

An e-Business Directory

Leveraging LDAP for an e-Business User Registry on iSeries | By Pat Fleming

Part two of this two-part series on LDAP examines its use as a user registry for iSeries e-business applications. Part one ran in the August iSeries Magazine e-Business Quarterly supplement.

Ever wonder what happens each time you add user information to an e-business network? The administrator must update information in more than one place (user registry) and content is limited to specific information about users. Each new application requires another information storage registry to be created because existing user registries can't accommodate expanding needs. Is it possible to add information without continuing to create registries? The answer is yes. This article delves into the details of how to address this problem using a user registry based on a lightweight directory access protocol (LDAP) directory.

User Registries

A centralized user registry, defined in an LDAP directory, can significantly increase administration efficiency. Information about existing users is stored and maintained in one location for use by all applications. For example, registering "John Smith" in the user registry may automatically enable this user to access Web-based applica-

tions. Additional information about this user also can be stored in the same directory entry for other uses.

Using an LDAP directory for a user registry provides advantages over a traditional database or text file. First, LDAP directories are hierarchical, allowing users to be organized based on business needs. For example, a directory entry for "John Smith" could be organized by the department John works in or by John's job type. Second, LDAP directories then can be used to authenticate the user. For example, the Web user "John Smith" can be authenticated by providing the LDAP server with his user ID and password. Finally, the directory permits different types of information about users to be managed. The directory entry for "John Smith" also could contain information about his Web experiences and preferences. Figure 1 (below) provides an

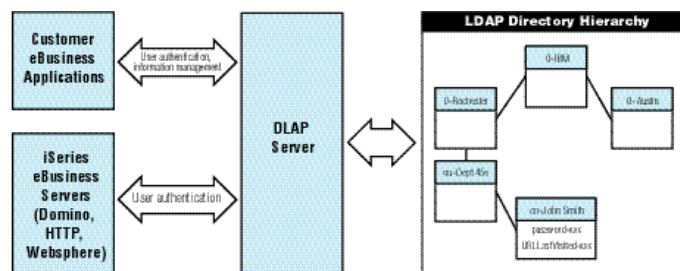
overview of a user registry implementation (authentication, hierarchical organization, application specific extensions) based on LDAP.

LDAP's Benefits

The key benefits of using an LDAP directory for a user registry include hierarchical organization of users, authentication of Web users and extensibility of user information.

All LDAP directory entry names are comprised of name components that form a hierarchy or tree structure. This type of structuring is similar to the Windows* and iSeries file systems where a path name is a hierarchy of folders and a file. In an LDAP directory, each directory entry can have additional directory entries, hierarchically arranged below it. For example, the top level of your directory hierarchy may be a company name followed by directory entries representing divisions,

Figure 1



departments and work groups. At each level, additional directory entries representing users may exist. This arrangement allows for easier user management. Because each level of the directory's hierarchy contains only a portion of the user registry, simple searches, such as listing all users within a specific department of a division, can be performed.

Referencing users in an LDAP directory can be done by specifying the distinguished name (DN) of the directory entry or referencing a specific identifier for the user, such as user ID. The format of a user's DN is defined by the X.500 standards and reflects the hierarchical structure of the user registry, much like a path name reflects the file system's hierarchy. For example, the DN for John Smith might look like "cn=John Smith, ou=dept 45e, ou=Rochester, o=IBM." Commas separate the name components of a DN. Each name component contains an attribute type name and value pair.

In this example, "cn=John Smith" is the first name component and identifies a directory entry that has a common name attribute type of "cn" containing a value of "John Smith." This particular DN also reflects that John Smith is a member of department 45e because this directory entry is located in the

Figure 3

```
import java.util.Hashtable;
import javax.naming.*;
import javax.naming.directory.*;

public class AuthenticateUser {
    public static void main(String[] args)
    {
        String userID="John Smith,ou=Rochester,o=IBM"; // example user ID
        String userPW="JohnPW"; // example PW
        //JNDI parameters are passed using environment properties
        Hashtable env = new Hashtable();
        env.put(Context.INITIAL_CONTEXT_FACTORY,"com.sun.jndi.Ldap.LdapCtxFactory");
        env.put(Context.PROVIDER_URL,"ldap://localhost:389");//LDAP server is on
        local system
        env.put(Context.SECURITY_AUTHENTICATION,"simple");// authenticate with user
        id /pw
        env.put(Context.SECURITY_PRINCIPAL,userID);
        env.put(Context.SECURITY_CREDENTIALS,userPW);
        try{
            Context ctx = new InitialContext(env); // attempt to authenticate the user
        } catch (AuthenticationException e) {
            System.out.println ("Successfully authenticated \" " + userID + "\"");
        } catch (CommunicationException e) {
            System.out.println ("Invalid user or password");
        } catch (NamingException e) {
            System.out.println ("LDAP server URL is invalid");
        } catch (UnexecutedError e) {
            System.out.println("Unexecuted error. " + e);
        }
    }
}
```

hierarchy under the directory entry "ou=dept45e." The attribute type "ou" contains a value of "dept 45e," the organizational unit name. From this DN we also determine that dept 45e is located at an organizational unit (ou) called Rochester and is part of the organization (o) IBM.

With some applications, such as Domino*, HTTP Server and WebSphere* Application Server, a user ID is used instead of a DN. A user ID, such as "JSmith," is a simpler reference to the user and allows users to more conveniently authenticate to these applications. The user ID is

stored in the user's directory entry, usually in the attribute type "uid".

Using LDAP to Authenticate Users

The design and development of e-business applications can be simplified by utilizing the security features of LDAP. Instead of an application maintaining its own list of valid users and performing its own authentication, the application could utilize the capabilities of a user registry built on an LDAP directory. Figure 2 and Figure 3 (page e2) show two authentication methods used by e-business applications for iSeries.

Applications can authenticate users by having them supply their DN and corresponding password during a bind operation with the LDAP server, which verifies that a directory entry exists for the user (valid user) and that the password is correct (authenticate the user). The code examples show how to authenticate the user "John Smith" using C and Java* programs.

When a user ID is used instead of the fully qualified DN, the application must first be configured to a specific portion of the directory

Figure 2

```
#include <ldap.h>
main()
{
    LDAP *ld;
    /* initialize a connection to the LDAP server */
    if ((ld = ldap_init("myiSeries.rochester.ibm.com",LDAP_PORT))!=NULL) {
        /* attempt to bind to the LDAP as the user as a way of
        authenticating the user*/
        if (ldap_simple_bind_s(ld, "cn=John Smith,ou=
        Rochester,o=IBM,"JohnPW")=LDAP_SUCCESS){
            /*John Smith has been authenticated*/
        }
        else{
            /*John Smith was not authenticated.
            Either he doesn't exist in the user registry or
            his password is invalid*/
        }
    }
    else{
        /*Unexpected error initializing a connection*/
    }
}
```

hierarchy in order to find the users. This is done by configuring the application's base DN, such as "ou=Rochester,o=IBM." The application performs a search, starting at the location specified, in the base DN, for all users with the ID specified by that user. The LDAP server returns a list of DNs that match the user ID. If the administrator has maintained unique user IDs in the user registry and that user has specified the correct user ID, the application uses the DN returned in the list to authenticate the user.

Getting the Most From Your User Registry

Most e-mail clients, such as Lotus Notes*, Netscape Communicator and Microsoft Outlook Express can be configured to use LDAP directories as address books. To leverage this, simply add e-mail addresses to your user registry and configure the mail client to use your LDAP directory as an address book.

To help get you started with a user registry, iSeries provides utilities that allow you to publish existing iSeries user information, including mail addresses, from iSeries system distribution directory (SDD) entries to an LDAP directory. Because the iSeries user registry is based on an LDAP directory, it can be extended with additional information needed by e-business applications. The user registry can help personalize a Web site by maintaining information about users. An example of this would be a human resource application being used to hold employee information.

Reaching Out With Your User Registry

To demonstrate how a user registry can be extended, we'll extend the user registry to accommodate personalized information about Web users. This modification allows Web

Figure 4

The LDIF file contains two records, separated by a blank line and starting with "dn:=schema" to indicate that the LDIF object to be modified is the server's schema. Note in each record, the line containing "xxx-oid" specifies a sample object identifier (OID) and should be replaced by a valid, dotted numeric OID. The first record creates a new attribute type, URLLastVisited with a syntax of string (represented by the OID). The second record creates a new object class, webUser, which specifies URLLastVisited as an optional attribute. webUser is defined as an auxiliary class, which allows it to be used with other object classes, such as inetOrgPerson.

```
dn:=schema
changetype: modify
add: attributetypes
attributetypes: (
  URLLastVisited-oid
  NAME 'URLLastVisited'
  DESC 'Contains the URL last visited by the Web user.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  USAGE userApplications
)

dn:=schema
changetype: modify
add: objectclasses
objectclasses: (
  webUser-oid
  NAME 'webUser'
  DESC 'Defines entries representing Web users.'
  SUP top
  AUXILIARY
  MAY (URLLastVisited)
)
```

site specialization.

The user registry consists of directory entries, where each directory entry represents an individual user. Each directory entry is defined by one or more object classes. Each object class is comprised of one or more attribute types and their corresponding values.

LDAP servers use a schema to define what can be stored in a directory entry. The schema is defined in LDAP server configuration text files and provides a list of attribute type definitions and object classes. To extend the user registry, modify these schema files by adding new attribute types and object classes. Starting with OS/400* V4R5, these modifications can be done using the LDAP utility ldapmodify and an LDAP Data Interchange File (LDIF) containing new schema definitions.

A typical schema for user-related information consists of, at the least, the 'inetOrgPerson' object class, which includes attribute types such as cn, sn (surname) and givenName. The example includes directory entries in the user registry using two object classes—inetOrgPerson and webUser. webUser is an object class

that is defined for extending the user registry to contain Web-specific information about users. See Figure 4 (above) for an example of an LDIF file containing one new attribute type, URLLastVisited and one new object class, webUser.

The LDIF file shown in Figure 4 can be used with the LDAP ldapmodify utility, which is run from either the Windows or Qshell command line. Refer to the iSeries Information Center Web site (www.iseries.ibm.com/infocenter) for details on using ldapmodify and formatting an LDIF file.

In Summary

Establishing a user registry for your e-business can save you money and time. An LDAP directory, with its hierarchy, security and extensibility, can provide a powerful and flexible user registry. **i**

Pat Fleming is an iSeries senior software engineer in Rochester, Minn. He is currently an IBM WebSphere Application Server for iSeries architect. Prior to that, he was an OS/400 Directory Services (LDAP) architect. He can be reached at flemingp@us.ibm.com.*