ORACLE®

Oracle WebLogic Server

JDBC Resource Ref Deployment Plan Mapping

Java EE Resource References

- Java EE provides an abstraction mechanism to externalize resource dependencies
  - JDBC, JMS, etc.
- Do the Logical:
  - Developer declares a resource-ref entry in application descriptor file
  - Uses resource-ref entry in code
- Do the Physical
  - Using vendor deployment descriptor, map the resource-ref to a JNDI-NAME of resource on server

Java EE Resource References

- web.xml

```
<resource-ref>
  <description>A logical reference to a DataSource
  <res-ref-name>jdbc/LogicalDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- weblogic.xml

```
<weblogic-web-app>
  <resource-description>
    <res-ref-name>jdbc/LogicalDS</res-ref-name>
    <jndi-name>MAP_THIS_DS</jndi-name>
  </resource-description>
</weblogic-web-app>
```

Java EE Resource References

- Web application
  - Looks up logical DataSource from Java EE ENC
  - Or is injected as a named resource
  - No direct reference to physical database location

```
DataSource logicalds = null;
String DS_NAME = "java:comp/env/jdbc/LogicalDS";
Connection c = null;
try  {
  Context ic = new InitialContext();
  logicalds = (DataSource)ic.lookup(DS_NAME);
  ...
} catch(Exception e) {
  ...
}
```

Hey … That’s Still Tightly Coupled

- If weblogic.xml contains JNDI-NAME reference then EAR file needs to be edited to change the value
  - Defeats the purpose a little …
- weblogic.xml

```
<weblogic-web-app>
  <resource-description>
    <res-ref-name>jdbc/LogicalDS</res-ref-name>
    <jndi-name>DS_NEEDS_MAPPING</jndi-name>
  </resource-description>
</weblogic-web-app>
```

ORACLE

The Deployment Plan

- Deployment plan fully externalizes the settings for an application
  - Override/change settings for the application
- Separate to the application
  - 1→ 1, 1→ M relationship to an application
  - Domain specific or agnostic
- Fed into the deployment process
  - Command line option, specified in console
  - Changed deployment plan is read on update operation

ORACLE

WLS Deployment Plan Tooling

- Tooling provided to “generate” a default deployment plan

```
Usage: java weblogic.PlanGenerator [options] [Path to application]


where options include:
  -plan <myplan.xml> Name of plan to create. If not specified a default
                     will be used.
  -dependencies      (default) create plan that exports all dependency
                     properties
  -declarations      create plan that exports all declaration properties
  -configurables     create plan that exports all configurable properties
                     except for dependencies and declarations
  -all               create plan that exports all changeable properties
```

ORACLE

WLS Deployment Plan Tooling

- Example:

```
$java weblogic.PlanGenerator -dependencies
-plan raw-plan.xml jdbc-resource-ref.ear

Generating plan for application jdbc-resource-ref.ear
Export option is: dependencies
Exporting properties...
Saving plan to raw-plan.xml...
<18/12/2008 04:46:25 PM CST> <Info> <J2EE Deployment SPI>
<BEA-260072> <Saved configuration for application, jdbc-
resource-ref.ear>
```

WLS Deployment Plan Tooling

- Resulting deployment plan:
  – Variable created for declared resource-ref entry

```
<deployment-plan>
  <application-name>jdbc-resource-ref.ear</application-name>
  <variable-definition>
    <variable>
      <name>
        ResourceDescription_jdbc/LogicalDS_JNDIName_12295809854680
      </name>
      <value xsi:nil="true"></value>
    </variable>
  </variable-definition>
```

WLS Deployment Plan Tooling

- Resulting deployment plan:
  - Mapping generated for resource-ref element

```
 <module-override>
   <module-name>jdbc-resource-ref-web.war</module-name>
   <module-descriptor>
     <root-element>weblogic-web-app</root-element>
     <variable-assignment>
      <name>
        ResourceDescription_jdbc/LogicalDS_JNDIName_12295809854680
      </name>
      <xpath>
/weblogic-web-app/resource-description/[res-ref-name="jdbc/LogicalDS"]/jndi-
name
       </xpath>
     </variable-assignment>
   </module-descriptor>
  </module-override>
```

ORACLE

Using a Deployment Plan

- Modify variable values to reflect physical resources of target server
  - Update value with JNDI name
- Modify module-descriptor to specify required behavior with variable value
  - add, remove, replace

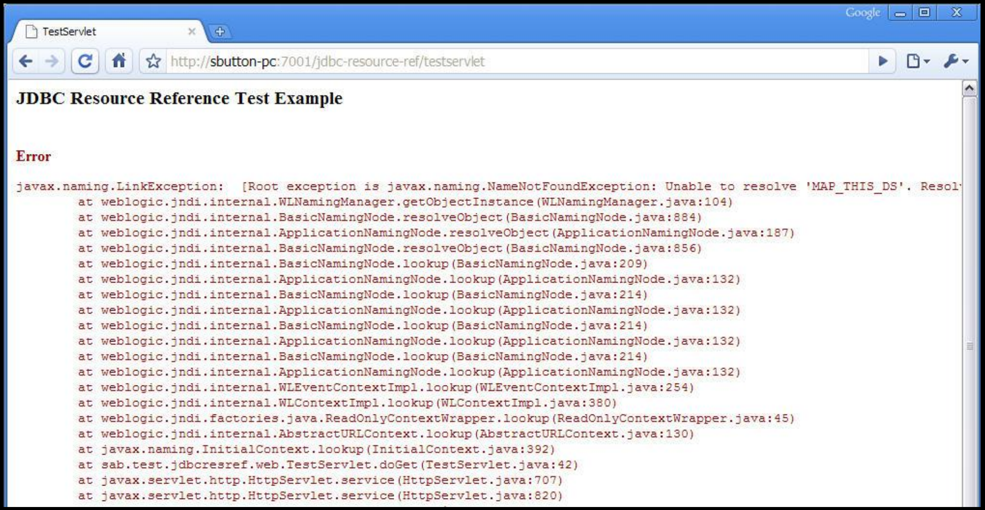
Example Deployment Plan

```
<deployment-plan>
  <application-name>jdbc-resource-ref.ear</application-name>
  <variable-definition>
    <variable>
      <name>ResourceDescription_jdbc/LogicalDS_JNDIName_12295809854680</name>
      <value>jdbc/XEDS</value>
    </variable>
  </variable-definition>
  <module-override>
   <module-name>jdbc-resource-ref-web.war</module-name>
   <module-descriptor>
     <root-element>weblogic-web-app</root-element>
     <variable-assignment>
      <name>ResourceDescription_jdbc/LogicalDS_JNDIName_12295809854680</name>
      <xpath>/weblogic-web-app/resource-description/[res-ref-
name="jdbc/LogicalDS"]/jndi-name</xpath>
      <operation>replace</operation>
     </variable-assignment>
   </module-descriptor>
  </module-override>
```

Deployment Process

- Deployment **without** deployment plan
  - Setting in weblogic.xml is used

```
$java weblogic.Deployer -username weblogic -password weblogic -
url t3://localhost:7001 -deploy -name jdbc-resource-ref jdbc-
resource-ref.ear
```



JDBC Resource Reference Test Example

Error

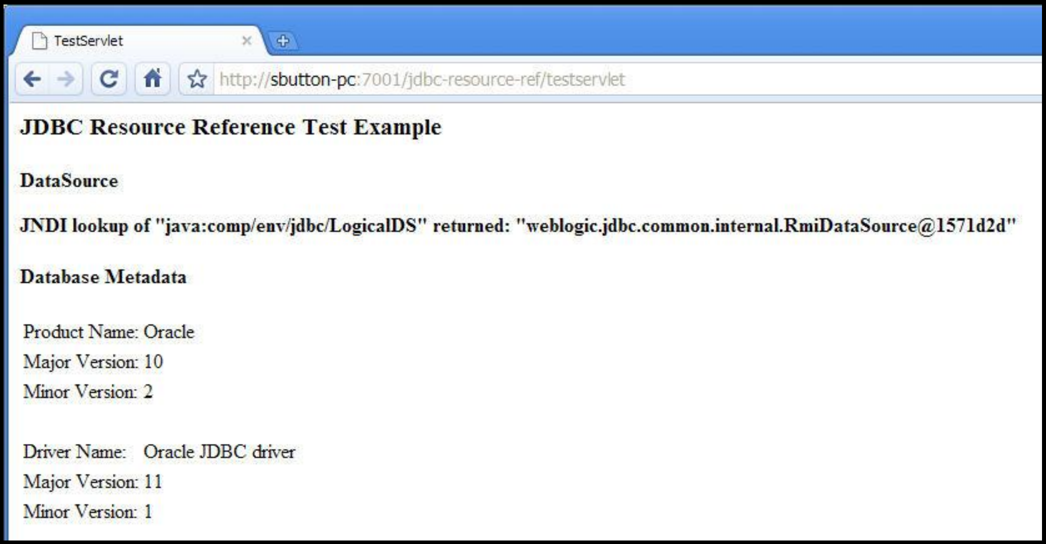
```
javax.naming.LinkException:  [Root exception is javax.naming.NameNotFoundException: Unable to resolve 'MAP_THIS_DS'. Resol
	at weblogic.jndi.internal.WLNamingManager.getObjectInstance(WLNamingManager.java:104)
	at weblogic.jndi.internal.BasicNamingNode.resolveObject(BasicNamingNode.java:884)
	at weblogic.jndi.internal.ApplicationNamingNode.resolveObject(ApplicationNamingNode.java:187)
	at weblogic.jndi.internal.BasicNamingNode.resolveObject(BasicNamingNode.java:856)
	at weblogic.jndi.internal.BasicNamingNode.lookup(BasicNamingNode.java:209)
	at weblogic.jndi.internal.ApplicationNamingNode.lookup(ApplicationNamingNode.java:132)
	at weblogic.jndi.internal.BasicNamingNode.lookup(BasicNamingNode.java:214)
	at weblogic.jndi.internal.ApplicationNamingNode.lookup(ApplicationNamingNode.java:132)
	at weblogic.jndi.internal.BasicNamingNode.lookup(BasicNamingNode.java:214)
	at weblogic.jndi.internal.ApplicationNamingNode.lookup(ApplicationNamingNode.java:132)
	at weblogic.jndi.internal.BasicNamingNode.lookup(BasicNamingNode.java:214)
	at weblogic.jndi.internal.ApplicationNamingNode.lookup(ApplicationNamingNode.java:132)
	at weblogic.jndi.internal.WLEventContextImpl.lookup(WLEventContextImpl.java:254)
	at weblogic.jndi.internal.WLContextImpl.lookup(WLContextImpl.java:380)
	at weblogic.jndi.factories.java.ReadOnlyContextWrapper.lookup(ReadOnlyContextWrapper.java:45)
	at weblogic.jndi.internal.AbstractURLContext.lookup(AbstractURLContext.java:130)
	at javax.naming.InitialContext.lookup(InitialContext.java:392)
	at sab.test.jdbcresref.web.TestServlet.doGet(TestServlet.java:42)
	at javax.servlet.http.HttpServlet.service(HttpServlet.java:707)
	at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
```

ORACLE

Deployment Process

- Deployment **with** deployment plan
  - Setting in deployment plan is used

```
$java weblogic.Deployer -username weblogic -password weblogic -
url t3://localhost:7001 -deploy -name jdbc-resource-ref -plan
raw-plan.xml jdbc-resource-ref.ear
```



JDBC Resource Reference Test Example

DataSource

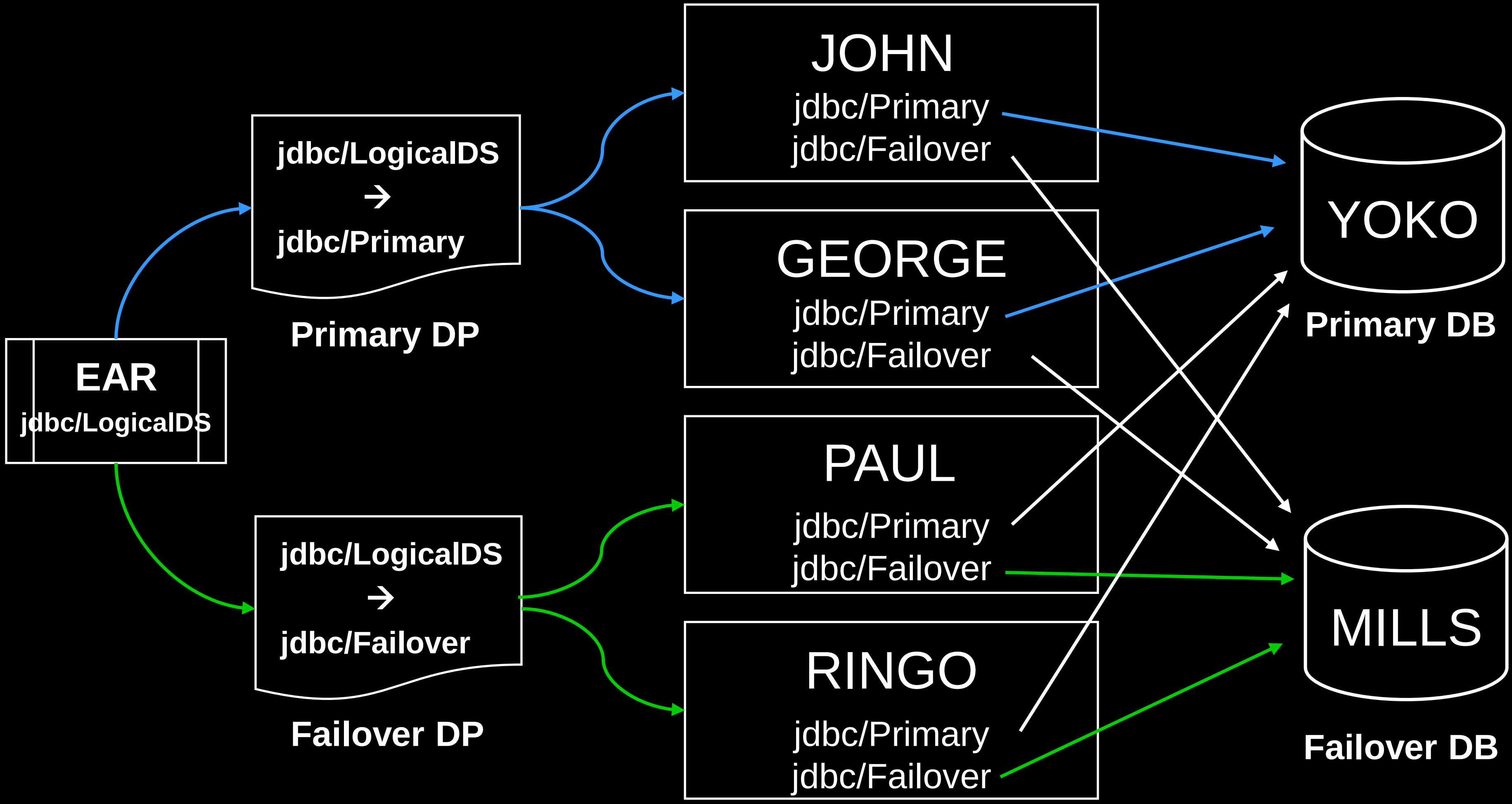
JNDI lookup of "java:comp/env/jdbc/LogicalDS" returned: "weblogic.jdbc.common.internal.RmiDataSource@1571d2d"

Database Metadata

```
Product Name: Oracle
Major Version: 10
Minor Version: 2

Driver Name:   Oracle JDBC driver
Major Version: 11
Minor Version: 1
```

Same App, Multi WLS Instances



Summary

- Allow externalization of resource declarations
- Use variable substitution, append operations
- Supplied during deployment process for all deployment tools/utilities (ant, weblogic.Deployer, console)
- Powerful capability!

ORACLE®