The userid (uid) attribute is defined and registered in RFC1274 [3][10].

This attribute may be used to construct the RDN for directory objects subordinate to objects named according to the procedure described in Section 4.1. This plan does not constrain how this attribute is used.

4.3 Common Name (cn)

The commonName (cn) attribute is defined and registered in X.500 [3][11].

This attribute may be used to construct the RDN for directory objects subordinate to objects named according to the procedure described in Section 4.1. This plan does not constrain how this attribute is used.

4.4 Examples of uid and cn Usage

Although this plan places no constraints on the use of the uid and cn attributes for name construction, we would like to offer some suggestions by way of examples.

In practice, we have used uid for the RDN for person objects were we could make use of an existing registry of names and cn for other objects.

Examples of existing registries of identifiers for person objects are RFC822 mailbox identifiers, employee numbers and employee "handles". Aside from the convenience to administrators of re-use of an existing store of identifiers, if it is ever necessary to display to a user his/her DN, there is some hope that it will be recognizable when such identifiers are used.

We have found RFC822 mailbox identifiers a particularly convenient source for name construction. When a person has several e-mail addresses, one will be selected for the purpose of user identification. We call this the "distinguished" e-mail address or the "distinguished" RFC822 mailbox identifier for the user.

For example, if there is a user affiliated with the organization Acme having distinguished e-mail address J.Smith@acme.com, the uid attribute will be:

    uid=J.Smith@acme.com

The domain component attributes of a user's DN will normally be constructed from the domain name of his/her distinguished e-mail address. That is, for the user uid=J.Smith@acme.com the domain component attributes would typically be:

    dc=acme, dc=com

The full LDAP DN for this user would then be:

    uid=J.Smith@acme.com, dc=acme, dc=com

Directory administrators having several RFC822 identifiers to choose from when constructing a DN for a user should consider the following factors:

- Machine-independent addresses are likely to be more stable, resulting in directory names that change less. Thus an identifier such as:

      js@acme.com

may well be preferable to one such as:

js@blaster.third-floor.acme.com.

- Use of some form of "handle" for the "local" part that is distinct from a user's real name may result in fewer collisions and thereby lessen user pain and suffering. Thus the identifier:

  js@acme.com

  may well be preferable to one such as:

  J.Smith@acme.com

Practical experience with use of the RFC822 mailbox identifier scheme described here has shown that there are situations where it is convenient to use such identifies for all users in a particular population, although a few users do not, in fact, possess working mailboxes. For example, an organization may have a existing unique identification scheme for all employees that is used as a alias to the employees' real mailboxes -- which may be quite heterogeneous in structure. The identification scheme works for all employees to identify unambiguously each employee; it only works as an e-mail alias for those employees having real mailboxes. For this reason it would be a bad assumption for directory-enabled applications to assume the uid to be a valid mailbox; the value(s) of the mail attribute should always be checked.

It is important to emphasize that the elements of the domain name of an RFC822 identifier may, BUT NEED NOT, be the same as the domain components of the DN. This means that the domain components provide a degree of freedom to support access control or other directory structuring requirements that need not be mechanically reflected in the user's e-mail address. We do not want under any condition to force the user's e-mail address to change just to facilitate a new system requirement such as a modification in an access control structure. It should also be noted that while we do not require that the domain components match the RFC822 identifier, we DO require that the concatenated domain components form a registered domain name, that is, one that is represented in the DNS. This automatically avoids name conflicts in the directory hierarchy.

To provide an example of a DN which deviates from what might be considered the default structure, consider the following scenario.

Suppose that J.Smith needs to be granted special permissions to information in the dc=acme, dc=com part of the LDAP DIT. Since it

will be, in general, easier to organize special users by their name structure than via groups (an arbitrary collection of DNs), we use subdomains for this purpose. Suppose the special permissions were required by users in the MIS organizational unit. A subdomain "mis.acme.com" is established, if it does not already exist, according to normal DNS procedures. The special permissions will be granted to users with the name structure:

uid=*, dc=mis, dc=acme, dc=com

The DN of J.Smith in this case will be:

uid=J.Smith@acme.com, dc=mis, dc=acme, dc=com

In principal, there is nothing to prevent the domain name elements of the RFC822 identifier from being completely different from the domain components of the DN. For instance, the DN for a J.Smith could be:

uid=J.Smith@worldnet.att.net, dc=mis, dc=acme, dc=com

While we do not REQUIRE that the domain name part of the uid match the dc components of the directory distinguished name, we suggest that this be done where possible. At a minimum, if the most significant pieces of the DN and the uid are the same (i.e., "dc=acme, dc=com" and "acme.com") the likelihood, based on a knowledge of a user's e-mail address, of discovering an appropriate directory system to contact to find information about the user is greatly enhanced.

The example above represents a situation where this suggestion isn't possible because some of the users in a population have mailbox identifiers that differ from the pattern of the rest of the users, e.g., most mailboxes are of the form local@acme.com, but a subpopulation have mailboxes from an ISP and therefore mailboxes of the form local@worldnet.att.net.

5.0 Implementation Issues

5.1 Directory Services Considerations

We envision the deployment of LDAP-based directory services on the Internet to take the form of loosely coupled LDAP servers. This coupling will occur at two levels.

Firstly, LDAP servers will be loosely connected into islands (i.e. a set of servers sharing a single DN namespace). The glue connecting the islands will be LDAP referral [12] information configured into the LDAP servers. An LDAP search directed to any server in such an

island can be answered, if the information is not available to that server, by an LDAP referral to another, more appropriate server within the same island.

Secondly, various techniques will be used span LDAP islands. The concept that enables such techniques is the LDAP URL [13]. By combining a DNS host name and port (corresponding to one or more LDAP servers) with a DN, the LDAP URL provides unified high-level identification scheme (an LDAP URL namespace) for directory objects.

Because an LDAP referral is expressed as one or more LDAP URL, these two levels of coupling may not sharply distinguished in practice.

We do not envision the X.500 model of a single DIT (i.e. a single DN namespace) to be viable in an environment of competing service providers. This naming plan does not attempt to produce DNs to hide the possibility that a given real-world object may have independently managed directory objects (entries) associated with it.

5.2 Directory Schema Implications of the Naming Plan

The traditional directory schema(s) developed for the X.500 standard and its application to the Internet [4] require extension to be used with the naming plan developed here. The extensions described below attempt to reuse existing schema elements as much as possible. The directory objects for which extensions are required are: organization, organizational unit, and various classes of leaf objects. We describe the schema modifications below for organization, organizationalUnit and selected leaf classes.

So as to continue to use existing structural object classes to the extent possible, we propose supplementing entries based on these classes with additional information from two new auxiliary object classes, dcObject and uidObject. They are specified using the notation in Section 4 of [14].

The auxiliary object class dcObject is defined in "Using Domains in LDAP Distinguished Names" [1].

The auxiliary object class uidObject is defined as:

(OID-TBD NAME 'uidObject' SUP top AUXILIARY MUST uid)

In a pure X.500 context, our schema would also require the definition of new name forms and structure rules. These concepts are not required, however, for the specification of LDAP schemas.

5.2.1 Organization Schema

The dc attribute is employed to construct the RDN of an organization object. This is enabled by adding the auxiliary class dcObject to the organization's objectClass attribute.

5.2.2 Organizational Unit Schema

The dc attribute is employed to construct the RDN of an organizationalUnit object (which is subordinate in the DIT to either an organization or an organizationalUnit object). This is enabled by adding the auxiliary class dcObject to the organizational unit's objectClass attribute.

5.2.3 Person Schema

No schema extensions are required for person objects if either the inetOrgPerson [15] (preferred) or the newPilotPerson object classes are used. The attribute uid is permissible in each class. For consistency, the uidObject could be added to person entry objectClass attributes to assist applications filtering on this object class attribute value. Use of other classes for person objects with RDN constructed with the uid attribute such as organizationalPerson requires the use of the uidObject class.

It has been traditional in X.500 and LDAP directory services to employ the common name (cn) attribute in naming. While this naming plan doesn't require use of the cn attribute in naming, it should be stressed that it is a required attribute in any class derived from the person class and is still quite important. It will play a significant role in enabling searches to find user entries of interest.

5.2.4 Certification Authority Schema

The certification authority (CA) object class is an auxiliary class, meaning it is essentially a set of additional attributes for a base class such as organizationalRole, organization, organizationalUnit or person. Except in the case where the base structural class is inetOrgPerson, use of the uid attribute to construct the RDN of a CA will require the auxiliary class uidObject to permit the uid attribute to be used. In the cases where organizationalUnit or organization is the base class for a CA, use of the auxiliary class dcObject will permit the RDN of the CA to be a domain component.

5.2.5 Server and Server Application Schema

Servers and server applications are typically represented, for want of anything better, by entries of the object class applicationProcess

(or a class derived from it). Sometimes the class applicationEntity is used. In either case, the uid attribute should probably not be employed to construct the RDN of a server or server application object. The standard schema uses the attribute cn for such RDNs.

Suppose one wants to use this naming plan both in the construction of DNs for SSL server certificates and for their storage in a directory. It is customary for clients connecting via SSL to compare the server's domain name (e.g. from the URL used to contact the server) with the value of the cn attribute in the subject field (i.e. subject's DN) of the server's certificate. For this reason, it is common practice to set the cn attribute to the server's domain name.

The naming and schema to handle this situation is best explained by an example. Consider the server "host.acme.com". Following the algorithm in "Using Domains in LDAP Distinguished Names" [1], the DN dc=host, dc=acme, dc=com is constructed. To conform to the existing practices just described, the server's subject DN for the SSL server certificate should be cn=host.acme.com, dc=host, dc=acme, dc=com and the server's certificate should be stored in a directory entry with this name. This entry should use application process or application entity as its structural object class and strong authentication user as is auxiliary class.

6.0 Security Considerations

Although access controls may be placed on portions of the DIT to deny browse access to unauthorized clients, it may be possible to infer directory names and DIT structure in such sensitive portions of the DIT from the results of DNS queries. Providing public visibility to some portions of the DIT may assist those make such inferences.

7.0 Acknowledgments

This plan has emerged in the course of a number of fruitful discussions, especially with David Chadwick, John Dale, Joe Gajewski, Mark Jackson, Ryan Moats, Tom Spencer and Chris Tzu.

8.0 References

[1] S. Kille, M. Wahl, A. Grimstad, R. Huber, S. Sataluri, "Using Domains in LDAP Distinguished Names", Internet Draft <draft-ietf-asid-ldap-domains-02.txt>, August 1997.

[2] X.500: The Directory -- Overview of Concepts, Models, and Service, CCITT Recommendation X.500, December, 1988.

[3] P. Barker, and S. Kille, "The COSINE and Internet X.500 Schema", RFC1274, 11/27/1991.

[4] The North American Directory Forum, "A Naming Scheme for c=US", RFC1255, September 1991.

[5] The North American Directory Forum, "NADF Standing Documents: A Brief Overview", RFC 1417, The North American Directory Forum", NADF, February 1993.

[6] W. Yeong, T. Howes, and S. Kille, "Lightweight Directory Access Protocol", RFC1777, 03/28/1995.

[7] M. Wahl, T. Howes, and S. Kille, "Lightweight Directory Access Protocol (v3)", Internet Draft <draft-ietf-asid-ldapv3-protocol-04.txt>, March 1997.

[8] S. Kille, "A String Representation of Distinguished Names", RFC1779, 03/28/1995.

[9] M. Wahl, S. Kille, T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", Internet Draft <draft-ietf-asid-ldapv3-dn-03.txt>, April, 1997.

[10] M. Wahl, "A Summary of the Pilot X.500 Schema for use in LDAPv3", Internet Draft <draft-ietf-asid-schema-pilot-00.txt>, March 1997.

[11] M. Wahl, "A Summary of the X.500 User Schema for use with LDAPv3", Internet Draft <draft-ietf-asid-ldapv3schema-x500 01.txt>, July 1997.

[12] T. Howes, M. Wahl, "Referrals and Knowledge References in LDAP Directories", Internet Draft, <draft-ietf-asid-ldapv3-referral-00.txt>, May 1997.

[13] T. Howes, M. Smith, "The LDAP URL Format", Internet Draft, <draft-ietf-asid-ldapv3-url-04.txt>, August 1997.

[14] M. Wahl, A. Coulbeck, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", Internet Draft <draft-ietf-asid-ldapv3-attributes-07.txt>, August 1997.

[15] M. Smith, "Definition of the inetOrgPerson Object Class", Internet Draft <draft-ietf-asid-inetorgperson-01.txt>, July 1997.

12. Authors' Addresses

Al Grimstad
AT&T
Room 1C-429, 101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

EMail: alg@att.com

Rick Huber
AT&T
Room 1B-433, 101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

EMail: rvh@att.com

Sri Sataluri
Lucent Technologies
Room 4D-335, 101 Crawfords Corner Road
Holmdel, NJ 07733-3030
USA

EMail: srs@lucent.com

Steve Kille
Isode Limited
The Dome, The Square
Richmond
TW9 1DT
UK

Phone: +44-181-332-9091
EMail: S.Kille@isode.com

Mark Wahl
Critical Angle Inc.
4815 W Braker Lane #502-385
Austin, TX 78759
USA

EMail: M.Wahl@critical-angle.com