

Oracle Application Server 10g Upgrade and Migration

ORACLE



Monika Dreher

Product Technology Services
ORACLE Corporation

ORACLE

Agenda

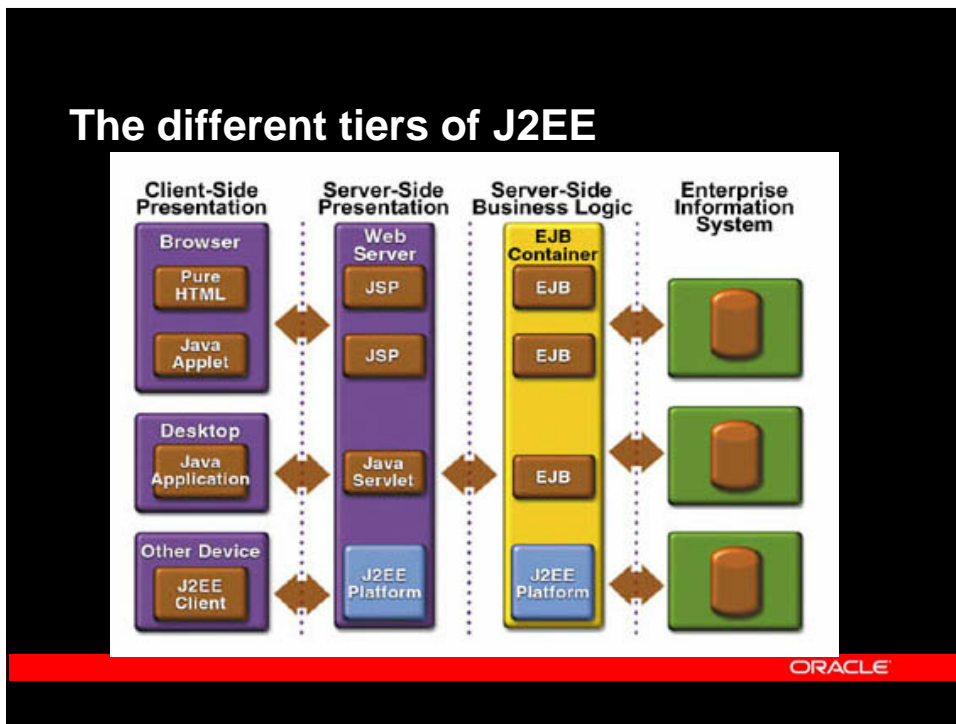
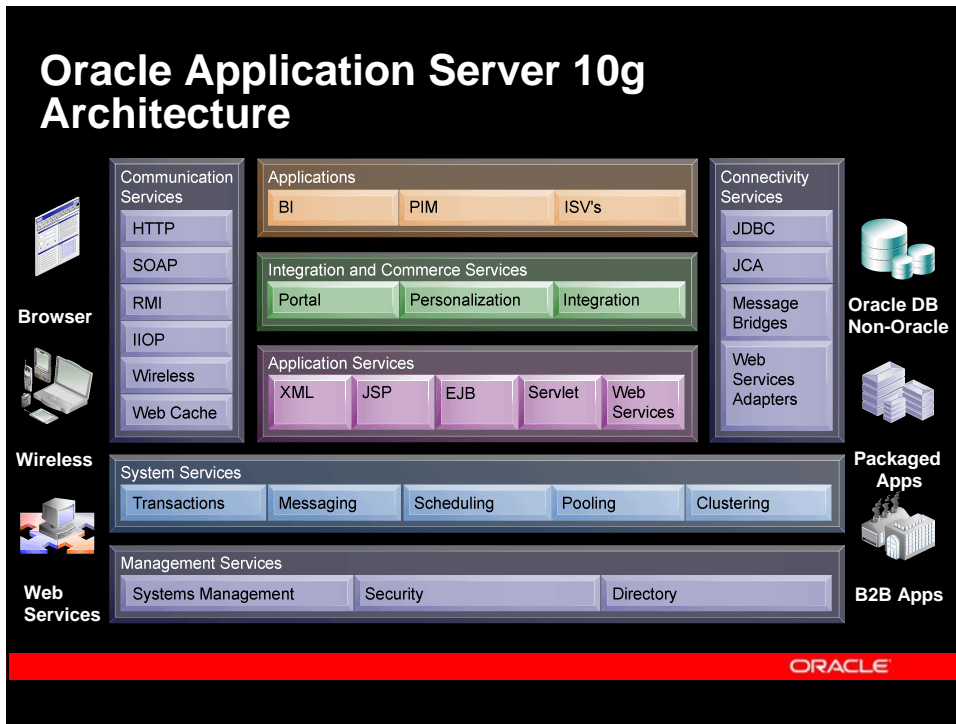
- Introduction to the Oracle Application Server 10g
- Quick start to Oracle OC4J
- The Migration steps
- Migration in more detail

ORACLE

Introduction to the Oracle Application Server 10g



ORACLE



Four different levels of product support

- OC4J Standalone
- Java Edition
- Standard Edition
- Enterprise Edition

ORACLE

Quick start to Oracle OC4J

The J2EE engine

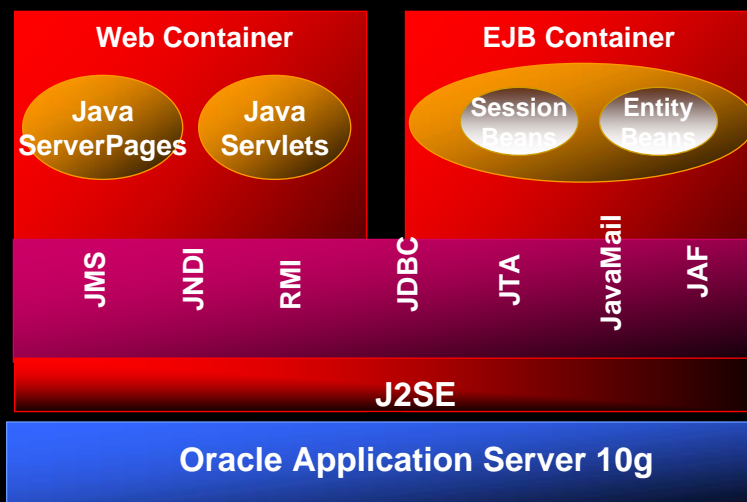
ORACLE

OC4J Overview

- Complete J2EE implementation
- Lightweight & High Performance
- Small memory footprint
- XML file based configuration for all aspects of server, web server and applications
- Simplified installation, configuration, deployment and administration
- Auto deployment and “Hot” deployment of J2EE Applications
- Clustering, Load balancing and Fail-over of Web & EJB Apps

ORACLE

Complete J2EE 1.3 implementation



ORACLE

Installation Pre-Requisites

- Identifying the JDK version

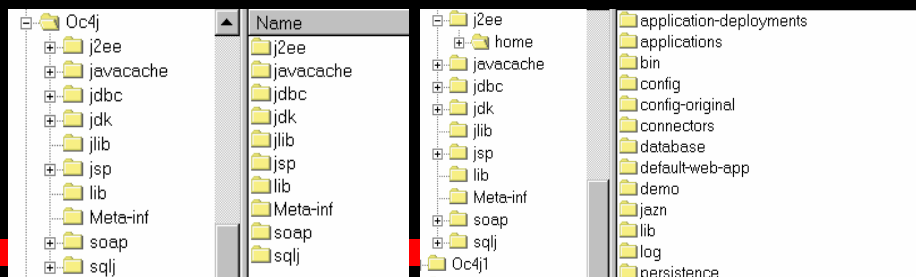
```
%>java -version  
java version "1.4.2"  
Java(TM) 2 Runtime Environment, Standard  
Edition (build 1.4.2-b28)  
Java HotSpot(TM) Client VM (build 1.4.2-b28,  
mixed mode)
```

ORACLE

Installing developers edition OC4J Server

Basic Installation

- `% cd $oc4j_home`
- `% unzip oc4j.zip`
- `% cd j2ee/home` (referenced as <j2ee-home>)
- `% java -jar oc4j.jar -install`
"Oracle9iAS (9.0.4.0.0) Containers for J2EE initialized"
- Directory Structure



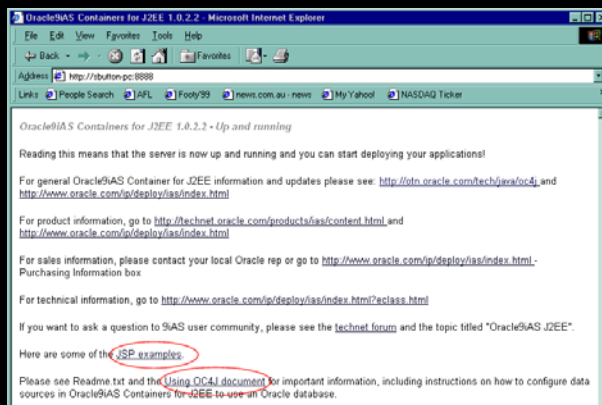
Starting up OC4J Server

- Startup Commands
- % java -jar oc4j.jar <options>
- option list:
 - -config <file>
 - -validateXML
 - -out [file]
 - -err
 - -install
 - -userThreads
 - -quiet
 - -version
 - -help

ORACLE

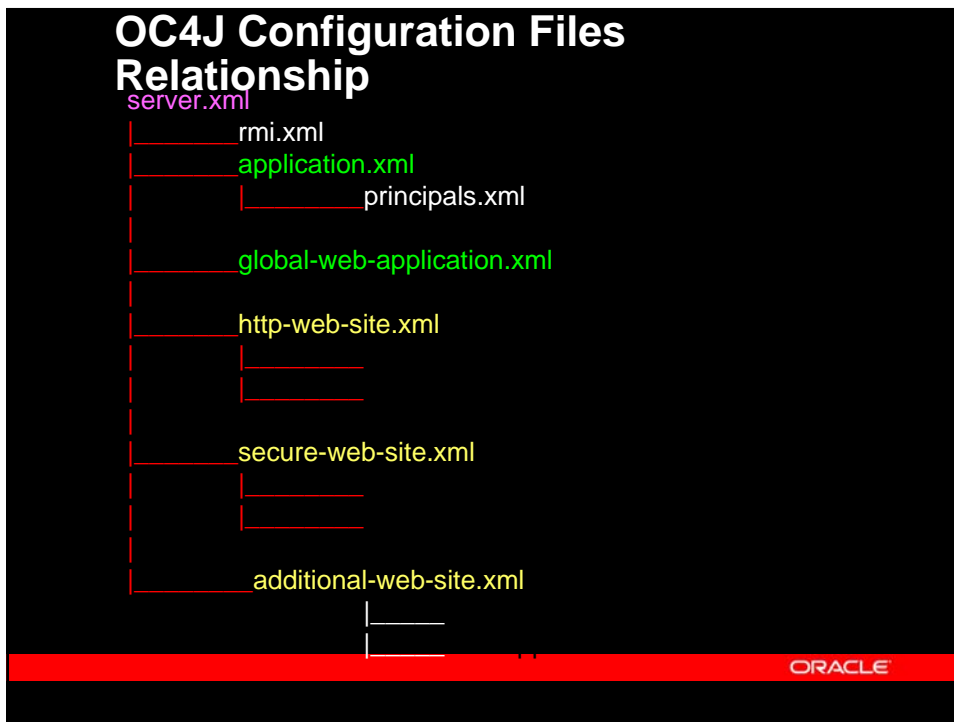
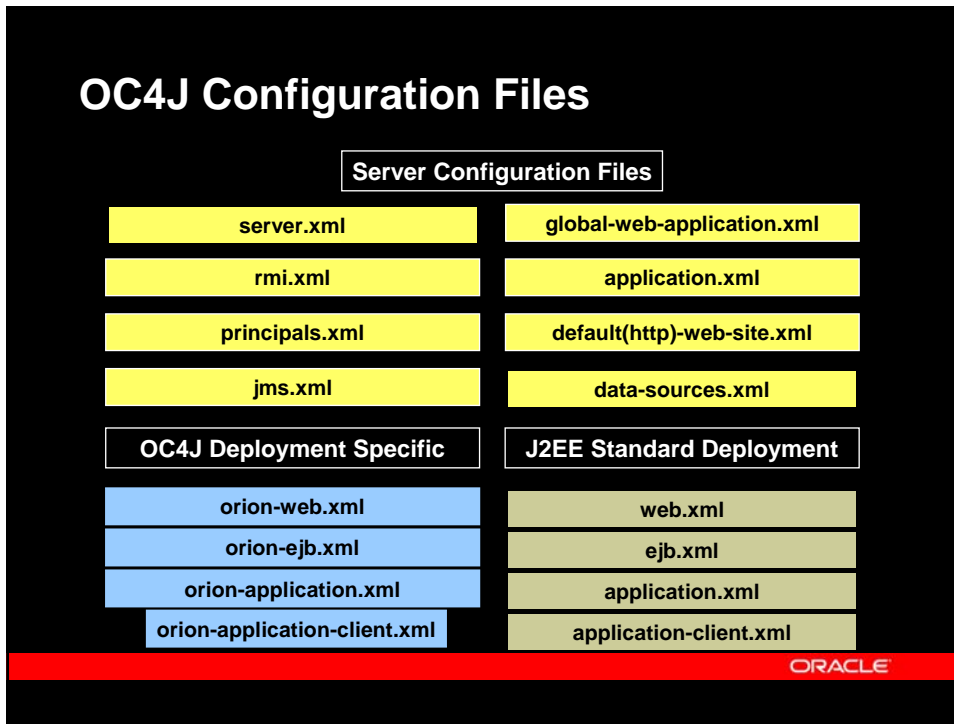
Verifying the Installation

- Access default Web site
http://<machine_name>:8888



- Using a different port
change port # in
.../http-web-site.xml

ORACLE



Example of Server.xml

```

<application-server>
  application-auto-deploy-directory=<auto-deploy-ears>
  application-directory="..applications"
  deployment-directory="..application-deployments"
  <rmi-config path="..rmi.xml" />
  <jms-config path="..jms.xml" />
  <log> <file path="..log/server.log" /> </log>

  <global-application name="default" path="application.xml" />
  <global-web-app-config path="global-web-application.xml" />
  <web-site path="..http-web-site.xml" />
  <web-site path="..oc4j-web-site.xml" />

  <application auto-start="true" deployment-directory="dir" parent="parentApps">
  <application name="news" path="..applications/news.ear" />

  <application name="petstore" path="..applications/petstore.ear" />
  <application name="oc4jApp" path="D:\oc4jApp" />
</application-server>

```

ORACLE

Example Web-Site.xml

```

<web-site host="hostname" port="n" cluster-island="1" display-
  name="WebSite" secure="false" use-keep-alives="true"
  virtual-hosts="hostname,...,hostname" />

<default-web-app application="defaultAppEAR" name="defaultAppWAR"
  shared="false" load-on-startup="false" />

<web-app application="PetStore" name="PetWeb" root="/petstore"
  shared="false" load-on-startup="false" />

<web-app application="oc4jApp" name="oc4jWeb" root="/oc4jRoot"
  shared="false" load-on-startup="false" />

<user-web-apps max-inactivity-time="n" path="path" />

<access-log format="format spec" path="path" suffix="suffix"
  split="none|hour|day|week|month" />

```

ORACLE

Deploying a stand alone EJB JAR File

- Editing XML Files directly

1. Edit application.xml

```
<ejb-module id="myEJB"  
  path="../../applications/myEJB.jar" />
```

2. Use command line tool

```
Java -jar admin.jar -deploy -file myEJB.jar  
  -deploymentName myEJB
```

ORACLE

Deploying a complete EAR file

- Editing XML files directly

- Edit server.xml and add

```
<application name="myShoppingApp"  
  path="myApp.ear" />
```

- Edit default-web-site.xml and add

```
<web-app application="myShoppingApp"  
  name="webFront" root="/webFront" />
```

- Access Web application with URL

<http://localhost:8888/webFront>

ORACLE

Deploy a complete EAR file

- Using admin command line

```
$ java -jar admin.jar  
-file myapp.ear  
-deploymentName myShoppingApp
```

```
$ java -jar admin.jar  
-bindWebApp myShoppingApp webFront  
/webfront
```

ORACLE

orion-ejb-jar.xml

- Contains OC4J specific application settings for EJBS
 - Various EJB flags such as copyByValue, exclusive-write-access, isolation-level etc
 - CMP Entity Bean Persistence to Database mapping
 - Tables
 - Fields
 - Relationships
 - etc

ORACLE

orion-web.xml

- Contains OC4J specific web settings
- Must be placed in the same directory as the standard web.xml
- Example of settings
 - Clusterable or not
 - development mode (checks for jsp page changes)
 - directory-browsing true/false
 - Local classpath information
 - EJB/Resource reference mappings..
 - etc.

ORACLE

jazn-data.xml

- Contains security definitions for JAAS/JAZN security provider
 - Roles
 - Users
 - Realms

ORACLE

The Migration steps

ORACLE

J2EE Migration

- Can be easy or difficult
 - Standard J2EE App → easy
 - Proprietary → difficult
- Is a moving target, different versions of the app server from a vendor will have different issues
- Web Tier tends to be the easiest to port
- EJB tiers can cause issues
 - Session beans port easily
 - Entity beans usually require assistance
 - Transaction boundaries may differ
- Platform specific code may need to be rewritten
 - E.g. BEA jolt

ORACLE

Basic Steps in any migration

- Identify differences between the application servers
- Remove platform specific proprietary features if possible
 - E.g. BEA JOLT, IBM Tags etc
- Port platform specific deployment descriptors
 - E.g. weblogic-ejb-jar.xml -> orion-ejb-jar.xml
- Port application in tiers starting from data tier (e.g.EJB, webservice) up to client

ORACLE

Step 1 : Convert XML Deployment Descriptor

- Standard J2EE descriptors need little changes and should port easily
- Proprietary J2EE descriptors will need rewriting
 - E.g. weblogic-ejb-jar.xml equivalent in OC4J is orion-ejb-jar.xml
- Definition of whats available in OC4J xml files can be obtained online via DTDs or online manuals
 - <http://xmlns.oracle.com/ias/dtds/orion-ejb->

ORACLE

Methods of rewriting XML deployment files

- Manual creation of platform specific deployment descriptor files, using vi,emacs, notepad etc
- XSLT
 - We provide sample BEA, JBoss & Borland
- Reverse Engineer project into JDeveloper
 - Let JDeveloper generate deployment descriptors
- Use JDeveloper's deployment descriptor wizards to assist in the authoring of the deployment descriptors

ORACLE

Example orion-web-ejb.xml JDeveloper Wizard

The screenshot shows the 'OC4J EJB Deployment Descriptor' wizard in JDeveloper. The left pane displays a tree view of the deployment descriptor structure, with 'shoppingCartEJB' selected under 'Enterprise JavaBeans'. The right pane contains configuration fields for the selected EJB, including 'Session EJB Name' (shoppingCartEJB), 'JNDI Location', 'Persistence Filename', 'Call Timeout', 'Timeout', and 'Maximum Number of Retries'. A 'Copy by Value' checkbox is checked. The bottom of the window features 'New...', 'Delete', 'Help', 'OK', and 'Cancel' buttons.

ORACLE

Step 2 : Migrate EJB Components

- Always migrate data layer first
- Usually the most difficult layer to port
- Unit Test each EJB as you go along
 - JDeveloper is able to create test client stubs which may assist in testing EJBS individually
- EJBQL & Finder queries may be different

ORACLE

Step 3 : Migrate Web/Client

- Migrate web/client tier
- Usually easier than EJB tier
- Beware that Oracle OC4J doesn't support client side transactions
- JNDI names can be different
 - Use JDeveloper to browse JNDI tree in OC4J

ORACLE

Step 4 : Migrate other components

- Datasources
- JMS Message Queues
- JCA Adaptors

ORACLE

Step 5 : Implementing Clustering

- Options
 - OC4J Standalone does not support clustering
 - Java Edition supports file based clustering
 - Oracle Application Server 10g supports managed Clustering
- When deploying an application to a managed cluster you must have the `<distributable>` tag in web.xml
 - All session objects are propagated to other nodes in island so they must be serializable

ORACLE

Migration in more detail

ORACLE

Migrating Servlets

ORACLE

Migrating Servlets

- Migrating servlets to AS is usually a smooth & effortless exercise
- Typically no modifications are necessary
 - However if proprietary server extensions have been used then they will almost certainly need to be rewritten.

ORACLE Cont.....

IBM Servlet Migration Issues



Websphere has non-standard extensions

- Websphere has two mode of deployment
 - Compatibility Mode: Servlet API 2.0 and 2.1
 - Compliance Mode : Servlet 2.2 (partial)
- Websphere Studio generated code not portable
- Websphere servlet extensions packages
- Web Application Deployment
 - No direct support for war files
 - custom .servlet file for configuring servlet

ORACLE Cont.....

IBM WebSphere Extensions to Servlets



- Servlet Session Management Package
 - `com.ibm.websphere.servlet.session`
(replace with J2EE `javax.servlet.http.HttpSession`)
- Personalization Package
 - `com.ibm.servlet.personalization` (replace with 9iAS Personalization)
- Event Manager package
 - `com.ibm.websphere.servlet.event` (replace with Servlet 2.3 API: `ServletContextEvent`, `HttpSessionEvent`, `HttpSessionBindingEvent`)
- Custom Servlet Filtering package
 - `com.ibm.websphere.servlet.filter` (replace with Servlet 2.3 API: `javax.servlet.Filter`, `FilterConfig`)
- Servlet Proxy Package
 - `com.ibm.websphere.servlet.request`(replace with `HttpServletRequest` Request, use servlet chaining features in Servlet 2.3)
 - `com.ibm.websphere.servlet.response`(replace with `HttpServletResponse` Response, use servlet chaining features in Servlet 2.3)

ORACLE Cont....

Migrating JSPs

ORACLE

Migrating JSPs

- Migrating JSPs to AS typically is a smooth & effortless exercise
- Proprietary tag libraries will need to be ported to either
 - Oracle proprietary tag libraries
 - Custom tag libraryTab
- Tag Libraries can be migrated with ease
- Proprietary extensions may need to be ported
 - E.g. BEA Weblogic htmlKona

ORACLE



Migrating JSP Custom Tags

- Custom tag libraries can be easily deployed on AS as part of a Web application
- Customer or 3rd Party Tags can be deployed on AS typically with very little changes
- Proprietary custom tags to be replaced with equivalent AS Custom Tags
 - E.g. IBM Websphere data tags can partially be replaced by OJSP tags

ORACLE

Using Tag Libraries in OC4J

- 3 Steps to using tag libraries in OC4J
 - 1. Copy the <tag-library>.tld into /WEB-INF directory and the necessary classes into WEB-INF/lib
 - 2. Edit the web.xml in the /WEB-INF directory

```
<taglib>
<taglib-uri>mytags</taglib-uri>
<taglib-location>
/WEB-INF/sqltaglib.tld
</taglib-location>
</taglib>
```

- 3. Use the tags in your JSP

ORACLE



IBM specific JSP Migration issues

- Bean scripting framework(BSF) (Websphere specific)
- Websphere specific Tags and properties
 - “<SERVLET>” tag (replace with “<jsp:include>” tag)
 - “<BEAN>” (replace with <jsp:useBean> tag)
 - “<CONNECT>”(create your custom tag library or use OJSP)
 - “<tsx:repeat>” (create your custom tag library)
 - “<tsx:dbmodify>” (create your custom tag library)

ORACLE

Migrating EJBs

ORACLE

Migrating EJBs

- Migration of EJBs to OC4J is typically straightforward
 - Session Beans migrate easier than Entity beans
- Typically little or no code modifications are required
- Code modifications often relate to minor JNDI lookup changes
- Implementation-specific adaptations for O/R mapping, container class generation, & customization of deployment properties are required

ORACLE

Migrating EJBs

- Standard J2EE components and deployment descriptors require almost no modifications
- Implementation-specific dependencies require modification
 - Hard-coded JNDI context access and lookups
 - Data source JNDI names and lookups
- Correct and re-generate the code and EJB archive file as required

ORACLE

Migrating Session Beans

- Migration of session beans involves only the generic steps of EJB migration
- Session EJBs are much easier to migrate than Entity EJBs since there are no persistence-related migration issues

ORACLE

Migrating Entity Beans

- Migration of Entity beans involves the generic steps of EJB migration
- Plus
 - removal of implementation-specific JNDI or DataSource lookups
 - Rewriting of specific database deployment descriptors
 - E.g. weblogic-ejb-jar.xml to orion-ejb-jar.xml
 - Removal of Proprietary APIs or Flags
 - Transaction Mgmt, Locking, Caching,...

ORACLE

Migrating CMP Entity Beans

- O/R mapping definitions are vendor dependent, require re-generation
 - E.g. weblogic-ejb-jar.xml, weblogic-cmp-rdbms-jar*.xml
- Use xslt transformers to give you a head start
- Check XML DTD & Documentation for options available
- DataSource for persistence to be configured for OC4J
- BEA WebLogic-specific container stub & skeleton classes need to be removed
 - OC4J will generate the appropriate equivalent stub & skeleton classes

ORACLE

Transactions

- OC4J (9.0.4) does not support client side transactions
 - Need to delegate transaction to middle tier, perhaps a façade
- BEA & IBM support distributed transactions across multiple JVMs
 - Presently OC4J does not support this
 - Planned for a release post 10g

ORACLE

Classloader issues

- Can occur at any tier of the application
- Common clashes are different versions of XML parsers
- Use OC4J Parent classloader architecture to overcome this
 - E.g. adding
`<web-app-class-loader search-local-classes-first="true" />`
to
`orion-web-app.xml`
- White papers exist which describe how OC4Js class loader work
 - See <http://otn.oracle.com> OC4J whitepapers

ORACLE

JNDI Issues

- In OC4J JNDI is rooted at the application level not at the Server level
 - Can be solved in some cases by the use of the <parent> directive in server.xml
- JNDI Clustering is not supported in OC4J 9.0.3, but is supported in OC4J 9.0.4

ORACLE

JDBC Datasources Migration

ORACLE

JDBC Migration issues

- In WebLogic the data source entries are made in the weblogic.properties file
- In OC4J the datasource entries are made in the data-sources.xml file or using Oracle Enterprise Manager
- dbKona code must be replaced with equivalent JDBC 2.0 code
- One should use <ejb-location> tag to get pooled and automatic container spawned transactions

ORACLE



JDBC Migration issues

- IBM Connection Pool manager
 - com.ibm.db2.jdbc.app.stdext.javax.sql.*;
 - com.ibm.ejs.dbm.jdbcext.*;
 - com.ibm.ejs.cm.*
 - com.ibm.ejs.cm.pool.*
 - (replace the above packages with J2EE javax.sql and java.sql packages)
- data access beans
 - com.ibm.db package
- Obtaining datasource reference
 - Need to change Websphere JNDI naming context with OC4J naming context

ORACLE

Example BEA Proprietary Extensions



BEA Specific

COM Support

Jolt for WebLogic

WebLogic Events

OC4J/J2EE Compatible

None - available from partners & 3rd party

Plain BEA Jolt for Tuxedo

JMS

ORACLE

Proprietary Hints and Tips

- Migration hints and tips document
- Separate MS-Word documents with hints and tips for popular J2EE application servers

ORACLE

Clustering and Deployment

ORACLE

Deployment is easy

- Standalone OC4J
 - Edit XML Files directly
 - Use java -jar admin.jar utility
- Using Enterprise Manager
 - Select OC4J Instance
 - Then select Deploy EAR, or WAR

ORACLE

Clustering Web Applications

- For clustered web solutions ensure
 - web.xml has <distributable> tag present
 - All session objects are serializable

ORACLE

Clustering EJB applications

- OC4J can cluster
 - stateful and Stateless session beans
- Three types of replication
 - JVM Termination replication
 - End of Call Replication
 - Stateful session context replication
 - Set using `replication` attribute of the `<session-deployment>` tag in `orion-ejb-jar.xml` file
- EJB Replication only works with ORMI protocol not RMI over IIOP
- For more info see EJB Developers guide (9.0.4) chapter 10

ORACLE

JDeveloper 10g

- Since Oracle JDeveloper 9i we provide an IDE that facilitates
 - code changes (refactoring, source control etc)
 - fine-tuning and rapid-deployment of the applications to OC4J
 - Wizard based generation of generic and specific XML files

ORACLE

Q U E S T I O N S
&
A N S W E R S

ORACLE