Introduction to JNDI

AIS Developer Documentation

AIS Training – 2007

MICHIGAN STATE UNIVERSITY

What is JNDI?

- Java Naming and Directory Interface (runs inside of JBoss, application server)
- In general, naming servers provide a way of associating names with objects
  - DNS – Domain Name Service
  - LDAP – Lightweight Directory Access Protocol
  - CORBA – Common Object Request Broker Architecture
  - NIS – Network Information Service
- JNDI provides the Java API to access naming and directory services

What is a Naming Service?

- Provides a way to map a human friendly name with a computer friendly object
  - File handles
  - IP addresses
  - Database connections
  - Distributed objects/services
- Association of a name with an object is called a “binding”

Why JNDI?

- Provides location independence
    - The client is independent of where the service is running
    - The naming service can be configured externally to the client
    - Clients do not have to know where remote objects are located
- Clients have a common API
- Works with a transport mechanism. The default mechanism is RMI.

A Hierarchical Namespace

- Names are usually hierarchical like file systems
  - /d6501/service/sentinel
- Each part is a “node” - /d6501/service/sentinel
  - d6501
  - service
  - sentinel
- Nodes in the “tree” are referred to as Contexts
  - Think “directories” of a file system
  - Objects can be bound to them
  - The same object can be bound to multiple contexts

javax.naming.Context

- Provides the methods for accessing and manipulating JNDI trees
- Common methods
  - lookup – locate an object by name
  - bind – insert an object in the namespace
  - rebind – associate an object with an existing name
  - unbind – remove a binding from the namespace
  - list – iterate through the children of a node
- InitialContext – subclass that provides our starting point into a naming tree (i.e. root directory)

```
Context ctx = new InitialContext();
Object  obj =
ctx.lookup( “/d6501/service/sentinel” );
```

How does it work?

- InitialContext gets information on naming service from system properties

  ```
  java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
  java.naming.provider.url=localhost:1099  (note for RMI AIS is using :8099 )
  ```

- They can also be provided by a file named “jndi.properties” which a J2EE client will use to bootstrap JNDI
- You can override them by passing a Java Properties collection containing them to the InitialContext constructor

  ```
  Properties prop = new Properties();
  prop.setProperty( “java.naming.factory.initial”,
                 “org.jnp.interfaces.NamingContextFactory” );
  prop.setProperty( “java.naming.provider.url”, 
  “localhost:1099” );
  Context ctx = new InitialContext( prop );
  Object  obj = 
  ctx.lookup( “/d6501/service/sentinel” );
  ```

Binding Objects

- Intermediate contexts must be created first

```
Context ctx = new InitialContext( prop );
Object  obj =
ctx.lookup( “/d6501/service/sentinel” );
ctx = ctx.createSubContext( “d6501” );
ctx = ctx.createSubContext( “service” );
ctx.bind( “sentinel”, sentinelService );
```

JNDI and JEE

- JEE containers implement Naming Services (JBossNS)
- Used by components (EJBs and Servlets) to access their “local environment”
  - Resources (data sources, etc.)
  - EJB references
  - Environment entries
- Applications can bind objects automatically via deployment descriptors or Annotations (Java 1.5)
  - web.xml and ejb-jar.xml declare an object to be bound to the local environment
  - jboss-web.xml and jboss.xml link the local declaration to the JNDI tree
- Accessed through the java:comp/env naming context

Java Management Extensions JMX

- JMX technology provides a way to manage resources such as applications, devices, and services.
- If you have something deployed, you can see what’s deployed and what’s bound to that name.
- In-class exercise to look at the JMX-Console. Start server, use browser to go to localhost:8080/jmx-console. (Note: need user ID and password to login)
  - Under JBoss section, click on service=JNDIView (You will get a list of MBean Operations.)
  - Under java.language.String list(), click invoke button (You will get a Web Applications list. Scan all the different namespaces. Global JNDI Namespace is at bottom.)
- Discuss where to find the JBoss properties including user ID and password. (c:\home2\jboss\server\ais\conf\props)