Variable Support for Apple’s LDAPv3 Plug-in

Dan Sinema
Agenda

- LDAP - Brief Introduction
- Mac OS X Architecture
- Mac OS X LDAP Plug-in
- What my project does
- Demonstration
- Q & A
LDAP - A Brief Introduction

- What is a Directory?
- Derived from X.500 (DAP)
- LDAP basic terms
What is a Directory?

“Fundamentally, what a directory service does is securely manage complex systems of interrelated information, and support the widespread distribution and speedy retrieval of that information—any information” (Sheresh & Shersh, 2001, p. 7).
Qualities of a Directory

- A defined namespace
- An extended search capability
- Authentication and access control
- Scales from small to large networks
- A datastore optimized for reads

(Sheresh & Sheresh, 2001, p.13)
Derived from X.500 (DAP)

- Directory Access Protocol (DAP)
- Originally standardized by ISO and ITU in 1988
- X.500 is an enormous standard
- Utilizes the OSI stack
- Costly to implement
LDAP

- Lightweight Directory Access Protocol
- LDAPv1 published as rfc 1487 in 1993 by the IETF and ISODE at UofMich
- Lower overhead*
- TCP/IP based*
- Widely accepted API*
- Uses the DNS namespace*

* (Sheresh & Sheresh, 2001, p.165)
LDAP Basic Terms

- Schema
- Object Class
- Attributes
Real Quick, What is an OID?

“An OID (Object Identifier) is a globally unique identifier for objects and attributes assigned by various international standards organizations including American National Standards Institute (ANSI) and the Internet Assigned Numbers Authority (IANA).” (Sheresh & Sheresh, 2001, p. 175)

Example: 1.3.6.1.4.1.4203
Sometimes compared to a map

Description of objects and attributes

attributetype ( 2.5.4.10 NAME ( 'o' 'organizationName' )
  DESC 'RFC2256: organization this object belongs to'
  SUP name )

attributetype ( 2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
  DESC 'RFC2256: organizational unit this object belongs to'
  SUP name )
Object Class

Similar to a class in C++ or Java

Used to describe objects in general terms

Meta object descriptor

```plaintext
objectclass  ( 2.16.840.1.113730.3.2.2
    NAME 'inetOrgPerson'
    DESC 'RFC2798: Internet Organizational Person'
    SUP organizationalPerson
    STRUCTURAL
    MAY ( 
        audio $ businessCategory $ carLicense $ departmentNumber $ 
        displayName $ employeeNumber $ employeeType $ givenName $ 
        homePhone $ homePostalAddress $ initials $ jpegPhoto $ 
        labeledURI $ mail $ manager $ mobile $ o $ pager $ 
        photo $ roomNumber $ secretary $ uid $ userCertificate $ 
        x500uniqueIdentifier $ preferredLanguage $ 
        userSMIMECertificate $ userPKCS12 )
  )
```
Attribute

Similar to data members in C++ and Java

Gives personality to Object Class

# employeeNumber
# Numeric or alphanumeric identifier assigned to a person, typically based
# on order of hire or association with an organization. Single valued.
attributetype ( 2.16.840.1.113730.3.1.3
   NAME 'employeeNumber'
   DESC 'RFC2798: numerically identifies an employee within an organization'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringsMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
   SINGLE-VALUE )
LDAP is Industry Standard

- Sun/Netscape SunONE (formerly iPlanet)
- Novell eDirectory
- Microsoft Active Directory
- OpenLDAP
<table>
<thead>
<tr>
<th></th>
<th>Aqua</th>
<th>AppleScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cocoa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carbon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quartz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenGL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QuickTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Darwin</td>
<td>Open Desktop</td>
<td></td>
</tr>
</tbody>
</table>
Mac OS X Directory Services

- DirectoryService daemon
- Plug-in structure
- Defines Nodes, Standard Record Types, Standard Attributes

(Apple, 2002)
Directory Services Cont.

- lookupd, used for UNIX compatibility
- NetInfo, legacy directory
- LDAP

(Apple, 2002)
Mac OS X LDAPv3 Plug-in

- Use Mac OS X Directory Services API
- OpenLDAP API (Wrapped in the LDAP Framework)
- Open sourced under the APSL (Apple Public Source License)
Features of LDAPv3 Plug-in

- Map LDAP objects and attributes to local objects and attributes
- Static assign values of attributes
Map LDAP to Local
Static Mappings

“#” signifies that the value is a static mapping.

The standard plug-in applies the same value to all users that login.
What My Project Does.
Adding Variable Support to the Static Mappings

- Administrators can customize the static mappings on a per user basis.
- Allows the use of the directory as-is, modifications are not required.
- Tokens use LDAP attribute names encased by ‘$’ Example: $uid$ The user would then be looking for the value of the “uid” attribute on the LDAP Server.
What My Code Does

LDAP Request - User = testuser

LDAP Reply - Data Returned from LDAP Server

- dn: uid=testuser, cn=users, dc=sinemas, dc=net
- uid: testuser
- cn: Test User
- gidNumber: 20
- uidNumber: 1029
- homeDirectory: /Network/Servers/www/Users/testuser
- objectClass: inetOrgPerson
- objectClass: posixAccount
- objectClass: shadowAccount
- objectClass: apple-user
- objectClass: extensibleObject

How it works

1. NFSHomeDirectory is mapped to #/tmp/$uid$
2. Finds ‘$'
3. Finds “uid” is requested attribute
4. Locates “uid” value from LDAP Server
   - uid: testuser
5. Replace “$uid$” with “testuser”
6. Final value of NFSHomeDirectory is - #/tmp/testuser
Q & A
Demonstration
References
