*by Jim DeRoest*

# Internet White Pages

***Jim DeRoest** has been involved (for better or worse) with IBM UNIX offerings from the IX/370 days, through PC/IX, AIX RT, AIX PS/2, AIX/370, PAIX, AIX/ESA and AIX V3. He is employed as an assistant director supporting academic and research computing at the University of Washington, and is the author of* AIX for RS/6000–System and Administration Guide *(McGraw-Hill). He plays a mean set of drums for the country gospel band Return. Email:* `deroest@cac.washington.edu`.

Who's who on the Internet and how do I find out? This is often the first question asked by new Internet users. The situation is akin to having a brand-new telephone installed in your home and discovering that you don't have a telephone directory. What good is a $20 per month connection to the world if you don't know how to contact anyone. Not to mention the fact that a brand-new PC and modem are a few orders of magnitude more expensive than the good old telephone you've been using for years. At least the phone company left you a 50-pound phone book when it installed your telephone.

Trying to find a user's mail address on the Internet is not a new problem. It has been around since the advent of electronic mail in the mid '70s when a store-and-forward protocol called UNIX-to-UNIX Copy Program (UUCP) was used to route mail around the network. Sure, you remember UUCP. To send a mail message, you had to include the names of all the machines that made up the "hop" path between you and the recipient in the mail address.

As UUCP was augmented by other network technologies, it resulted in some very strange-looking mail addresses. For example,

```
rita::IN%"beaver!lumpy!deroest@
uwavm.u.washington.edu"
```

Unfortunately, some of these nightmare addressing schemes are still with us. Try to explain an address like this to someone whose greatest technical accomplishment is mastering the television remote control.

Nobody wants a 50-pound email white pages directory to supplement all the other hard copy documents that came with the computer, telephone and microwave oven. Just give us Web-based tools that will allow us to do online searches using strings like "Jim DeRoest, Seattle, WA, USA" to locate a user's mail address, public key or Web page URL. The directory servers operating in the back room behind these tools should be able to intercommunicate to resolve information that may be located at some remote site. Resolving a directory request should work much the same way that the Internet Domain Name Service works when trying to resolve a domain

name to a particular IP address.

Over the years, a number of notable applications have been developed to address these requirements. You might be familiar with tools like `finger`, `ph` and `whois`. The latest white pages protocol craze being touted by Web application players such as Netscape Communications Corp. has its roots in the X.500 specification and is dubbed Lightweight Directory Access Protocol (LDAP).

## Lightweight Directory Access Protocol

Like its predecessors, LDAP provides a means for locating directory information about people, applications and services. Although somewhat open to interpretation, the specification is directed at providing white pages-type information as opposed to files and documents that are better suited to technologies such as Application Configuration Access Protocol (ACAP). I say "open to interpretation" because LDAP is easily extensible to serve additional data types and content.

The LDAP protocol (see RFC 1777) is based on the X.500 Directory Access Protocol (DAP) but does not incur the complexity or resource requirements inherent in X.500 DAP. Although streamlined, LDAP is interoperable with X.500 DAP clients and servers. LDAP runs over any TCP-based network.

As a directory protocol, LDAP defines a structure for accessing and managing a hierarchical database of attribute/value pairs (see RFC 1778). The LDAP directory hierarchy reflects the geographical and administrative structure that makes up the name space of a represented organization. LDAP directory attributes are descriptive objects associated with each object that make up the information hierarchy. Objects are content typed to support data interchange with external clients and servers.

Currently, there are two contenders for type representation in LDAP, Multipurpose Internet Mail Extensions (MIME) and Versit Personal Data Interchange, formerly eCard (see Table 2). Each technology has a number of proponents. The good news is that work is under way to promote interoperability between the two types.

## Distinguished Name

The group of attributes that make up an entry are collectively know as a Distinguished Name, or DN (see RFC 1779). A view of an entry at some subtree in the name space hierarchy is called the Relative Distinguished Name (RDN). The syntax that describes a particular DN in the name space is called the LDAP Data Interchange Format (LDIF), which looks very X.500-esque (see Table 1).

For example, my DN would look something like this:

```
"DN:CN="Jim DeRoest"
    OU="Computing & Communications"
    O="University of Washington"
    C=US
    UID=deroest
    EMAIL=deroest@cac.washington.edu
```

I know what you're thinking: To the end user, this syntax is about as pleasant as the ugly email address I showed you earlier. The idea here is that an LDAP client will hide this syntax from the end user. For example, a Web browser client might collect search strings from an HTML form, bundle it into LDIF format and append it to a URL that identifies an LDAP HTTP server (see RFC 1959):

```
ldap://ldap.washington.edu/O="University of
Washington", \
      OU="Computing & Communications",CN="Jim DeRoest"
```

Another option discussed in Netscape's white paper, "An Internet Approach to Directories" (see Table 2), describes an extension to the Internet Domain Name Service (DNS). A new DNS resource record called "DX" could be implemented to support mail client directory queries for email addresses. The DX record would function much like the current mail handler MX record by identifying the LDAP server that held the directory entry for the requested site.

**Table 1**
**X.520 Distinguished Name Keywords**

| Key | Attribute |
|---|---|
| CN | CommonName |
| L | LocalityName |
| ST | StateOrProvinceName |
| O | OrganizationName |
| OU | OrganizationalUnitName |
| C | CountryName |
| STREET | StreetAddress |

## LDAP Security

One of the nice features in LDAP is its adaptability to access control mechanisms for governing access rights to entries and attributes in an LDAP database. These mechanisms can include public key, X.509 certificate validation, SSL V4.0 access controls and Kerberos authentication. The LDAP implementation from the University of Michigan provides support for authentication via Kerberos V4.

LDAP Authentication and access control could be used to distribute administrative tasks for managing a directory name space to the various groups that make up an organization. One might also envision allowing individual users to update their common name information, address or store a public key.

Version 3 of the LDAP draft specification is looking into the issues required to support X.509 certificates. This includes string encoding for DN information embedded in a certificate and extensions to support certificate revocation lists (CRLs).

The Internet Engineering Task Force (IETF), although not working directly on X.500 issues, is addressing X.509 certificate infrastructure related to LDAP. Much of this work is

being done by the Access, Searching and Indexing of Directories (ASID) working group in the IETF (see Table 2).

## Directory API

RFC 1823 describes a C binding API called LDAP Application Programming Interface (LAPI), which is used to query and bind directory architectures. An LDAP implementation from the University of Michigan uses its own internal API called `slapi` to provide back-end interfaces to external databases. These include a UNIX shell interface (SHELL), which can be used to invoke custom shell scripts, a DBM interface (LDBM) and an interface for UNIX password files (PASSWD).

The LDBM interface can be used to access data sets implemented in `btree`, `hash`, GNU `dbm` or UNIX `ndbm` formats. These back ends facilitate layering LDAP over existing directory data sets. We are currently using the SHELL back-end interface here at the University of Washington in a pilot project to provide LDAP access to our locally developed accounting database, which comprises a name space of more than 70,000 user entries.

## Distributed Architecture

LDAP can be deployed within an organization as a set of master and slave servers to support replication for fault tolerance and to improve access performance. Each server runs a local `slapd` daemon to service LDAP client requests. A `slurpd` daemon is run to replicate the name space in distributed environments.

Subtrees that represent administrative subsets of the organization's name space can be distributed among master servers to provide local administrative control. Note that all updates to a subtree are handled by the master server that controls that portion of the name space. Writes are centralized, and reads are distributed among the servers.

A problem in current LDAP implementations is that there is no mechanism for referring directory queries to servers that represent the particular subsets of the name space designated in a request. A general request for "Jane Doe" could end up searching every server on the network. To address the problem of fixed referrals, a new system of "Forwarding Indexes" is being architected to refer non-local name space queries to the appropriate remote servers that administer requested portions of the name space. Each server makes a compressed version of its local index and makes it available to other servers to support these types of referrals. Note that these indexes require filtering support to facilitate those organizations that do not want to make their full directory space available to external searching.

## Implementations

As I mentioned earlier, Netscape is a proponent of LDAP. It has a very informative white paper that describes the protocol along with product devel-

## Table 2. LDAP Information

**Netscape Directory Server**
`http://partner.netscape.com/newsref/ref/ldap.html`

**University of Michigan**
`http://www.umich.edu/~rsug/ldap/`

**Stanford University**
`http://www-leland.stanford.edu/group/networking/directory/x500ldapfaq.html`
`http://www-leland.stanford.edu/~bbense/Inst.html`

**IETF ASID Charter**
`http://www.ietf.cnri.reston.va.us/html.charters/asid-charter.html`

**MIME FAQ**
`http://phonebk.duke.edu:8001/clients/mimefaq1.html`

**Versit Personal Data Interchange**
`http://www.versit.com/`
`http://www.imc.org/pdi/`

**Relevant RFCs**
RFC 1777 "Lightweight Directory Access Protocol"
RFC 1778 "The String Representation of Standard Attribute Syntaxes"
RFC 1779 "A String Representation of Distinguished Names"
RFC 1823 "The LDAP Application Program Interface"
RFC 1959 "An LDAP URL Format"

opment directions, available from its Web site (see Table 2). The Netscape Directory Server implementation closely follows the LDAP work done at the University of Michigan. This is not entirely surprising in that Netscape employs most of the original University of Michigan LDAP development team.

Take a look at the University of Michigan's Web site to see how LDAP is used to support browser access to white pages information (see Table 2). The directory name space is made up of more than 115,000 entries. This number is likely to be out of date by the time you read this so visit the University of Michigan's home page to get the current statistics on its LDAP configuration.

Another interesting use of LDAP can be found on Stanford University's Web site. The university is using LDAP query capabilities in `sendmail` Version 8.8 to resolve mail forwarding for its `sendmail` servers. The Stanford Web page describes the architecture along with a general FAQ and tutorial on LDAP (see Table 2). Note that these are just a few of the production LDAP implementations currently running on the Internet.

It's probably still a little early to see if LDAP will sweep the Web as the directory service of choice. There's a fair amount of work to be done in the specification, and there are other contenders that can provide this type of service. However, it's worth considering that LDAP does have some significant commitment from large players like Netscape, Novell Inc., Banyan Systems Inc. and IBM Corp. It never hurts to have some big sticks when negotiating new protocols in the Internet community.