Packaging and Deploying a J2CA Connector

Valerie Pressley
Hewlett-Packard
Review – J2EE Architecture
Overview

- Connector = Resource Adapter
- Environment
  - Managed
  - Non-managed
Packaging and Deployment

- Packaging
  - Resource Adapter Archive
  - .rar extension

- Deployment
  - Software module
  - Operational environment
Packaging
Connector Components

- Classes that implement connector functionality
- Utility Java classes
- Platform-dependent native libraries
- Help files and documentation
- XML Deployment Descriptors
Connector Functionality Classes

- Client-level API
  - CCI API or
  - Adapter-specific API
- Classes that implement J2CA contracts
  - Connection Management
  - Transaction Management
  - Security
Deployment Descriptors

- ra.xml
- weblogic-ra.xml
ra.xml

- General Information
- Interface and Implementation Classes
- Transaction Support Level
- Authentication Information
- Security Permissions
- Configuration Properties
General Information

- Connector name
- Connector description
- URI of a UI icon for the connector
- Name of the vendor who provides the connector
- Licensing requirement and description
- Type of the EIS system supported
- Version of the connector architecture specification supported by the connector
- Version of the connector
Example

<connector>
  <display-name>My Connector</display-name>
  <vendor-name>Hewlett Packard</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>Oracle</eis-type>
  <version>2.0</version>
  <license>
    <license-required>false</license-required>
  </license>
</connector>
Interface and Implementation Classes

- ManagedConnectionFactory class
- ConnectionFactory interface and implementation class
- Connection interface and implementation class
Example

```xml
<managedconnectionfactory-class>
    connector.SampleManagedConnectionFactory
</managedconnectionfactory-class>
<connectionfactory-interface>
    javax.resource.cci.ConnectionFactory
</connectionfactory-interface>
<connectionfactory-impl-class>
    connector.SampleConnectionFactory
</connectionfactory-impl-class>
<connection-interface>
    javax.resource.cci.Connection
</connection-interface>
<connection-impl-class>
    connector.SampleConnection
</connection-impl-class>
```
Transaction Support Level

- **NoTransaction**: The connector does not support resource manager local or JTA transactions

- **LocalTransaction**: The connector supports resource manager local transactions

- **XATransaction**: The connector supports both resource manager local and JTA transactions
Example

```xml
<resourceadapter>
  
  <transaction-support>
    NoTransaction
  </transaction-support>
  
  
</resourceadapter>
```
Authentication Information

- **Authentication Mechanism**
  - BasicPassword
  - Kerbv5

- **Credential Interface**
  - PasswordCredential
  - GenericCredential

- **Reauthentication support**
Example

```xml
<resourceadapter>
  ..
  ..
  <authentication-mechanism>
    <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
    <credential-interface>javax.resource.spi.security.PasswordCredential</credential-interface>
  </authentication-mechanism>
  <reauthentication-support>false</reauthentication-support>
  ..
  ..
</resourceadapter>
```
Security Permissions

- Grant statements for permissions
- Specify permissions other than default permissions of application server
Example

```xml
<security-permission>
  <security-permission-spec>
    grant { permission java.net.NetPermission "specifyStreamHandler"; };
  </security-permission-spec>
</security-permission>
```
Configuration Properties

Properties:
- Name
- Type
- Description
- Default value (optional)

Per ManagedConnectionFactory instance
Well-Defined Properties

<config-property-name>ServerName</config-property-name>
<config-property-name>PortNumber</config-property-name>
<config-property-name>UserName</config-property-name>
<config-property-name>Password</config-property-name>
<config-property-name>ConnectionURL</config-property-name>
Example

<config-property>
  <description>The port number the database is listening on</description>
  <config-property-name>PortNumber</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>1521</config-property-value>
</config-property>
**weblogic-ra.xml**

- Connection pool and logging parameters
- Security Principal Mapping
- Configuration Property Mapping
- Additional elements

*Specific to BEA WebLogic Server*
Example (NEW)

```xml
<weblogic-connection-factory-dd>
  <connection-factory-name>SampleConnector.rar</connection-factory-name>
  <jndi-name>SampleConnector</jndi-name>
  <pool-params>
    .
    .
    .
  </pool-params>
  <security-principal-map>
  </security-principal-map>
</weblogic-connection-factory-dd>
```
Connection Pool

- initial-capacity
- max-capacity
- capacity-increment
- shrinking-enabled
- shrink-period-minutes
- connection-cleanup-frequency
- connection-duration-time
- connection-maxidle-time
- connection-profiling-enabled
Example (NEW)

```
<weblogic-connection-factory-dd>

  .

  .

  <pool-params>
    <initial-capacity>0</initial-capacity>
    <max-capacity>1</max-capacity>
    <capacity-increment>1</capacity-increment>
    <shrinking-enabled>false</shrinking-enabled>
    <shrink-period-minutes>200</shrink-period-minutes>
  </pool-params>

  .

  .

</weblogic-connection-factory-dd>
```
Logging

- logging-enabled
- log-filename
Example

<logging-enabled>true</logging-enabled>
<log-filename>intellibo_ra.log</log-filename>
Security Principal Mapping

- security-principal-map
Example

<security-principal-map>
  <map-entry>
    <initiating-principal>raruser</initiating-principal>
    <initiating-principal>javajoe</initiating-principal>
    <resource-principal>
      <resource-username>scott</resource-username>
      <resource-password>tiger</resource-password>
    </resource-principal>
  </map-entry>
</security-principal-map>
Configuration Property Mapping

- map-config-property
Example

<map-config-property>
  <map-config-property-name>ConnectionURL</map-config-property-name>
  <map-config-property-value>jdbc:cloudscape:demo;create=true</map-config-property-value>
</map-config-property>
Additional Elements

- connection-factory-name
- jndi-name
Example

```xml
<connection-factory-name>
    SampleConnectionFactory
</connection-factory-name>
<jndi-name>
    eis/SampleConnector
</jndi-name>
```
Default \texttt{weblogic-ra.xml}

\begin{verbatim}
<weblogic-connection-factory-dd>
  <connection-factory-name>
    __TMP_CFNAME_.\config\mydomain\applications\whitebox-notx.rar</connection-factory-name>
  <jndi-name>
    __TMP_JNDINAME_.\config\mydomain\applications\whitebox-notx.rar</jndi-name>
  <pool-params>
    <initial-capacity>0</initial-capacity>
    <max-capacity>1</max-capacity>
    <capacity-increment>1</capacity-increment>
    <shrinking-enabled>false</shrinking-enabled>
    <shrink-period-minutes>200</shrink-period-minutes>
  </pool-params>
  <security-principal-map>  </security-principal-map>
</weblogic-connection-factory-dd>
\end{verbatim}
Sample Directory Structure

/META-INF/ra.xml
/META-INF/weblogic-ja.xml
/howto.html
/images/icon.jpg
/ra.jar
/ccj.jar
/win.dll
/solaris.so
Packaging the .rar

- Create a temporary staging directory anywhere on your hard drive.
- Compile or copy the resource adapter Java classes into the staging directory.
- Create a JAR to store the resource adapter Java classes. Add this JAR to the top level of the staging directory:
  ```
  jar cvf jar-file.jar -C staging-dir
  ```
Packaging the .rar (Continued)

- Create a META-INF subdirectory in the staging directory.
- Create an ra.xml deployment descriptor in the META-INF subdirectory and add entries for the resource adapter.
- Create a weblogic-ra.xml deployment descriptor in the META-INF subdirectory and add entries for the resource adapter.*

*Specific to WebLogic Server
Packaging the .rar (Continued)

- Create the RAR with a JAR command:
  - `jar cvf jar-file.rar -C staging-dir`
Deploying
Deployment Approaches

- **Standalone Deployment**
  - Resource adapter deployed by itself

- **Bundled Deployment**
  - Resource adapter assembled into a J2EE application
  - Resource adapter deployed as part of the J2EE application
**Standalone Deployment**

- Multiple applications share one connector
- Disadvantages:
  - Deployer must resolve external dependencies
  - Deployer must deploy correct connector version
Bundled Deployment

- Connector deployed inside an application
- Connector visible only to components in application
- No external connector dependencies to resolve
- Guarantees correct version of connector
Deploying a Connector

A Connector (also called a resource adapter) module is a system-level software driver used by an application server such as WebLogic Server to connect to an Enterprise Information System (EIS). A resource adapter serves as the "J2EE connector." Resource adapters contain the Java, and if necessary, the native modules required to interact with the EIS.

The deployment of a Connector module is similar to the deployment of Web application modules, EJB modules, and Enterprise Applications. Like these deployment units, you can deploy a Connector module in an exploded directory format or as a RAR (Resource Adapter Archive) file. Configuring and deploying a Connector module in a WebLogic Server domain enables WebLogic Server to serve the modules of the resource adapter to clients.

To configure a Connector module, click the "Deploy a new Connector Module..." link.
Deploying a Connector 

(Continued)

Deploy a Connector Module

Select the archive for this Connector module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your Connector module contains valid descriptors.

Location: \localhost\C\bea\user_projects\domains\ovsidomain

- applications
- connectors
- myserver

Target Module
Deploying a Connector (Continued)

Deploying a Connector Module

Select the archive for this Connector module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and confirm your Connector module contains valid descriptors.

Location: localhost \ C:

- bea
- dblib
- chetool
- crmdcons
- Compaq
- connectors
- cygwin
- data
- DM
- Documents and Settings
Deploying a Connector (Continued)

Deploy a Connector Module

Select the archive for this Connector module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should upload your file(s) and/or confirm your Connector module contains valid descriptors.

Location: localhost1, 1 connectors
- Gen-Oracle-Connector.rar
- SampleConnector.rar
- StoredProcedureConnector.rar

Target Module
Deploying a Connector (Continued)

Deploy a Connector Module

Review your choices and deploy

Deployment Targets

Your Connector module will be deployed to the following locations:

```
StoredProcedureConnector will be deployed to
Servers - myserver
```

Source Accessibility

Since this is a single server environment, no further stage configuration is required. The server will access this Connector module’s files from the location specified.

Identity

Enter a name to be used to identify this Connector module.

```
Name: StoredProcedureConnector

The name of this Connector module deployment.
```
Deploying a Connector (Continued)

This page allows you to view the deployment status of each Connector module, and to stop or redeploy individual Connector modules. (To configure additional deployment targets for these Connector modules, click the Targets tab.)

<table>
<thead>
<tr>
<th>Target</th>
<th>Target Type</th>
<th>Deployment Status</th>
<th>Status of Last Action</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>mserver</td>
<td>Server</td>
<td>Available</td>
<td>Success</td>
<td>Stop</td>
</tr>
</tbody>
</table>
Viewing the JNDI Tree

This page allows you to view the deployment status of each Connector module, and to stop or redeploy individual Connector modules. (To configure additional deployment targets for these Connector modules, click the Targets tab.)

<table>
<thead>
<tr>
<th>Target</th>
<th>Target Type</th>
<th>Deployment Status</th>
<th>Status of Last Action</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>myserver</td>
<td>Server</td>
<td>Available</td>
<td>Success</td>
<td>Stop</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Redeploy</td>
</tr>
</tbody>
</table>
Viewing the JNDI Tree
The Deployment Descriptor

This connector module is a deployed archive. Dynamic update of archived connector module descriptor is not supported. Any descriptor changes will have to be done outside the console. After making such changes, please undeploy the module for the changes to take effect. To enable dynamic update of connector module descriptor, please undeploy and deploy the connector module in exploded form.

Deployment Descriptors

ra.xml
weblogic-ra.xml
The Deployment Descriptor

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC "-//Sun Microsystems, Inc.//DTD Connector 1.0//EN" 'http://java.sun.com/dtd/connector_1_0.dtd'>
<connector>
  <display-name>SQL Stored Procedure Connector</display-name>
  <vendor-name>Tivoli</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>StoredProcedureConnector</eis-type>
  <version>1.0</version>
  <license>
    <description>GPL</description>
    <license-required>false</license-required>
  </license>
  <resourceadapter>
    <managed-connection-factory-class>com.trilogica.truaccess.connector.sql.StoredProcedureManagedConnectionFactory</managed-connection-factory-class>
  </resourceadapter>
</connector>
```

Valerie Pressley — Packaging and Deploying a J2CA Connector