



JavaOneSM
Sun's 2002 Worldwide Java Developer Conference™

Java Naming and Directory InterfaceTM (JNDI)

Rosanna Lee
Vincent Ryan
Scott Seligman

Java Security, Networking, & Naming
Sun Microsystems, Inc.

Overview

- Secure LDAP access
- JNDI and Active Directory
- JNDI and DNS
- Future Directions

Secure LDAP Access

- Security features provide support for authentication, privacy, and integrity of directory access
- Interoperable with new security features in Windows .NET Server, iPlanet Directory Server



LDAP: Authentication

- Support for Simple Authentication and Security Layer (SASL) authentication mechanisms
 - Digest–MD5 (RFC 2831) SASL Authentication
 - GSS–API/Kerberos v5 SASL Authentication
 - PKI Certificate–based authentication via the "EXTERNAL" SASL mechanism
- Clear–text authentication inside encrypted channel (for example, SSL/TLS)



LDAP: Privacy and Integrity

- LDAP over SSL
- Start TLS extension (RFC 2830)
- Negotiated SASL security layer
 - Digest–MD5
 - GSS–API/Kerberos v5



JNDI & Active Directory

- LDAP Controls
- LDAP Extensions
- LDAPAuthentication
- LDAP Schema



LDAP Controls & Active Directory

- Published Controls
 - Server Side Sorting (RFC 2891)
 - Paged Results (RFC 2696)
 - Virtual List View
 - Tree Delete
 - Directory Synchronization
- Requires LDAP Booster Pack
 - `com.sun.jndi.ldap.ct1` package



LDAP Controls & Active Directory (continued)

- 11 Proprietary controls
 - Use the `com.sun.jndi.ldap.BasicControl` class
- Code Sample:

```
LdapContext ctx =  
    new InitialLdapContext(env, null);  
  
// activate the Show Deleted Object control  
ctx.setRequestControls(new Control[]{  
    new BasicControl("1.2.840.113556.1.4.417")});  
  
// perform search
```



Code Sample: Server Side Sorting

```
LdapContext ctx = new InitialLdapContext(env, null);

// sort results by their cn attribute
ctx.setRequestControls(new Control[]{
    new SortControl(new String[]{"cn"}, true)});

// list the current context
NamingEnumeration results = ctx.list("");
While (results != null && results.hasMore()) {
    System.out.println(
        ((NameClassPair)results.next()).getName());
}
```



Code Sample: Server Side Sorting (continued)

```
Control[] respCtrls;  
  
// examine response controls (if any)  
if ((respCtrls = ctx.getResponseControls()) != null) {  
    for (int i = 0; i < respCtrls.length; i++) {  
        if (respCtrls[i] instanceof SortResponseControl){  
            SortResponseControl src =  
                (SortResponseControl)respCtrls[i];  
            if (! src.isSorted()) {  
                System.out.println("sort error: " +  
                    src.getResultCode());  
                throw src.getException();  
            }  
        }  
    }  
}
```



LDAP Extensions & Active Directory

- Start TLS (RFC 2830)
 - Requires J2SDK v1.4
 - Requires Windows .NET Server
- LDAP over SSL
- SSL session re-use issue:
 - Broken in Windows 2000 Server
 - Fixed in Windows .NET Server (Beta 3)



Code Sample: Start TLS

```
LdapContext ctx = new InitialLdapContext(env, null);

// activate Start TLS
StartTlsResponse tls =
    (StartTlsResponse) ctx.extendedOperation(
        new StartTlsRequest());

// establish the TLS connection
SSLSession sslSession = tls.negotiate();

// perform protected LDAP operations

tls.close(); // close the TLS connection

// perform unprotected LDAP operations
```



LDAP Authentication & Active Directory

- SASL GSSAPI (Kerberos v5)
 - Set user account properties
 - Use DES encryption keys for this account
 - Do not require Kerberos pre-authentication
 - Reset user password
- SASL EXTERNAL
 - Requires Windows .NET Server
- SASL DIGEST-MD5
 - Requires Windows .NET Server



LDAP Schema & Active Directory

- LDAP Schema
 - RFC 2252 format
 - Read-only
 - Accessible only to authenticated users
 - `cn=Aggregate,cn=Schema,cn=Configuration,dc=example,dc=net`
- Warning: some standard attributes have been redefined
 - E.g., `cn` attribute is defined as single-valued



LDAP Schema & Active Directory (continued)

- Internal Schema
 - Proprietary format
 - Tree structure
 - Read/write – no undo
 - Requires Registry setting to permit modifications
 - Schema Administrators group
 - `cn=Schema,cn=Configuration,dc=example,dc=net`
- Schema tools in LDAP Service Provider v1.2.4
 - `CreateJavaSchema`, `CreateCorbaSchema`



DNS Service Provider

- Access all DNS data via JNDI API
- Find all resource records for a domain, or only a particular type ("A", "MX", *etc.*)
- List all children of a domain
 - if supported by the DNS server
- May request authoritative information only
- Read-only (at this time)
- In J2SDK 1.4 and available unbundled



DNS Data Mapping

- DNS nodes ("domains") represented by JNDI directory contexts
- Resource records represented by JNDI attributes
 - IN A 192.168.0.1
IN MX 10 sun.com
 - attribute "A" with value "192.168.0.1"
attribute "MX" with value "10 sun.com"



DNS Properties

- Typically set within program, or in resource file within JAR

```
java.naming.factory.initial=  
    com.sun.jndi.dns.DnsContextFactory
```

```
java.naming.provider.url=  
    dns://192.168.1.1/sun.com
```

```
// Use sparingly
```

```
java.naming.authoritative=true
```



Code Sample: DNS via JNDI

```
import javax.naming.directory;  
...  
ctx = new InitialDirContext();  
  
// Look for all records of host "www".  
attrs1 = ctx.getAttributes("www");  
  
// Look for MX records of host "mailhost".  
attrs2 = ctx.getAttributes("mailhost",  
                             new String[] {"MX"});  
  
// List all hosts in the "sfbay" domain.  
hosts = ctx.list("sfbay");
```



DNS Provider for InetAddress

- Bypass native platform's host name resolution and use JNDI to access DNS directly
- System properties
 - `sun.net.spi.nameservice.provider.1 = dns,sun`
 - `sun.net.spi.nameservice.nameservers = 192.168.1.1`
 - `sun.net.spi.nameservice.domain = sun.com`
- Example
 - `addr = InetAddress.getByName("host1");`



Future Directions

- Ensure conformance to LDAP bis standards
- Promote classes for standard LDAP controls to `javax.naming.ldap` package
- Support connection pooling
- Improve manageability and monitoring of the LDAP and DNS providers
- Support automatic location of LDAP servers via DNS



Future Directions (continued)

- Support automatic location of DNS servers
- Support Dynamic DNS (updates)
- Support Secure DNS
- FCS DSML v1 provider
- Support for DSML v2
- Support for manipulating LDAP names and URLs
- Incorporate use of standard Java SASL API (JSR 28)



Q&A

Session BOF-3196, Java Naming and Directory Interface™ (JNDI)



Further Information on JNDI

- JNDI Home Page
 - <http://java.sun.com/jndi/>
- The JNDI Tutorial
 - <http://java.sun.com/jndi/tutorial/>
- JNDI-INTEREST mailing list & archive
 - <http://java.sun.com/jndi/staytouch.html>

