Coherence 12.1.2 Managed Servers

Noah Arliss – Software Development Manager (Sheriff)

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remain at the sole discretion of Oracle.

Evolving to simplify the…

- Development Lifecycle
  - Grid Archives, Maven and additional tooling
- Deployment Model
  - Managed Servers and Grid Archives
- Operations Management
  - Centralized configuration and management of your runtime

Agenda

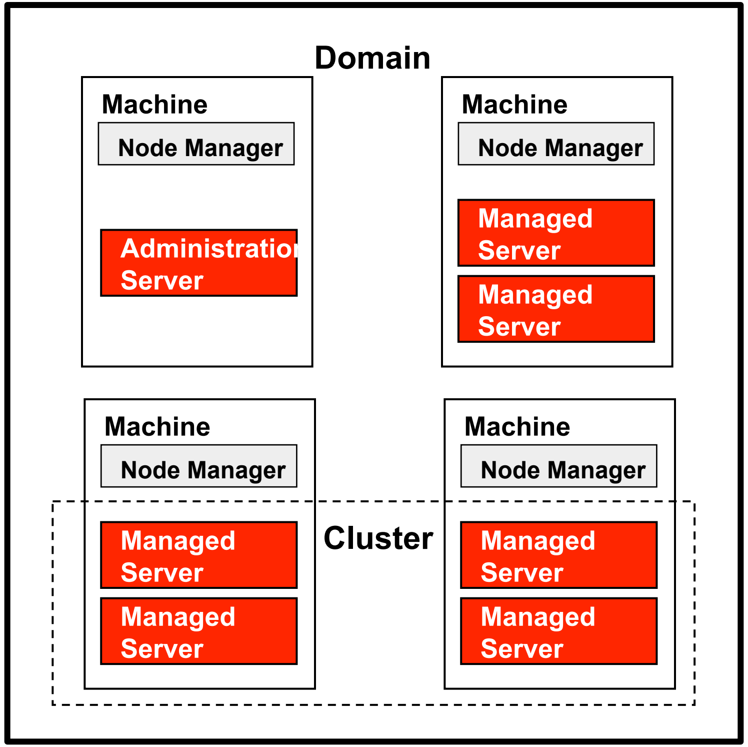
- Definition of terms
- Managed Coherence Servers
- The Grid Archive (GAR)
- Maven Support
- Deployment Models
- Weblogic, Maven and WLST
- Tying it all together

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

WebLogic Manageability

WebLogic Terminology

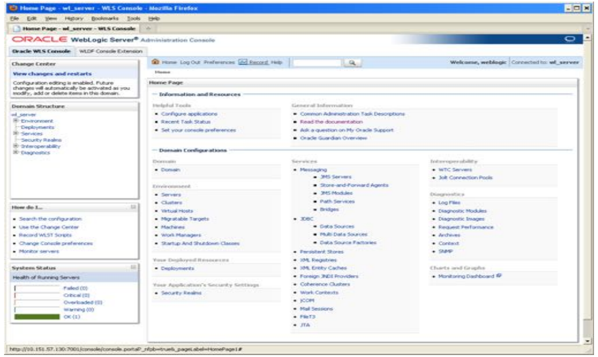
- Domain - group of instances and clusters under unified control
- Administration Server - central configuration controller for the domain
- Managed Server - Instance that hosts applications and required resources
- Cluster - group of managed servers to provide increased scalability and reliability
- Node Manager - per-machine process used to start, stop and auto-restart instances



WebLogic Manageability

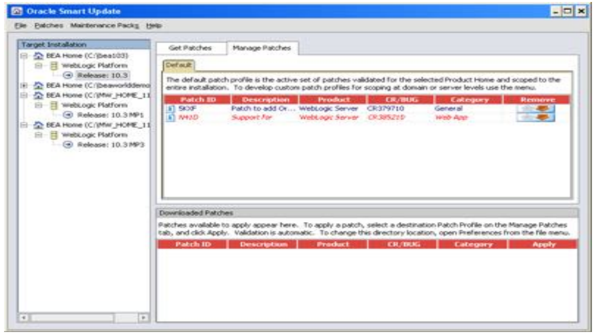
Configuration Management, Automation

- Configuration Wizard
  - Create configurations from templates
  - GUI and scripting support
  - Automate domain creation and replication
- WebLogic Scripting Tool
  - Automate management via Jython-based scripts
  - Config, deploy, monitor, lifecycle
  - “Record” console actions to script
  - WLST script editor in OEPE



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

http://10.151.57.130:7001/console/console.portal?_nfpb=true&_pageLabel=HomePage1#



The default patch profile is the active set of patches validated for the selected Product Home and scoped to the entire installation. To develop custom patch profiles for scoping at domain or server levels use the menu.

Patch ID	Description	Product	CR/BUG	Category	Remove
SKXF	Patch to add Or...	WebLogic Server	CR379710	General	
NH2D	Support for	WebLogic Server	CR385210	Web App	

Patches available to apply appear here. To apply a patch, select a destination Patch Profile on the Manage Patches tab, and click Apply. Validation is automatic. To change this directory location, open Preferences from the file menu.

Patch ID	Description	Product	CR/BUG	Category	Apply

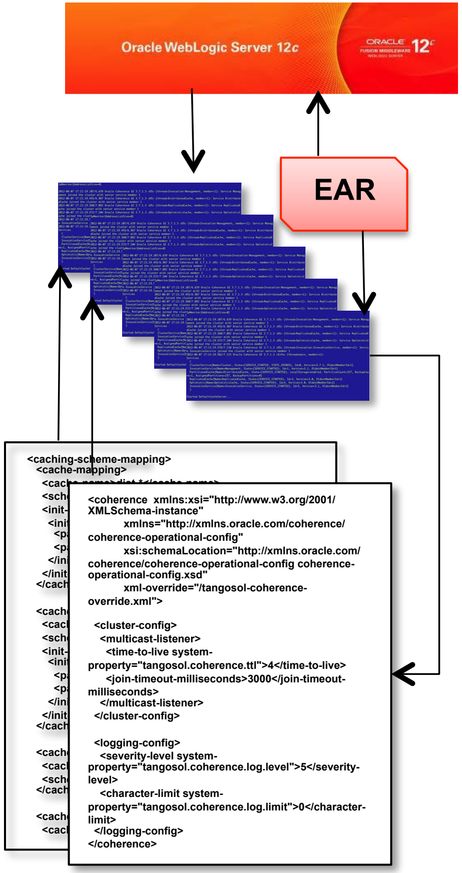
© 2010 Oracle Corporation

Agenda

- Definition of terms
- Managed Coherence Servers
- The Grid Archive (GAR)
- Maven Support
- Deployment Models
- Weblogic, Maven and WLST
- Tying it all together

Traditional Deployment Approach

- Traditional Deployment & Management Approach
  - Coherence applications have traditionally been deployed as a jar incorporated into a Java application such as a WAR/EAR or standalone Java application
  - Typically there are usually two roles within a Coherence application:
    - Servers – hold cache data and perform server processing
    - Clients – do not hold data but perform data operations
  - Servers are usually started by Running a Java class (DefaultCacheServer) using scripts specifying an array of JVM arguments to configure an instance
- Challenges with this approach
  - The lifecycle of these artifacts are usually managed separately by custom shell scripts/utilities and not by WebLogic Server (WLS)
  - This can make the development, deployment and management of these type of applications more difficult
  - No “standard” way to deploy/manage Coherence across different platforms and environments
  - Every customer has no option but to re-invent the wheel



```
<caching-scheme-mapping>
  <cache-mapping>
```

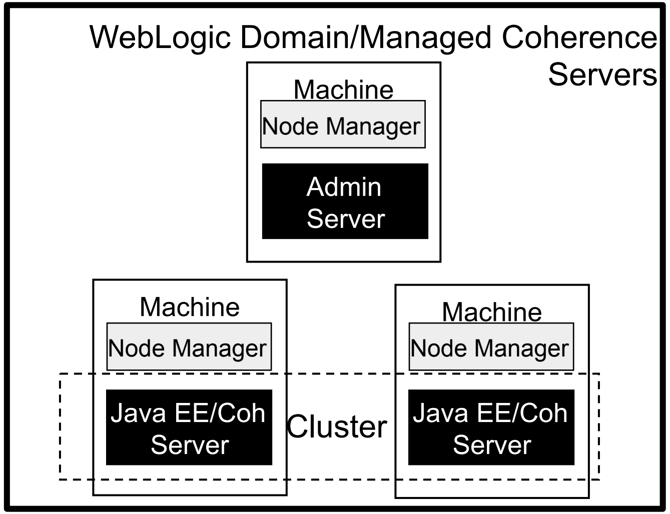
```
<coherence  xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
            xmlns="http://xmlns.oracle.com/coherence/
coherence-operational-config"
            xsi:schemaLocation="http://xmlns.oracle.com/
coherence/coherence-operational-config coherence-
operational-config.xsd"
            xml-override="/tangosol-coherence-
override.xml">

  <cluster-config>
    <multicast-listener>
      <time-to-live system-
property="tangosol.coherence.ttl">4</time-to-live>
      <join-timeout-milliseconds>3000</join-timeout-
milliseconds>
    </multicast-listener>
  </cluster-config>

  <logging-config>
    <severity-level system-
property="tangosol.coherence.log.level">5</severity-
level>
    <character-limit system-
property="tangosol.coherence.log.limit">0</character-
limit>
  </logging-config>
</coherence>
```

Managed Coherence Servers - Introduction

- Simplified and streamlined development and management environment
- Introduces the Grid Archive (GAR)
  - Follows the Java EE model
  - Consolidates configuration into a single distributable entity
  - Provides Application Isolation & Lifecycle Events
- Leverages WebLogic infrastructure
  - Similar to EJB, JMS and JCA
  - Management through Configuration Wizard, WebLogic admin console, WLST, JMX, FMWC
  - Deployment of Coherence applications (GARs)
  - Node Manager for lifecycle management
  - Release and install Coherence libraries on the system classpath (no need to package in your application)
- Coherence remains available “standalone” including support for GARs



WebLogic Domain/Managed Coherence Servers

Machine

Node Manager

Admin Server

Machine

Node Manager

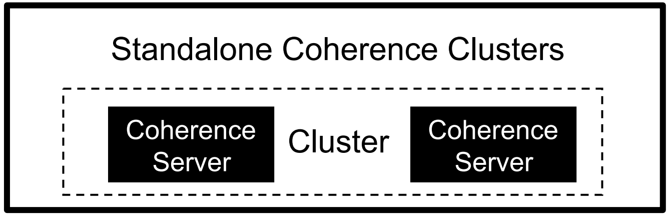
Java EE/Coh Server

Cluster

Machine

Node Manager

Java EE/Coh Server



Standalone Coherence Clusters

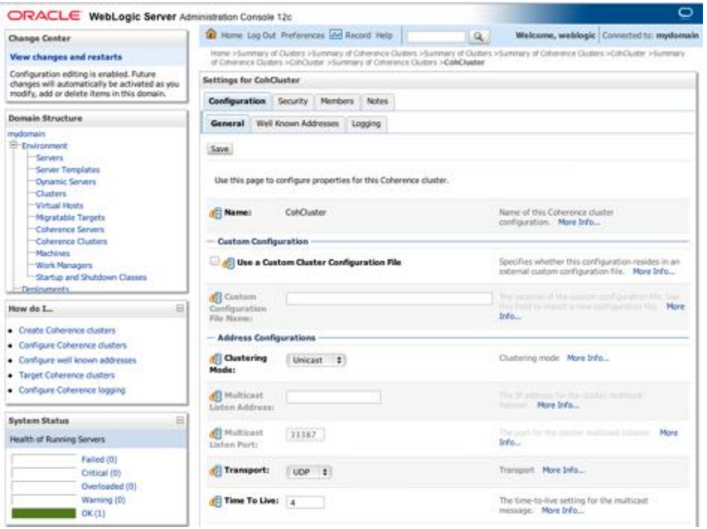
Coherence Server

Cluster

Coherence Server

Managed Coherence Servers - Benefits

- Simplified Operations Management
  - Configure, manage and deploy Coherence from the WebLogic Server administration console or via WLST
  - Manage Coherence resources centrally
  - Automatic WKA generation based on domain servers
  - Tight Maven integration
  - Ability to carry out rolling Restarts
  - Separates Operational Config from Cache Config
- Simplified development and deployment
  - GAR manages all application build artifacts
  - Simplified application deployment across entire data grid
  - Based on industry standard Java EE
  - Automatic application isolation
  - OEPE tooling available
- Easier integration for WebLogic Server-based applications that leverage Coherence
  - GAR's deployed in an EAR automatically make Coherence resources available to WAR's and EJB's in the EAR



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Home >Summary of Clusters >Summary of Coherence Clusters >Summary of Clusters >Summary of Coherence Clusters >CohCluster >Summary of Coherence Clusters >CohCluster >Summary of Coherence Clusters >CohCluster

Settings for CohCluster

Use this page to configure properties for this Coherence cluster.

Name: CohCluster — Name of this Coherence cluster configuration. More Info...

Custom Configuration

□ Use a Custom Cluster Configuration File — Specifies whether this configuration resides in an external custom configuration file. More Info...

Custom Configuration File Name: — The location of the custom configuration file. Use this field to import a new configuration file. More Info...

Address Configurations

Clustering Mode: Unicast — Clustering mode More Info...

Multicast Listen Address: — The IP address for the cluster multicast listener. More Info...

Multicast Listen Port: 33387 — The port for the cluster multicast listener. More Info...

Transport: UDP — Transport More Info...

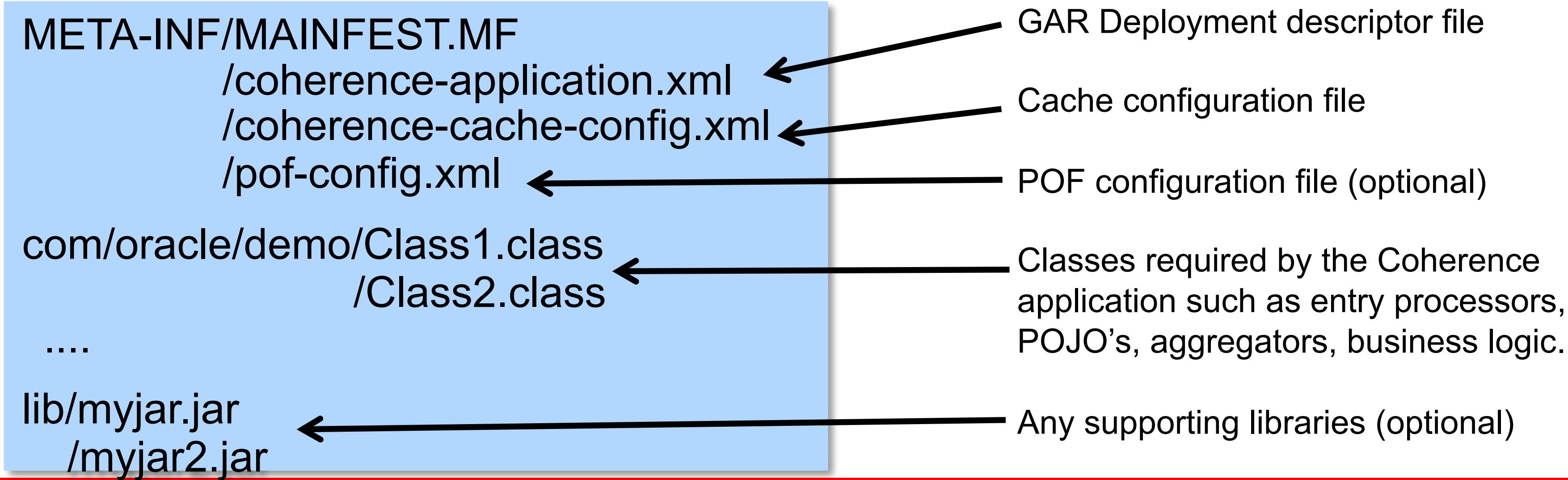
Time To Live: 4 — The time-to-live setting for the multicast message. More Info...

Agenda

- Definition of terms
- Managed Coherence Servers
- The Grid Archive (GAR)
- Maven Support
- Deployment Models
- Weblogic, Maven and WLST
- Tying it all together

What is a Grid Archive (GAR)?

- A GAR file all the dependent artifacts required for the operation of a Coherence application!
- The structure of a GAR is as follows:



```
META-INF/MAINFEST.MF
            /coherence-application.xml
            /coherence-cache-config.xml
            /pof-config.xml

com/oracle/demo/Class1.class
                    /Class2.class
  ....

lib/myjar.jar
   /myjar2.jar
```

GAR Deployment descriptor file

Cache configuration file

POF configuration file (optional)

Classes required by the Coherence application such as entry processors, POJO’s, aggregators, business logic.

Any supporting libraries (optional)

What is a Grid Archive (GAR)?

- The GAR deployment descriptor, coherence-application.xml contains the following:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<coherence-application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                       xmlns="http://xmlns.oracle.com/weblogic/coherence-application">

  <cache-configuration-ref
          override-property="cache-config/ExamplesGar">META-INF/example-cache-config.xml
  </cache-configuration-ref>

  <pof-configuration-ref>META-INF/pof-config.xml</pof-configuration-ref>

  <application-lifecycle-listener>
      <class-name>com.tangosol.examples.container.gar.LifecycleReactor</class-name>
  </application-lifecycle-listener>

  <configurable-cache-factory-config />
</coherence-application>
```

Pointer to cache configuration file, with optional override property

Pointer to POF configuration file

Optional implementation of LifeCycleListener for prę/post start processing

Optional implementation of ConfigurableCache Factory

Agenda

- Definition of terms
- Managed Coherence Servers
- The Grid Archive (GAR)
- Maven Support
- Deployment Models
- Weblogic, Maven and WLST
- Tying it all together

Maven

- There are three levels of maven support in 12.1.2:
  - The maven sync plugin – integrating coherence (and the rest of FMW) into your own maven repositories
    - Simply install the maven sync plugin to your repository
    - Then run the sync plugin to install FMW 12c elements into your desired repository
  - The GAR archetype
    - To generate a new coherence project
  - The wls-maven-plugin
    - For operational and deployment of wls (more on this later)

The Maven Sync Plugin

- Documentation:
  - http://docs.oracle.com/middleware/1212/wls/WLPRG/maven.htm#CHDCEEIA
- Install:
  - mvn -DpomFile=$ORACLE_HOME/oracle_common/plugins/maven/com/oracle/maven/oracle-maven-sync/12.1.2/oracle-maven-sync.12.1.2.pom -Dfile=$ORACLE_HOME/oracle_common/plugins/maven/com/oracle/maven/oracle-maven-sync/12.1.2/oracle-maven-sync.12.1.2.jar install:install-file
- Synchronize:
  - mvn -Doracle-maven-sync.oracleHome=$ORACLE_HOME -Doracle-maven-sync.testOnly=false com.oracle.maven:oracle-maven-sync:12.1.2-0-0:push

The GAR Archetype

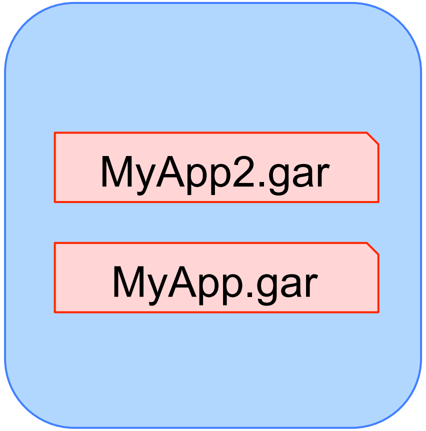
- Generate a coherence GAR maven project
  - mvn archetype:generate -DarchetypeGroupId=com.oracle.coherence -DarchetypeArtifactId=maven-gar-archetype -DarchetypeVersion=12.1.2-0-0 -DgroupId=com.mygroup –DartifactId=mytest

Agenda

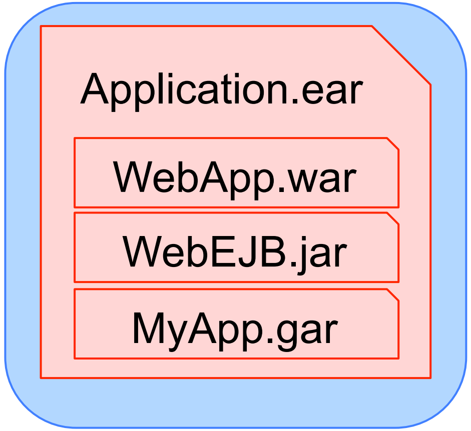
- Definition of terms
- Managed Coherence Servers
- The Grid Archive (GAR)
- Maven Support
- Deployment Models
- Weblogic, Maven and WLST
- Tying it all together

GAR Deployment Modes

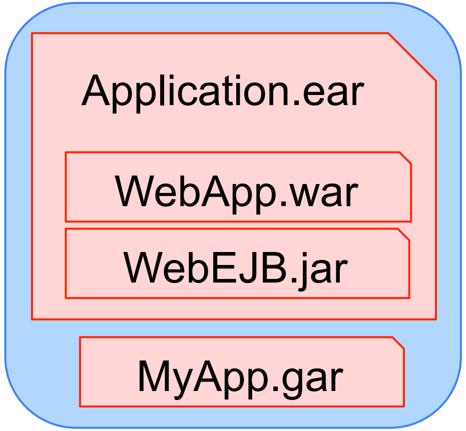
- GAR modules can be deployed to a Managed Server in number of ways to support Coherence based applications



Stand Alone – caches are automatically isolated



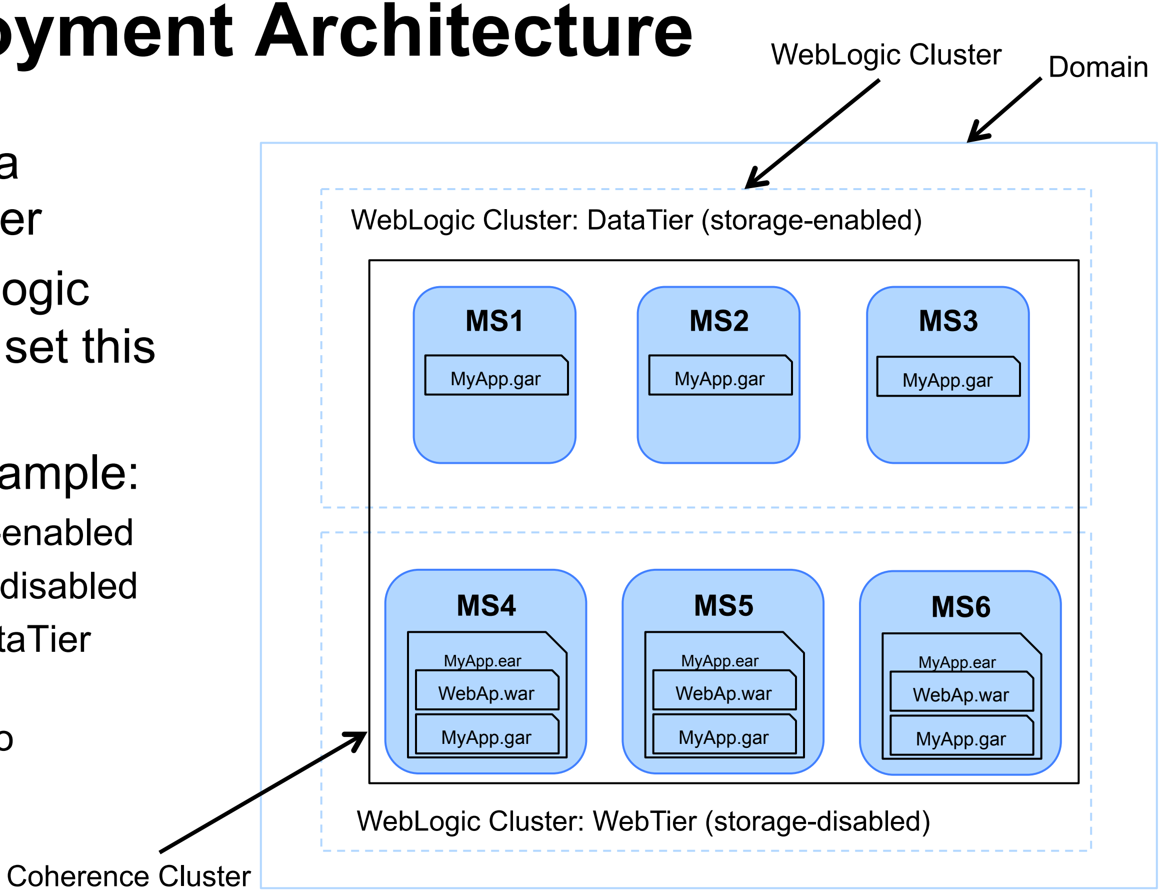
Embedded GAR - available to war/jar files in EAR



Deployed as shared library – available to all WAR/EAR

Example Deployment Architecture

- Best practice is to have a storage-disabled client tier
- We can utilize the WebLogic clustering capabilities to set this in Coherence
- Example deployment example:
  - DataTier cluster is storage-enabled
  - WebTier cluster is storage-disabled
  - MyApp.gar deployed to DataTier cluster
  - MyWebApp.ear deployed to WebCluster



DefaultCacheServer and GARs

- DefaultCacheServer is the traditional way a cluster node is started
- DefaultCacheServer now understands GARs and takes:
  - A path to a GAR or an exploded JAR as an argument
  - An optional name for the GAR deployment – used for scoping and service prefix

```
$ java ... com.tangosol.net.DefaultCacheServer./ExampleGAR.gar ExampleGAR

Oracle Coherence GE 12.1.2.0 <Info> (thread=main, member=n/a): Loaded Application Descriptor from
"file:/var/folders/mn/qvw_qbjn087gr7zzr5vb4jlr0000gn/T/gar-ExampleGAR.674352717341411129/META-INF/coherence-
application.xml";
Oracle Coherence GE 12.1.2.0 <Info> (thread=main, member=n/a): Loaded cache configuration from "file:/var/folders/mn/
qvw_qbjn087gr7zzr5vb4jlr0000gn/T/gar-ExampleGAR.674352717341411129/META-INF/example-cache-config.xml”
Services
  (
  Cluster Service{Name=Cluster, State=(SERVICE_STARTED, STATE_JOINED), Id=0, Version=12.1.2, OldestMemberId=1}
  Invocation Service{Name=Management, State=(SERVICE_STARTED), Id=1, Version=12.1.2, OldestMemberId=1}
  Partitioned Cache{Name=ExampleGAR:PartitionedPofCache, State=(SERVICE_STARTED), LocalStorage=enabled, PartitionCount=257,
BackupCount=1, AssignedPartitions=257, BackupPartitions=0}
  )

Started GarServer...
```

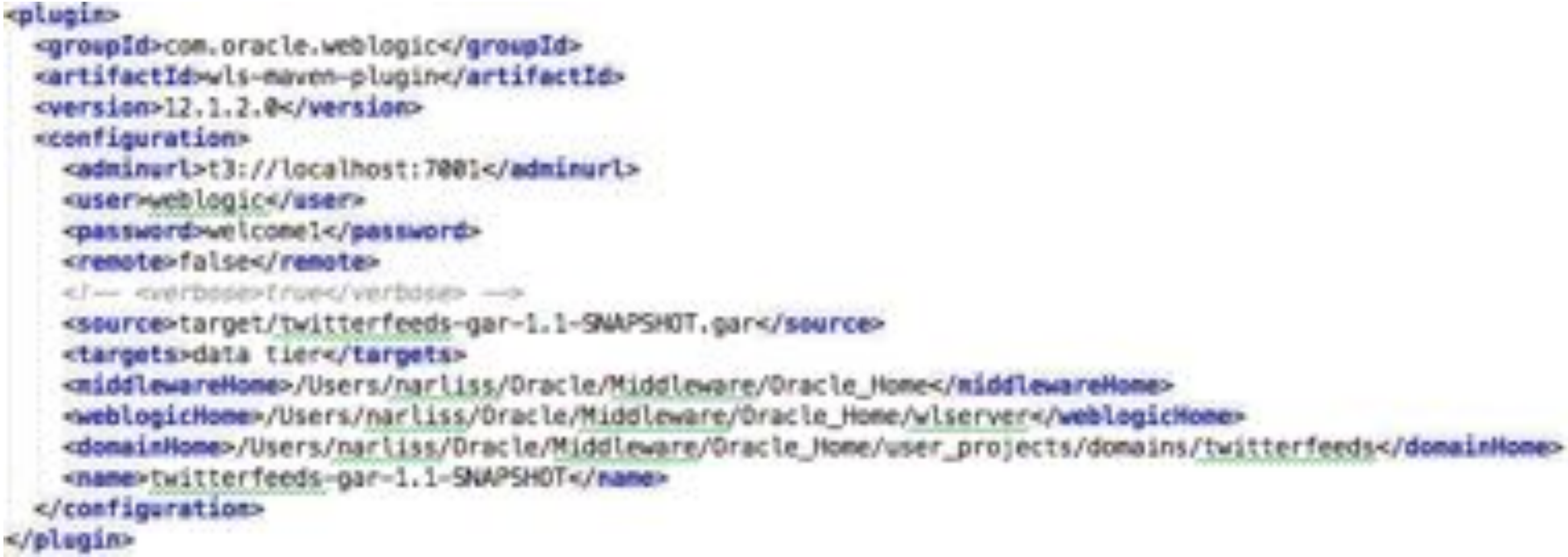
Agenda

- Definition of terms
- Managed Coherence Servers
- The Grid Archive (GAR)
- Maven Support
- Deployment Models
- Weblogic, Maven and WLST
- Tying it all together

The wls-maven-plugin

- Documentation:
  - http://docs.oracle.com/cd/E24329_01/web.1211/e24368/maven.htm#autoId2
- Weblogic Server Goals that support:
  - Deploy/undeploy
  - Domain creation
  - Starting and stopping servers
  - Run wlst
  - Much much more

The wls-maven-plugin (sample config)



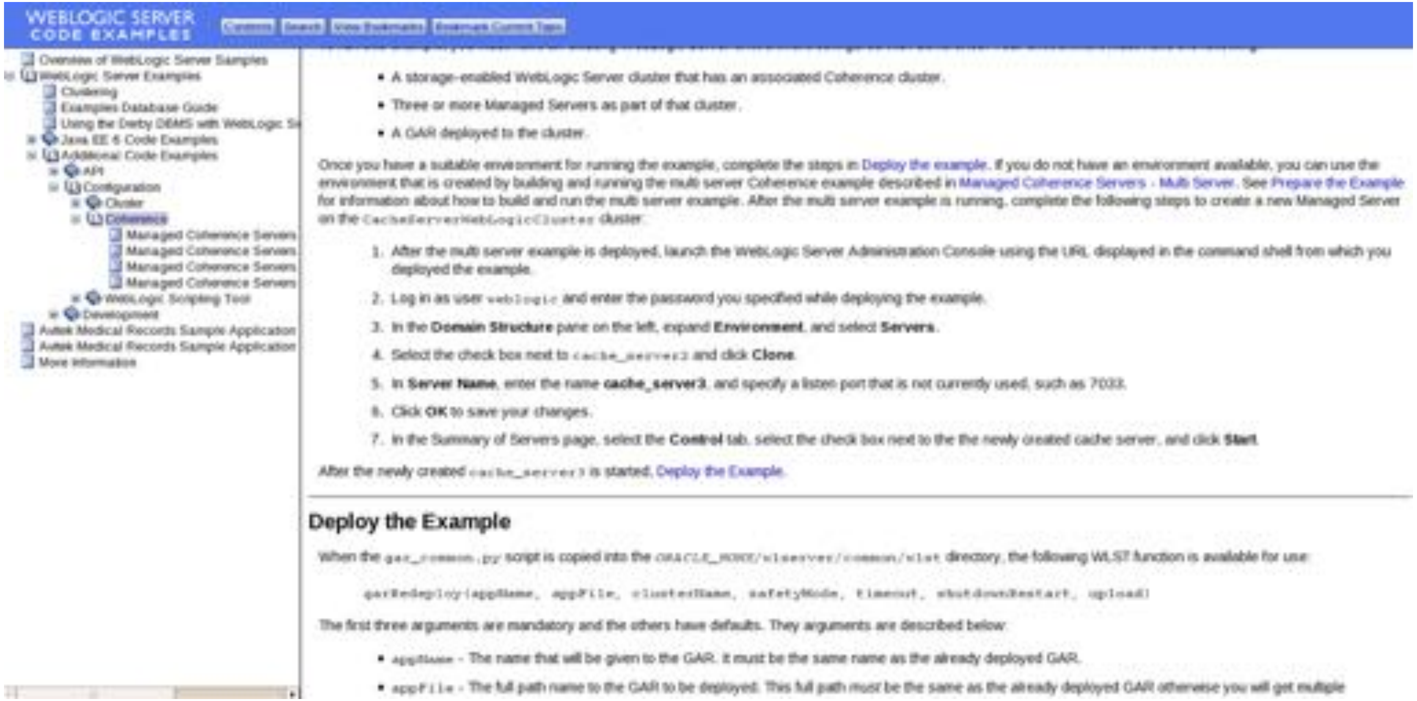
```
<plugin>
  <groupId>com.oracle.weblogic</groupId>
  <artifactId>wls-maven-plugin</artifactId>
  <version>12.1.2.0</version>
  <configuration>
    <adminurl>t3://localhost:7001</adminurl>
    <user>weblogic</user>
    <password>welcome1</password>
    <remote>false</remote>
    <!-- <verbose>true</verbose> -->
    <source>target/twitterfeeds-gar-1.1-SNAPSHOT.gar</source>
    <targets>data tier</targets>
    <middlewareHome>/Users/narliss/Oracle/Middleware/Oracle_Home</middlewareHome>
    <weblogicHome>/Users/narliss/Oracle/Middleware/Oracle_Home/wlserver</weblogicHome>
    <domainHome>/Users/narliss/Oracle/Middleware/Oracle_Home/user_projects/domains/twitterfeeds</domainHome>
    <name>twitterfeeds-gar-1.1-SNAPSHOT</name>
  </configuration>
</plugin>
```

WLST – Weblogic Scripting Tool

- Jython based CLI for Weblogic that supports:
  - http://docs.oracle.com/cd/E15051_01/wls/docs103/config_scripting/using_WLST.html
  - Propagation of domains
  - Retrieval/Setting of domain configuration elements
  - Server Lifecycle Management
  - Automation of domain configuration tasks
  - Much much more…

Coherence Rolling Redeploy Example

- Ships with 12.1.2 complete install examples:



- A storage-enabled WebLogic Server cluster that has an associated Coherence cluster.
- Three or more Managed Servers as part of that cluster.
- A GAR deployed to the cluster.

Once you have a suitable environment for running the example, complete the steps in Deploy the example. If you do not have an environment available, you can use the environment that is created by building and running the multi server Coherence example described in Managed Coherence Servers - Multi Server. See Prepare the Example for information about how to build and run the multi server example. After the multi server example is running, complete the following steps to create a new Managed Server on the `CacheServerWebLogicCluster` cluster.

1. After the multi server example is deployed, launch the WebLogic Server Administration Console using the URL displayed in the command shell from which you deployed the example.
2. Log in as user `weblogic` and enter the password you specified while deploying the example.
3. In the **Domain Structure** pane on the left, expand **Environment**, and select **Servers**.
4. Select the check box next to `cache_server1` and click **Clone**.
5. In **Server Name**, enter the name **cache_server3**, and specify a listen port that is not currently used, such as 7033.
6. Click **OK** to save your changes.
7. In the Summary of Servers page, select the **Control** tab, select the check box next to the the newly created cache server, and click **Start**.

After the newly created `cache_server3` is started, Deploy the Example.

Deploy the Example

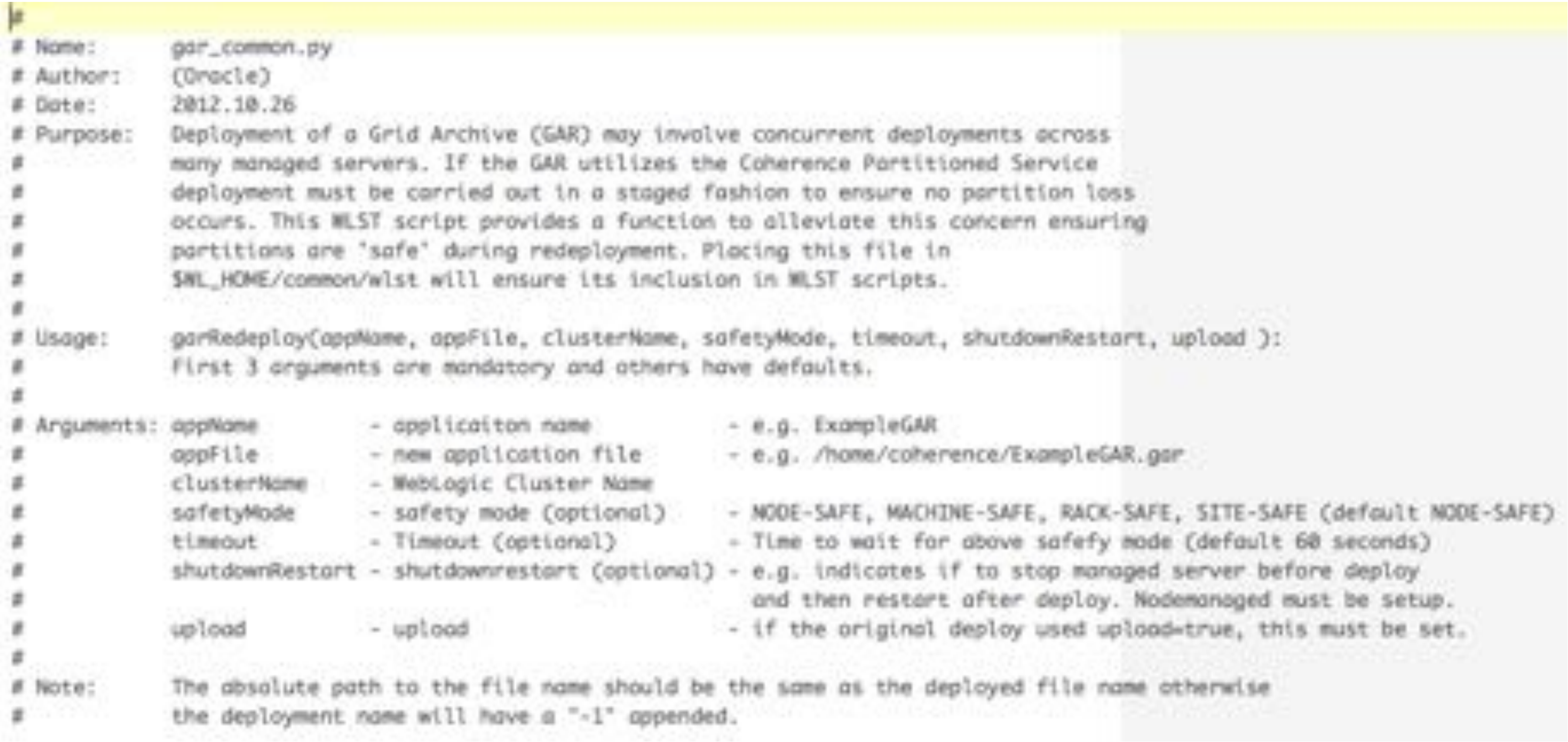
When the `gar_common.py` script is copied into the `ORACLE_HOME/wlserver/common/wlst` directory, the following WLST function is available for use:

```
garRedeploy(appName, appFile, clusterName, safetyMode, timeout, shutdownRestart, upload)
```

The first three arguments are mandatory and the others have defaults. They arguments are described below:

- `appName` - The name that will be given to the GAR. It must be the same name as the already deployed GAR.
- `appFile` - The full path name to the GAR to be deployed. This full path must be the same as the already deployed GAR otherwise you will get multiple

Coherence Rolling Re-deploy Example



```
#
# Name:       gar_common.py
# Author:     (Oracle)
# Date:       2012.10.26
# Purpose:    Deployment of a Grid Archive (GAR) may involve concurrent deployments across
#             many managed servers. If the GAR utilizes the Coherence Partitioned Service
#             deployment must be carried out in a staged fashion to ensure no partition loss
#             occurs. This WLST script provides a function to alleviate this concern ensuring
#             partitions are 'safe' during redeployment. Placing this file in
#             $WL_HOME/common/wlst will ensure its inclusion in WLST scripts.
#
# Usage:      garRedeploy(appName, appFile, clusterName, safetyMode, timeout, shutdownRestart, upload ):
#             First 3 arguments are mandatory and others have defaults.
#
# Arguments: appName         - applicaiton name              - e.g. ExampleGAR
#             appFile         - new application file          - e.g. /home/coherence/ExampleGAR.gar
#             clusterName     - WebLogic Cluster Name
#             safetyMode      - safety mode (optional)        - NODE-SAFE, MACHINE-SAFE, RACK-SAFE, SITE-SAFE (default NODE-SAFE)
#             timeout         - Timeout (optional)            - Time to wait for above safefy mode (default 60 seconds)
#             shutdownRestart - shutdownrestart (optional) - e.g. indicates if to stop managed server before deploy
#                                                             and then restart after deploy. Nodemanaged must be setup.
#             upload          - upload                        - if the original deploy used upload=true, this must be set.
#
# Note:       The absolute path to the file name should be the same as the deployed file name otherwise
#             the deployment name will have a "-1" appended.
```

Tying it all together

- Create an application using maven
- Deploy the application to the domain using maven
- Test the application
- Redeploy the application while the system is running

Join the Coherence Community

http://coherence.oracle.com

- @OracleCoherence
- facebook.com/OracleCoherence
- blogs.oracle.com/OracleCoherence
- Oracle Coherence Users
- youtube.com/OracleCoherence
- coherence.oracle.com/display/CSIG
  Coherence Special Interest Group

ORACLE

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

Coherence Customer Advisory Board

Q&A

ORACLE®