Distributed Configuration Management
Administrator's Guide
10g Release 2 (10.1.2)
Part No. B13997-01

December 2004
# Contents

**Send Us Your Comments** ........................................................................................................................................ vii

**Preface** ............................................................................................................................................................ ix

- Intended Audience .................................................................................................................................................... ix
- Documentation Accessibility ....................................................................................................................................... ix
- Structure ................................................................................................................................................................... x
- Related Documents ................................................................................................................................................ x
- Conventions ........................................................................................................................................................... xi

## 1 Distributed Configuration Management Overview

1.1 What is Distributed Configuration Management? ................................................................................................. 1-1

1.1.1 Elements of a Distributed Configuration ........................................................................................................... 1-2

1.2 Distributed Configuration Management Architecture ................................................................................................... 1-5

1.2.1 The Distributed Configuration Management Bootstrap Sequence ......................................................................... 1-5

## 2 Using the dcmctl Utility

2.1 Managed Configuration Overview ............................................................................................................................. 2-1

2.2 Oracle Application Server Farm Creation and Maintenance Tasks .............................................................................. 2-2

2.2.1 Designating the OracleAS File-based Farm Repository Host ................................................................................ 2-3

2.2.2 Adding Instances to an OracleAS Farm .................................................................................................................. 2-4

2.2.3 Creating a DCM-Managed OracleAS Cluster .......................................................................................................... 2-4

2.2.4 Understanding DCM-Managed OracleAS Cluster Membership ............................................................................... 2-5

2.2.5 Adding an Instance to a DCM-Managed OracleAS Cluster ...................................................................................... 2-8

2.2.6 Removing an OracleAS Instance From a DCM-Managed OracleAS Cluster ......................................................... 2-9

2.2.7 Exporting and Importing Configurations From a File-based Repository ................................................................. 2-9

2.2.8 Moving an Instance From One OracleAS File-based Farm to Another ................................................................. 2-10

2.2.9 Moving An Instance to a OracleAS Database-based Farm .................................................................................... 2-11

2.2.10 Enabling SSL For Communication Between Instances in an OracleAS File-based Farm 2-12

2.2.11 Improving Performance in a Large OracleAS File-based Farm ........................................................................... 2-15

2.2.12 Removing an Instance From the Infrastructure After Executing the leaveFarm Command ............................... 2-15

2.3 Creating and Maintaining OC4J Instances In DCM-Managed OracleAS Clusters .................................................. 2-18

2.3.1 Creating an OC4J Instance ................................................................................................................................... 2-18

2.3.2 Deleting an OC4J Instance ................................................................................................................................... 2-19
3 Archiving A Managed Configuration

3.1 DCM Archiving Commands ................................................................. 3-1
3.2 Initial Archive Creation ................................................................. 3-1
3.3 Archive and Instance Compatibility ........................................... 3-2
3.4 Exporting and Importing Archives .............................................. 3-2
3.5 Using Automatic Archiving .......................................................... 3-3
3.5.1 Tuning Automatic Archiving to Improve Performance .......... 3-5
3.6 Preserving Configurations With Archive Commands ................. 3-5
3.7 Importing and Applying an Archive to an Instance .................. 3-5
3.8 Benefits of Archiving ................................................................. 3-6

A dcmctl Commands

A.1 Notes for Using dcmctl Commands ............................................. A-1
A.1.1 dcmctl Options ................................................................. A-2
A.1.2 Using the dcmctl Shell ......................................................... A-4
A.2 Types of dcmctl Commands ...................................................... A-4
A.3 Alphabetical Listing of dcmctl Commands ................................ A-5
A.3.1 !! .......................................................... A-6
A.3.2 addOPMNLink ................................................................. A-6
A.3.3 applyArchiveTo ................................................................. A-7
A.3.4 applyClusterTo ................................................................. A-7
A.3.5 applyInstanceTo ................................................................. A-8
A.3.6 configRepositorySSL ......................................................... A-8
A.3.7 createArchive ................................................................. A-9
A.3.8 createCluster ................................................................. A-10
A.3.9 createComponent .............................................................. A-10
A.3.10 deployApplication ........................................................... A-11
A.3.11 destroyInstance ............................................................... A-12
A.3.12 echo ................................................................. A-13
A.3.13 exit ................................................................. A-13
A.3.14 exportArchive ................................................................. A-14
A.3.15 exportRepository ............................................................. A-14
A.3.16 fileCleanUp ................................................................. A-15
A.3.17 getComponentType ........................................................ A-16
A.3.18 getError ................................................................. A-16
A.3.19 getOPMNPorCode ... code ................................................ A-17
A.3.20 getRepositoryId ............................................................ A-17
A.3.21 getReturnStatus ............................................................. A-18
A.3.22 getState ................................................................. A-18
A.3.23 help ................................................................. A-19
A.3.24 importArchive ............................................................... A-20
A.3.25 importRepository ........................................................... A-20
B Troubleshooting DCM

B.1 Problems and Solutions ................................................................. B-1
  B.1.1 OPMN Cannot Start ................................................................. B-2
  B.1.2 DCM Daemon Cannot Start ................................................... B-5
  B.1.3 DCM Daemon Does Not Start As Expected ............................. B-5
  B.1.4 DCM Operations Fail ............................................................. B-6
  B.1.5 Invalid username/password ..................................................... B-6
Send Us Your Comments

Distributed Configuration Management Administrator’s Guide, 10g Release 2 (10.1.2)
Part No. B13997-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

■ Did you find any errors?
■ Is the information clearly presented?
■ Do you need more information? If so, where?
■ Are the examples correct? Do you need more examples?
■ What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

■ Electronic mail: appserverdocs_us@oracle.com
■ FAX: 650.506.7365 Attn: Oracle Application Server
■ Postal service:
  Oracle Corporation
  Oracle Application Server Documentation
  500 Oracle Parkway, M/S 1op6
  Redwood Shores, California 94065
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.
This guide describes the Distributed Configuration Management framework and describes the `dcmctl` utility command syntax and usage. It includes a troubleshooting section that identifies common problems and how to diagnose and solve them.

This guide does not discuss Oracle Application Server configurations, or recommend how to manage these.

**Intended Audience**

This manual is intended as a reference for system administrators who use the `dcmctl` utility, or those seeking a summary of Distributed Configuration Management architecture and capabilities.

**Documentation Accessibility**

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

**Accessibility of Code Examples in Documentation**

JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation**

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.
Structure

This guide contains the following chapters:

**Chapter 1, "Distributed Configuration Management Overview"**
This chapter describes the Distributed Configuration Management (DCM) architecture and functionality.

**Chapter 2, "Using the dcmctl Utility"**
This chapter explains how to use the dcmctl command-line utility to set up and manage configurations.

**Chapter 3, "Archiving A Managed Configuration"**
This chapter explains how to use DCM archiving functions.

**Appendix A, "dcmctl Commands"**
This appendix contains a listing of all DCM commands, providing a description of the command, its syntax, and usage examples.

**Appendix B, "Troubleshooting DCM"**
This appendix lists problems that might occur when you use DCM and one or more solutions. It also contains approaches to troubleshooting problems with components related to DCM.

Related Documents

These documents may also be of interest to the system administrator using DCM:

- [Oracle Application Server Concepts](#)
- [Oracle Application Server Administrator’s Guide](#)
- [Oracle Process Manager and Notification Server Administrator’s Guide](#)
- [Oracle Application Server High Availability Guide](#)
## Conventions

The following conventions are also used in this manual:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>...</td>
<td>Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted</td>
</tr>
<tr>
<td><strong>monospace text</strong></td>
<td>File names, path names, command names, code, URLs</td>
</tr>
<tr>
<td><strong>monospace bold text</strong></td>
<td>Typed user input.</td>
</tr>
<tr>
<td><strong>monospace italic text</strong></td>
<td>Variables in text or code.</td>
</tr>
<tr>
<td>/</td>
<td>Used in all directory paths; assumed equivalent to the backward slash convention for Windows.</td>
</tr>
</tbody>
</table>
This chapter briefly describes the Distributed Configuration Management (DCM) architecture and functionality.

This chapter covers the following topics:

- What is Distributed Configuration Management?
- Distributed Configuration Management Architecture

1.1 What is Distributed Configuration Management?

Distributed Configuration Management is a management framework that enables you to create and manage multiple Oracle Application Server Instances that you can manage as a single Oracle Application Server Instance. Multiple Oracle Application Server Instances enable the application server to handle large volumes of traffic reliably, since the workload is distributed among multiple Oracle Application Server Instances.

Distributed Configuration Management enables you to:

- Keep a configuration synchronized across multiple Oracle Application Server Instances
- Archive and restore versions of configurations
- Export and import configurations between Oracle Application Server Instances

With DCM, you can manage configuration information for the following Oracle Application Server components and applications:

- Oracle HTTP Server (OHS)
- Oracle Application Server Containers for J2EE (OC4J)
- Oracle Process Manager and Notification Server (OPMN)
- Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider (JAZN)
- J2EE Applications
1.1.1 Elements of a Distributed Configuration

Distributed configuration management enables you to synchronize the configuration of, archive, and import and export multiple Oracle Application Server Instances as if they were a single Oracle Application Server Instance. This guide refers to such configurations as “DCM-configured”. To provide the management functionality, DCM keeps information about an Oracle Application Server Instance’s configuration in the DCM Metadata Repository. The repository and the Oracle Application Server Instances comprise the distributed configuration. In this section, each element of the distributed configuration and its relationship to other elements is described in detail.

1.1.1.1 What is an Oracle Application Server Instance?

The fundamental element of the distributed configuration is the Oracle Application Server Instance: the set of processes required to operate the components that are configured. The terms “instance” and “installation” are sometimes used interchangeably, but note the distinction: an instance is the set of processes that operate the components, whereas an installation is the set of files installed into an Oracle home. The distinction is important in the context of DCM management, because the DCM management scope does not extend to all files installed in the Oracle home—only those related to the set of processes that are needed to operate the configured components.

1.1.1.2 What is a DCM-Managed Oracle Application Server Cluster?

A DCM-Managed OracleAS Cluster is a collection of Oracle Application Server Instances with identical configuration and application deployment characteristics. To become members of a DCM-Managed OracleAS Cluster, Oracle Application Server Instances must be of the same installation type and version, reside on a like operating system (for example, UNIX and Linux) and contain only Oracle HTTP Server, OC4J, OPMN, and JAZN components.

1.1.1.3 What is an Oracle Application Server Farm?

An Oracle Application Server Farm is a collection of OracleAS Clusters and Oracle Application Server Instances that share the same DCM Metadata Repository. The shared repository is called the repository host. Section 1.1.1.4.2, "File-based Repository (Repository Host)” describes the repository host and its use.

An Oracle Application Server Farm may be of one of the following types: an OracleAS File-based Farm or an OracleAS Database-based Farm.

1.1.1.4 Understanding the DCM Metadata Repository

In order to manage configurations, DCM stores information about the configuration (called configuration metadata in this guide) in the DCM Metadata Repository.

All configuration data is stored in the DCM Metadata Repository. The DCM Metadata Repository is a distinct metadata repository that is not dependent on the Oracle Application Server Metadata Repository.

The DCM Metadata Repository contains:

- Configuration files for Oracle HTTP Server, OC4J, OPMN, and JAZN components
- Deployed J2EE applications
- Information about the Oracle Application Server Instance or OracleAS Cluster

A DCM Metadata Repository is stored in the file system or in the database, in one of the following configurations:
What is Distributed Configuration Management?

- Oracle Application Server File-based Farm with a standalone Oracle Application Server Instance (described in Section 1.1.1.4.1, "File-based Repository (Standalone Oracle Application Server Instance)" on page 1-3)

- Oracle Application Server File-based Farm with repository host (described in Section 1.1.1.4.2, "File-based Repository (Repository Host)" on page 1-3)

- Oracle Application Server Database-based Farm (described in Section 1.1.1.4.3, "Database-based repository" on page 1-3)

1.1.1.4.1 File-based Repository (Standalone Oracle Application Server Instance) A single Oracle Application Server Instance that is not part of an OracleAS Cluster or Farm is called a standalone Oracle Application Server Instance. Every Oracle Application Server Instance has a local file-based repository. When an Oracle Application Server Instance is associated with an OracleAS File-based Farm, but is not the repository host, the local file-based repository contains the Bill of Materials (BOM) that DCM uses to validate that the Oracle Application Server Instance’s configuration is synchronized with the configuration metadata in the repository. When the Oracle Application Server Instance is not associated with an Oracle Application Server Farm, the local file-based repository is the only stored representation of the Oracle Application Server Instance’s configuration information.

1.1.1.4.2 File-based Repository (Repository Host) When an Oracle Application Server Instance is defined as the repository host for an OracleAS File-based Farm, the repository for that Oracle Application Server Instance contains the configuration metadata for all Oracle Application Server Instances in the OracleAS File-based Farm. Since the repository host Oracle Application Server Instance stores configuration information on its file system, the repository host Oracle Application Server Instance should use mirrored or RAID disks to increase availability. However, when the repository host Oracle Application Server Instance is unavailable:

- OracleAS Clusters using it still function normally, but cannot update any configuration information.

- Read-only configuration operations are not affected on any running Oracle Application Server Instance (the Oracle Application Server Farm’s cluster-wide configuration information is distributed and managed through the local repository.

- Operations that attempt to change configuration information in the repository will cause an error. These operations must be delayed until the repository host Oracle Application Server Instance is available, or until the repository host Oracle Application Server Instance is relocated to another Oracle Application Server Instance within the Oracle Application Server Farm (see repositoryRelocated).

1.1.1.4.3 Database-based repository A Database-based repository is comprised of DCM schema. Storing the DCM Metadata Repository in a database may be useful as part of a site’s high availability and backup strategy. Using a Database-based repository, the database serves as the repository host.

For all three types of DCM Metadata Repository: Database-based repository, file-based repository in standalone mode, or file-based repository host mode, an Oracle Application Server Instance always has a local file-based repository. In cases in which the Oracle Application Server Instance is not included in an OracleAS Farm, the local file-based repository is the sole persistent storage for that Oracle Application Server Instance’s configuration metadata.
When you make configuration changes using either Oracle Enterprise Manager 10g Application Server Control Console or the dcmct1 utility, the configuration management system updates the DCM Metadata Repository to reflect your changes.

**See Also:** Section 2.2.1, "Designating the OracleAS File-based Farm Repository Host" on page 2-3, Section 2.2.2, "Adding Instances to an OracleAS Farm" on page 2-4 and Section 2.2.3, "Creating a DCM-Managed OracleAS Cluster" on page 2-4 for information on setting up the file-based repository and working with Oracle Application Server Instances and OracleAS Clusters.

### 1.1.1.5 Understanding Synchronization of the Metadata Repository and the Oracle Application Server Instance

The DCM Metadata Repository is the definitive source for DCM-managed configuration information. If there is a difference between the Oracle Application Server Instance configuration stored in the repository and the Oracle Application Server Instance configuration in the associated ORACLE_HOME file system, the configuration in the file system is updated with the configuration in the repository. When the DCM repository and the file system configuration information are identical, the configuration is synchronized; its **In Sync Status** is true. (See `getstate` and `resyncInstance`.)

Immediately after a configuration change, DCM automatically attempts to resynchronize the members of an OracleAS Cluster. If an Oracle Application Server Instance in an OracleAS Cluster is not available, the resynchronization occurs the next time the DCM daemon on the Oracle Application Server Instance is started. The DCM daemon can be started manually with the `opmnctl startproc ias-component="dcm-daemon"` command.

When a configuration change is made in a Oracle Application Server Farm or in a OracleAS Cluster, DCM attempts to ensure that the change will be successful by first applying the change to the local Oracle Application Server Instance, before attempting to propagate the change to other Oracle Application Server Instances. If the configuration change fails in the local Oracle Application Server Instance, its effects are automatically rolled back; no changes are made to any Oracle Application Server Instance.

In some cases, a configuration change may succeed on the local Oracle Application Server Instance, but fail on other Oracle Application Server Instances in the Oracle Application Server Farm or OracleAS Cluster. This could occur for many reasons, including insufficient disk space, file system security, or connectivity from the Oracle Application Server Instance to a dependent service such as Oracle Internet Directory or the database. In these cases, the In Sync status of the Oracle Application Server Instance is set to **false**.

When the In Sync status is set to **false**, the event is recorded, with details, in the DCM log file. When the problem that caused the status to be set to false is resolved, you should resynchronize the Oracle Application Server Instance, using the `dcmct1 resyncinstance` command to copy the configuration stored in the repository for an Oracle Application Server Instance to the file system for the Oracle Application Server Instance (see `resyncInstance`).

The `updateConfig` command is another synchronization command that requires special handling. The `updateConfig` command takes configuration information from the file system and places this configuration information in the DCM repository. Ensure that you understand the guidelines for using the `updateConfig` command before you use it (see `updateConfig`).

---

What is Distributed Configuration Management?
1.2 Distributed Configuration Management Architecture

Distributed Configuration Management consists of clients, a daemon, and the DCM Metadata Repository. Figure 1–1 shows the relationship between these and other Oracle Application Server components. The clusterable plugin’s configuration is managed within the DCM Metadata Repository; clusterable plugins have a copy of persistent data stored in the DCM Metadata Repository. Non-clusterable plugins do not.

Figure 1–1 Distributed Configuration Management Architecture

The Oracle Enterprise Manager 10g Application Server Control Console and dcmctl contain the DCM client JAR file.

1.2.1 The Distributed Configuration Management Bootstrap Sequence

It is helpful at times to understand the order in which components are started upon initialization. The DCM bootstrap sequence is as follows:

1. The DCM client checks to determine whether Oracle Process Manager and Notification Server is running.
   a. If not, it starts Oracle Process Manager and Notification Server.
   b. If yes, it discovers and uses it.

2. The DCM client checks to determine whether the DCM daemon is running.
   a. If not, it starts the daemon.
b. If yes, it discovers and uses it.

---

**Note:** Do not stop the DCM daemon explicitly (using the `opmnctl stopall` command) when an active DCM client is in use. If you stop the DCM daemon explicitly, then you must restart the DCM client.

---

3. The DCM daemon checks the configuration file version of Oracle HTTP Server, OC4J, Oracle Process Manager and Notification Server, and Java Authentication and Authorization Service (using the configurable plug-ins shown in Figure 1–1).

4. The DCM daemon updates the configuration file versions, if required.

5. The DCM daemon restarts Oracle Process Manager and Notification Server, if required.
This chapter explains how to use the `dcmctl` utility to create and manage the elements of a managed configuration: Oracle Application Server File-based Farm, DCM-Managed Oracle Application Server Cluster, Oracle Application Server Instance, and component instance. Database-based repository creation and maintenance tasks are not documented here, because they are performed with tools other than the `dcmctl` utility—namely, the Oracle Universal Installer and the Repository Creation Assistant. See the Oracle Application Server Administrator’s Guide for those instructions.

This chapter contains the following topics:

- Managed Configuration Overview
- Oracle Application Server Farm Creation and Maintenance Tasks
- Creating and Maintaining OC4J Instances In DCM-Managed OracleAS Clusters
- Best Practices for DCM-Managed OracleAS Cluster Management

2.1 Managed Configuration Overview

The managed configuration elements are related as shown in Figure 2–1. "Managed" means that a repository (file-based or database) is involved in administration. DCM-Managed OracleAS Cluster configurations can be created and kept in synch on an individual, manual basis, but that method is not discussed here.

The broadest scope of administration of a DCM-managed configuration is the Oracle Application Server Farm, in which DCM-Managed OracleAS Clusters of OracleAS Instances and non-clustered, or standalone, OracleAS Instances may reside. An OracleAS Farm may be created during installation, or with the dcmctl utility. Figure 2–1 depicts the elements, relationships, and cardinalities within an OracleAS Farm.
2.2 Oracle Application Server Farm Creation and Maintenance Tasks

An OracleAS Farm is defined as a collection of OracleAS Clusters and OracleAS Instances that share the same DCM metadata repository. The OracleAS Farm is the scope of administration for the OracleAS Clusters and OracleAS Instances that share a repository.

This section explains how to set up an OracleAS File-based Farm and add OracleAS Instances to it, create and maintain DCM-Managed OracleAS Clusters of the OracleAS Instances, and perform other system administration tasks. The following topics are included:

- Designating the OracleAS File-based Farm Repository Host
- Adding Instances to an OracleAS Farm
2.2.1 Designating the OracleAS File-based Farm Repository Host

You must designate the OracleAS Instance that will be the repository host of the OracleAS File-based Farm. The OracleAS Instance you designate must be a standalone OracleAS Instance (that is, not associated with any OracleAS Farm). Follow these steps to designate the repository host of a OracleAS File-based Farm:

1. Navigate to the OracleAS Instance that will be the repository host OracleAS Instance.

2. Determine whether it is a standalone OracleAS Instance by issuing this command:
   \[\text{dcmctl whichFarm}\]
   If \text{dcmctl} returns an OracleAS File-based Farm name, the OracleAS Instance is still part of an OracleAS File-based Farm, and is thus still associated with an existing repository. To place the OracleAS Instance in a standalone state:
   a. Issue this command:
      \[\text{dcmctl leaveFarm}\]
      The OracleAS Instance is now in a standalone state, and all components running in it are stopped.

3. Issue this command:
   \[\text{dcmctl getRepositoryID}\]
   \text{dcmctl} returns a repository identifier in the format \text{hostname:port}.

4. Issue this command:
   \[\text{dcmctl joinFarm -r repository ID}\]
   In the preceding command, \text{repository ID} is the \text{hostname:port} value returned by the \text{getRepositoryID} command.
   The current OracleAS Instance is established as the repository host instance of the OracleAS File-based Farm. The OracleAS File-based Farm configuration information is stored in the file-based repository in the repository host OracleAS Instance.
2.2.2 Adding Instances to an OracleAS Farm

An OracleAS Instance can be part of an OracleAS Farm as a standalone OracleAS Instance (without also being a DCM-Managed OracleAS Cluster member) as shown in Figure 2–1. This makes it possible to manage as one unit a collection of UNIX, Windows, and Linux OracleAS Instances. (Instances of like operating systems may be clustered; however, before an OracleAS Instance can join a DCM-Managed OracleAS Cluster, it must be in the OracleAS Farm). Follow these steps to bring an OracleAS Instance into an OracleAS File-based Farm:

1. Obtain the repository ID of the repository host OracleAS Instance of the OracleAS File-based Farm the OracleAS Instance will join (if necessary, by issuing this command):

   
   dcmctl getRepositoryID

   dcmctl returns a repository identifier in the format hostname:port.

2. In the OracleAS Instance that will join the OracleAS File-based Farm, issue this command:

   
   dcmctl joinFarm -r repository ID

   In the preceding command, repository ID is the repository identifier returned in Step 1.

3. (Optional) Verify that the OracleAS File-based Farm was joined as expected by issuing this command:

   
   dcmctl whichfarm -v

   dcmctl returns the farm name, host instance name, host name and repository type.

2.2.3 Creating a DCM-Managed OracleAS Cluster

A DCM-Managed OracleAS Cluster is created as a member of an OracleAS Farm, as depicted in Figure 2–1. Follow these steps to create a DCM-Managed OracleAS Cluster:

1. Navigate to the Oracle home of an OracleAS Instance that belongs to the OracleAS Farm in which you want to create a DCM-Managed OracleAS Cluster.

2. Issue this command:

   
   dcmctl createCluster -cl nameOfCluster

   In the preceding command, nameOfCluster is the identifier for the DCM-Managed OracleAS Cluster.

   **Tip:** See createCluster for syntax and rules, or issue the command dcmctl help createcluster to view help text.

   dcmctl lists the DCM-Managed OracleAS Clusters as shown:

   1  cluster2
   2  cluster1

   The DCM-Managed OracleAS Cluster does not have any member OracleAS Instances upon creation. You add OracleAS Instances to the DCM-Managed OracleAS Cluster using the joinCluster command.
2.2.4 Understanding DCM-Managed OracleAS Cluster Membership

After a DCM-Managed OracleAS Cluster is created, you can add OracleAS Instances to it. Before you add an OracleAS Instance to a DCM-Managed OracleAS Cluster, review and understand the following guidelines about DCM-Managed OracleAS Cluster configuration and the characteristics of clusterable OracleAS Instances.

2.2.4.1 How the Common Configuration is Established

The order in which the OracleAS Instances are added to the DCM-Managed OracleAS Cluster is significant. The common configuration that will be replicated across the DCM-Managed OracleAS Cluster is established by the first OracleAS Instance added to the cluster. The configuration of the first OracleAS Instance added is inherited by all OracleAS Instances that subsequently join the DCM-Managed OracleAS Cluster.

The common configuration includes all cluster-wide configuration information—namely, DCM-Managed OracleAS Cluster and OracleAS Instance attributes, such as components configured. For example, if the first OracleAS Instance to join the cluster has four OC4J instances, then the common configuration includes those four OC4J instances and the applications deployed on the OC4J instances. Instances that subsequently join the DCM-Managed OracleAS Cluster replicate the OC4J instances and their deployed applications. (In addition, when the OracleAS Instance joins the DCM-Managed OracleAS Cluster, DCM removes any OC4J components that do not match the common configuration). Furthermore, changes to one OracleAS Instance in the DCM-Managed OracleAS Cluster, such as adding new OC4J instances or removing OC4J instances, are replicated across the DCM-Managed OracleAS Cluster; the components configured are part of the replicated cluster-wide, common configuration.

When the last OracleAS Instance leaves a DCM-Managed OracleAS Cluster, the DCM-Managed OracleAS Cluster becomes an empty DCM-Managed OracleAS Cluster, and the next OracleAS Instance to joins the DCM-Managed OracleAS Cluster provides the new common configuration for the DCM-Managed OracleAS Cluster.

2.2.4.2 Parameters Excluded from the Common Configuration: Instance-Specific Parameters

Some parameters only apply to a given OracleAS Instance or computer, and are therefore not propagated automatically to all of the OracleAS Instances in a DCM-Managed OracleAS Cluster. Therefore, when changing any of these parameters, you must apply the changes to the appropriate OracleAS Instance.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>island definitions</td>
<td>Specific to an OracleAS Instance. Stateful OC4J applications require that all of the islands in each OC4J instance across the DCM-Managed OracleAS Cluster have the same name.</td>
</tr>
<tr>
<td>number of processes</td>
<td>Specific to a computer. You may want to tune this parameter according to the computer’s capabilities.</td>
</tr>
<tr>
<td>command-line options</td>
<td>Specific to a computer.</td>
</tr>
<tr>
<td>port numbers for RMI, JMS and AJP communication</td>
<td>Specific to a computer.</td>
</tr>
</tbody>
</table>
### Table 2–2  **Oracle HTTP Server Instance-Specific Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApacheVirtualHost</td>
<td>Specific to a computer.</td>
</tr>
<tr>
<td>Listen</td>
<td>Specific to a computer. This directive binds the server to specific addresses or ports.</td>
</tr>
<tr>
<td>OpmnHostPort</td>
<td>Specific to a computer.</td>
</tr>
<tr>
<td>Port</td>
<td>Specific to a computer. This directive specifies the port to which the standalone server listens.</td>
</tr>
<tr>
<td>User</td>
<td>Specific to a computer.</td>
</tr>
<tr>
<td>Group</td>
<td>Specific to a computer.</td>
</tr>
<tr>
<td>NameVirtualHost</td>
<td>Specific to a computer. This directive specifies the IP address on which the server receives requests for a name-based virtual host. This directive can also specify a port.</td>
</tr>
<tr>
<td>ServerName</td>
<td>Specific to a computer. This directive specifies the host name that the server should return when creating redirection URLs. This directive is used if <code>gethostbyname</code> does not work on the local host. You can also use it if you want the server to return a DNS alias as a host name (for example, <a href="http://www.abcompany.com">www.abcompany.com</a>).</td>
</tr>
<tr>
<td>PerlBlob</td>
<td>Specific to a computer.</td>
</tr>
</tbody>
</table>

### Table 2–3  **Oracle Process Manager and Notification Server Instance-Specific Parameters**

<table>
<thead>
<tr>
<th>Parameter in opmn.xml file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All configuration for the notification server: opmn/notification-server</td>
<td>Specific to a computer.</td>
</tr>
<tr>
<td>process-manager elements:</td>
<td>Specific to an OracleAS Instance.</td>
</tr>
<tr>
<td>log-file</td>
<td></td>
</tr>
<tr>
<td>process-module</td>
<td></td>
</tr>
<tr>
<td>ias_instance attributes:</td>
<td>Specific to an OracleAS Instance.</td>
</tr>
<tr>
<td>id</td>
<td></td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td></td>
</tr>
<tr>
<td>ORACLE_CONFIG_HOME</td>
<td></td>
</tr>
<tr>
<td>The following elements and attributes of opmn/process-manager/ias-instance /ias-component/process-type port.range start stop ping restart process-set</td>
<td>Specific to an OracleAS Instance. Although instance-specific, these elements and attributes have default configurations (the configurations are not propagated, but retrieved from repository). The <code>MissingLocalValuePolicy</code> flag indicates that element or attribute has a default value: <code>MissingLocalValuePolicy=&quot;UseRepositoryValue&quot;</code></td>
</tr>
</tbody>
</table>
2.2.4.3 Rules and Guidelines for Adding Instances to DCM-Managed OracleAS Clusters

Instances that are to be added to DCM-Managed OracleAS Clusters are subject to the following restrictions:

- Instances in an DCM-Managed OracleAS Cluster must be of the same installation type and version and reside on a like operating system (Solaris, Linux, and HP-UX are like operating systems).
- Instances in an DCM-Managed OracleAS Cluster must contain only Oracle HTTP Server, OC4J, OPMN and OracleAS JAAS Provider (JAZN) components.
- Configuration changes can be made in any OracleAS Instance in the DCM-Managed OracleAS Cluster, and will be propagated across the

<table>
<thead>
<tr>
<th>Parameter in opmn.xml file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>opmn/process-manager/ias-instance</td>
<td>The following module-data/category/data[id='config-file'] attributes and elements: ias-component/module-data/category/data[id='config-file'] ias-component/process-type/module-data/category/data[id='config-file'] ias-component/process-type/process-set/module-data/category/data[id='config-file'] environment ias-component/environment ias-component/process-type/environment ias-component/process-type/process-set/environment</td>
</tr>
<tr>
<td>All other components whose id is not HTTP_Server or OC4J in opmn/process-manager/ias-instance/ias-component: [id='dcm-daemon'] [id='WebCache'] [id='OID'] [id='IASPT'] [id='wireless'] [id='ProcessConnect'] [id='ReportsServer1'] [id='Discoverer'] [id='LogLoader'] [id='Custom']</td>
<td>Most of the HTTP_Server and OC4J parameters are cluster-wide; only those shown here are instance-specific. In general: Any data element whose id is config-file is instance-specific. Any environment element is instance-specific. Note: Example B–1 in Appendix B, &quot;Troubleshooting DCM&quot; on page B-4, shows a complete opmn.xml file that shows the hierarchy of these elements and attributes.</td>
</tr>
</tbody>
</table>
DCM-Managed OracleAS Cluster from there. However, it is probably a good idea to designate one OracleAS Instance as the origin of configuration changes.

2.2.4.4 Naming Restrictions For Instances and Applications

Do not use the host name of the computer that contains the Oracle Application Server installation when naming applications, OC4J instances, OracleAS Instances, or J2EE applications. In a clustered environment, in particular, do not use the host name, Oracle home, or the IP address of any Oracle Application Server installation in the DCM-Managed OracleAS Cluster.

For example, if your computer has the host name of abc.com, you should not create a new OC4J instance or J2EE application that consists of, or contains, the string abc.com. This rule also applies to the Oracle home directory path or the IP address of the computer.

The reason for this restriction has to do with the clustering, archival, and cloning features of DCM. When archiving or capturing configuration parameters, DCM scans all configuration files for occurrences of the host name, Oracle home and IP address. When it finds occurrences of these values, it replaces the values with symbolic macros and stores them in the DCM Repository.

When DCM applies these configurations to other installations, the occurrences of host name, Oracle home and IP address are replaced with the values for that installation to which the configuration is being applied. Consequently, if OC4J instance names or J2EE application names contain host names, Oracle home names or IP addresses, they are inappropriately substituted if a configuration is applied to other installations.

2.2.5 Adding an Instance to a DCM-Managed OracleAS Cluster

Follow these instructions to add an OracleAS Instance to an DCM-Managed OracleAS Cluster:

1. (Optional) Find the names of the DCM-Managed OracleAS Clusters in the local OracleAS Farm by issuing this command:

   `dcmctl listClusters`

   `dcmctl` returns a list of DCM-Managed OracleAS Cluster names that belong to the OracleAS Farm that is associated with the local OracleAS Instance.

2. Navigate to the OracleAS Instance that you want to join to the DCM-Managed OracleAS Cluster.

3. (Optional) Determine whether the OracleAS Instance can be added to a DCM-Managed OracleAS Cluster by issuing this command:

   `dcmctl isClusterable`

   If `dcmctl` returns `true`, then continue with Step 4.

4. Issue this command:

   `dcmctl joinCluster -cl nameOfCluster`

   In the preceding command, `nameOfCluster` is the identifier for the DCM-Managed OracleAS Cluster you want the OracleAS Instance to join.
The OracleAS Instance is joined to the DCM-Managed OracleAS Cluster and stopped. The `dcmctl joincluster` command lists the OracleAS Instances in the DCM-Managed OracleAS Cluster in the following format:

```
1  10g01.my.pc.mycompany.com
2  10g02.my.pc.mycompany.com
```

5. (Optional) Start the OracleAS Instance by issuing this command:

```
opmnctl startall
```

### 2.2.6 Removing an OracleAS Instance From a DCM-Managed OracleAS Cluster

Follow these instructions to remove an OracleAS Instance from a DCM-Managed OracleAS Cluster:

1. (Optional) Find the names of the OracleAS Instances in the DCM-Managed OracleAS Cluster by issuing this command:

```
dcmctl listInstances -cl nameOfCluster
```

   `dcmctl` returns a list of OracleAS Instance names in the DCM-Managed OracleAS Cluster.

2. Navigate to the OracleAS Instance that you want to remove from the DCM-Managed OracleAS Cluster.

3. Issue this command:

```
dcmctl leaveCluster
```

**Tip:** See `leaveCluster` for instructions on removing an DCM-Managed OracleAS Cluster from an OracleAS Instance other than the local OracleAS Instance, or issue the command `dcmctl help leavecluster` to view help text.

### 2.2.7 Exporting and Importing Configurations From a File-based Repository

One of the benefits of managing configurations with DCM is that configuration information is portable from one OracleAS Instance to another. This section explains how to save and restore a repository and move an OracleAS Instance from one repository to another.

Oracle Application Server provides commands to save a file-based repository, so that you can restore it if the repository host OracleAS Instance stops functioning, or its file system is damaged. The `exportRepository` command saves the file-based repository; then, with the `importRepository` command, you can restore the saved configuration information to the repository host OracleAS Instance, or to a different OracleAS Instance in the OracleAS File-based Farm.

To export the repository from the repository host OracleAS Instance, issue this command:

```
dcmctl exportRepository -file file name
```
In the preceding command, *file name* is the name of the file in which the configuration information will be stored.

Follow these steps to import a saved file-based repository:

1. Stop the DCM daemon in each OracleAS Instance of the OracleAS File-based Farm except for the current OracleAS Instance (in which you are using `dcmctl`). Issue the following command in each OracleAS Instance of the OracleAS File-based Farm:

   ```
   ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=dcm-daemon
   ```

2. Issue this command on the system that is to be the repository host OracleAS Instance for the OracleAS File-based Farm:

   ```
   dcmctl importRepository -file file name
   ```

   In the preceding command, *file name* is the name of the file you specified in the `exportRepository` command.

3. On the system that was the repository host, indicate that the OracleAS Instance is no longer the host OracleAS Instance by issuing the following command:

   ```
   dcmctl repositoryRelocated
   ```

### 2.2.8 Moving an Instance From One OracleAS File-based Farm to Another

You can move an OracleAS Instance that belongs to one OracleAS File-based Farm to another OracleAS File-based Farm. To move to another repository, you use the `leaveFarm` and `joinFarm` commands.

When an OracleAS Instance leaves an OracleAS File-based Farm, it becomes a standalone OracleAS Instance. This means that:

- The OracleAS Instance’s DCM-managed configuration metadata in the repository is moved to the OracleAS Instance’s local, file-based repository.
- Archives of the OracleAS Instance remain in the farm; they are not moved to the new farm.
- If there are connections to the Infrastructure database for other components (Oracle Application Server Single Sign-On, Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider, and Oracle Internet Directory) they remain.

Follow these steps to move an OracleAS Instance from one repository to another:

1. Issue the following command in the OracleAS Instance that you will move to another OracleAS File-based Farm:

   ```
   dcmctl whichFarm
   ```

   If `dcmctl` returns an OracleAS File-based Farm name, the OracleAS Instance is still part of a OracleAS File-based Farm, and is thus still associated with an existing repository.

2. Issue this command:

   ```
   dcmctl leaveFarm
   ```

   The OracleAS Instance is now in a standalone state.
3. Use the `createArchive` command to create a baseline archive of the OracleAS Instance. For example:

   ```
dcmctl createArchive -arch myArchive
   ```

4. Issue the following command in one of the OracleAS Instances that belongs to the OracleAS File-based Farm whose repository the OracleAS Instance will join:

   ```
dcmctl getRepositoryId
   ```

dcmctl returns the repository identifier in the format `hostname:port`.

5. Join the OracleAS Instance to the OracleAS File-based Farm of the repository identified in Step 4 by issuing this command:

   ```
dcmctl joinFarm -r repository ID
   ```

---

**Note:** The change will not be reflected in the Application Server Control Console until it is restarted. In the `ORACLE_HOME/bin` directory, issue these commands:

   ```
   ORACLE_HOME/bin> emctl stop iasconsole
   ORACLE_HOME/bin> emctl start iasconsole
   ```

---

### 2.2.9 Moving An Instance to a OracleAS Database-based Farm

You can use the `dcmctl` utility to move an OracleAS instance that is joined to a OracleAS File-based Farm to a OracleAS Database-based Farm.

To move an instance from a OracleAS File-based Farm to an OracleAS Database-based Farm, first disassociate the instance from its current repository by leaving the OracleAS File-based Farm. The instance is then in a standalone state, and you can join the instance to the OracleAS Database-based Farm.

Follow these steps to move the instance:

1. Determine whether the instance is still part of an OracleAS File-based Farm using the following command:

   ```
dcmctl whichFarm
   ```

2. If the output indicates that the instance is part of a OracleAS Database-based Farm, issue the following command to put the instance in a standalone state:

   ```
dcmctl leaveFarm
   ```

3. If you want the instance to join a OracleAS Database-based Farm, and the instance was previously a member of that OracleAS Database-based Farm, use the following command to rejoin the OracleAS Farm:

   ```
dcmctl joinFarm
   ```

   If this step fails, then the instance was not previously a member of the OracleAS Database-based Farm. Therefore, its configuration metadata is not in the DCM repository, which means that you will need to use Application Server Control Console to join the OracleAS Database-based Farm.
2.2.10 Enabling SSL For Communication Between Instances in an OracleAS File-based Farm

You can configure DCM to send configuration information between instances using the Secure Sockets Layer (SSL) protocol. This feature ensures the security of messages sent between all instances in the OracleAS File-based Farm, and prevents unauthorized instances from joining the OracleAS File-based Farm.

This section provides instructions for setting up SSL and certificate-based security for instances in a OracleAS File-based Farm. The complete procedure consists of the following tasks:

1. Generating the Keystore
2. Stopping the Oracle Application Server Processes on Each Instance
3. Setting Up Keystore Information File on Each Instance in the OracleAS File-based Farm
4. Enabling SSL in the dcmCache.xml File
5. Verifying that Configuration Changes are Effected
6. Starting the Instances in the OracleAS File-based Farm
7. Adding a New Instance to a SSL-Enabled OracleAS File-based Farm

### 2.2.10.1 Generating the Keystore

Use the JDK `keytool` command to generate a certificate and create the keystore, as documented in:

http://java.sun.com/j2se/1.4.1/docs/tooldocs/solaris/keytool.html

If you have already generated the key pair and obtained the certificate for OC4J, then you can use the same keystore you previously obtained.

To use SSL certificate-based security, a Java keystore must be created on each instance in the OracleAS File-based Farm. This keystore may be the same as that used by other Java applications, or it can be unique for OracleAS File-based Farm configuration. Note the path to each keystore location for each instance in the OracleAS File-based Farm.
2.2.10.2 Stopping the Oracle Application Server Processes on Each Instance

In each instance of the OracleAS File-based Farm, issue the following commands to stop OracleAS processes:

```
ORACLE_HOME/bin/emctl stop iasconsole
ORACLE_HOME/dcm/bin/opmnctl stopall
```

2.2.10.3 Setting Up Keystore Information File on Each Instance in the OracleAS File-based Farm

After obtaining the keystore and certificate information, you must create the file that holds the keystore information on each OracleAS instance in the OracleAS File-based Farm.

---

**Important:** The keystore information file must be set up for the repository host instance OracleAS File-based Farm before it is set up in any other instance. To find the repository host and host instance, execute the following:

```
dcmctl whichfarm -v
```

To set up the keystore information file, perform these steps, beginning with the repository host instance OracleAS File-based Farm. The steps can be performed on the remaining instances in any order.

1. Copy the keystore file that you generated in the first step, “Generating the Keystore,” to a location in the local host.
2. Use the `configRepositorySSL` command as follows:

```
dcmctl configRepositorySSL -keystore path_to_keystore -storepass password
```

The generated file, `.ssl.conf`, is stored in the `ORACLE_HOME/dcm/config` directory.

2.2.10.4 Enabling SSL in the `dcmCache.xml` File

To enable or disable the use of SSL, modify the `<useSSL>` attribute in the `ORACLE_HOME/dcm/config/dcmCache.xml` file. Table 2–4 lists the configuration parameters for enabling SSL.

Optionally, you can specify the location of the file that was generated using `configRepositorySSL` by modifying the value of the `<sslConfigFile>` element. If you modify this value, you need to copy the `.ssl.conf` file that `configRepositorySSL` generated to the new file that you specify using the `<sslConfigFile>` element.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;useSSL&gt;</code> true</td>
<td>Set to <code>true</code> to use SSL or to <code>false</code> to disable the use of SSL for DCM communications. The default value is <code>false</code>. Valid values: <code>true</code>, <code>false</code></td>
</tr>
</tbody>
</table>
2.2.10.5 Verifying that Configuration Changes are Effected

Ensure that the configuration changes are effected by executing the following command on each instance in the OracleAS File-based Farm, beginning with the repository host instance:

dcmctl getstate

This shows the synchronization state of the local instance in the OracleAS File-based Farm.

2.2.10.6 Starting the Instances in the OracleAS File-based Farm

After the security configuration is consistent across all the instances in the OracleAS File-based Farm, restart each instance, beginning with the repository host instance, using the following command:

```
ORACLE_HOME/opmn/bin/opmnctl startall
ORACLE_HOME/bin/emctl start iasconsole
```

2.2.10.7 Adding a New Instance to a SSL-Enabled OracleAS File-based Farm

You can add a standalone instance to a OracleAS File-based Farm that is using SSL. On the standalone machine:

1. Copy the keystore file that you generated in the first step, "Generating the Keystore," to a location on the local computer.

2. Use the configRepositorySSL as follows on each instance to create the keystore information file:

```
dcmctl configRepositorySSL -keystore path_to_keystore -storepass password
```

The generated file, .ssl.conf, is stored in ORACLE_HOME/dcm/config.

3. Follow the instructions in Section 2.2.2, "Adding Instances to an OracleAS Farm" on page 2-4 to join the instance to the farm.
2.2.11 Improving Performance in a Large OracleAS File-based Farm

You can improve the performance of a large OracleAS File-based Farm by tuning parameters in the in ORACLE_HOME/dcm/config/dcmCache.xml file. Table 2–5 lists the parameters and the sample values that were used to improve the performance of an OracleAS File-based Farm with 12 OracleAS Instances.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Adjusted Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;clean-interval&gt;</td>
<td>180</td>
<td>540</td>
<td>As the size of the farm increases, there are more cached objects. The process of cleaning the cache entails removing references to invalidated objects at a given interval. The more cached (and invalidated) objects, the longer it takes to clean the cache. With an increased number of objects, the cleanup interval should be set higher to decrease the frequency of cache cleanup, which is CPU-intensive.</td>
</tr>
<tr>
<td>&lt;max-objects&gt;</td>
<td>2000</td>
<td>6000</td>
<td>If the log file contains CacheFullException or RegionFullException errors, increase the value of max-objects.</td>
</tr>
<tr>
<td>&lt;disksize&gt;</td>
<td>256</td>
<td>768</td>
<td>The disksize parameter should be set to a value larger than the sizes of the J2EE deployment EAR files.</td>
</tr>
</tbody>
</table>

After you change ORACLE_HOME/dcm/config/dcmCache.xml, you must restart:

- The dcm daemon, by issuing these commands in ORACLE_HOME/opmn/bin:
  
  ```
  opmnctl stopproc ias-component=dcm-daemon
  opmnctl startproc ias-component=dcm-daemon
  ```

- The dcmctl shell, by issuing these commands in ORACLE_HOME/dcm/bin:
  
  ```
  dcmctl shell
  exit
  ```

- The Enterprise Manager daemon, by issuing these commands in ORACLE_HOME/bin:
  
  ```
  emctl stop iasconsole
  emctl start iasconsole
  ```

2.2.12 Removing an Instance From the Infrastructure After Executing the leaveFarm Command

The leaveFarm command removes an OracleAS Instance from an OracleAS Farm, but in an OracleAS Database-based Farm, the OracleAS Instance remains in the Identity Management configuration. You may choose to remove the OracleAS Instance from the Identity Management configuration.

The configuration files and entries that must be changed depend on the installation type of the OracleAS Instance that was removed from the OracleAS Database-based Farm. Follow the steps in the section for the applicable installation type. After the steps are performed, the OracleAS Instance is a standalone OracleAS Instance.
2.2.12.1 J2EE and Web Cache Instance Using Identity Management (Oracle Internet Directory)

Follow these steps to remove the OracleAS Instance from the Identity Management configuration:

1. Use the `oidadmin` tool to remove the entries created during installation to register the OracleAS Instance with Oracle Internet Directory.

   One of the entries to be removed is shown in bold italic:

   "orclApplicationCommonName=instance name, cn=IAS Instances, cn=IAS, cn=Products, cn=OracleContext

   The other entry is shown in bold italic (in the uniquename attribute of the following entry):

   "cn=Associated Mid-tiers, OrclReferenceName=repository name, cm=IAS Infrastructure Databases, cn=Products, cn=OracleContext

2. Edit the `ORACLE_HOME/config/ias.properties` file to remove the Oracle Internet Directory host, port, dbname, and instance password. Change the `UseInfrastructure` flag to false.

3. Change the OracleAS JAAS Provider entries, editing the `ORACLE_HOME/j2ee/home/config/jazn.xml` file to replace the following entry:

   orclApplicationCommonName=jaznadmin1,cn=JAZNContext,cn=products,cn=OracleContext

   with this entry:

   ```xml
   <jazn provider="XML" location="/jazn-data.xml">
   </jazn>
   ```

   Remove the corresponding Oracle Internet Directory entry also.

4. Modify the LDAP configuration as follows:
   a. Remove the `ORACLE_HOME/network/admin/ldap.ora` file.
   b. Edit the `ORACLE_HOME/network/admin/sqlnet.ora` file to remove the LDAP naming from the list of namings.

2.2.12.2 J2EE and Web Cache Installation with Database-based Repository

To remove the OracleAS Instance from the Identity Management configuration of a J2EE and Web Cache Installation:

Edit the `ORACLE_HOME/config/iasschema.xml` file to change the DCM repository association entry as follows:

from this entry:

```xml
<SchemaConfigData>
  <ComponentName>DCM</ComponentName>
  <BaseName>DCM</BaseName>
  <Override>true</Override>
  <SchemaName>DCM</SchemaName>
  <DBConnect>iassdb.us.oracle.com</DBConnect>
  <Password></Password>
  <SvcPassword></SvcPassword>
</SchemaConfigData>
```
to the following entry:

```xml
<SchemaConfigData>
  <ComponentName>DCM</ComponentName>
  <BaseName>DCM</BaseName>
  <Override>false</Override>
  <SchemaName/>
  <DBConnect/DBConnect>
  <Password/>
  <SvcPassword/>
</SchemaConfigData>
```

### 2.2.12.3 J2EE and Web Cache Installation Using Identity Management (Oracle Internet Directory) with Database-based Repository

Follow these steps to remove the OracleAS Instance from the Identity Management configuration:

1. Use the `oidadmin` tool to remove the entries created during installation to register the OracleAS with Oracle Internet Directory.

   One of the entries to be removed is shown in bold italic:
   
   ```
   "orclApplicationCommonName=instance name, cn=IAS Instances, cn=IAS, cn=Products, cn=OracleContext
   ```
   
   The other entry is shown in bold italic (in the uniquename attribute of the entry):
   
   ```
   "cn=Associated Mid-tiers, OrclReferenceName=repository name, cm=IAS Infrastructure Databases, cn=Products, cn=OracleContext
   ```

2. Edit the `ORACLE_HOME/config/ias.properties` file to remove the Oracle Internet Directory host, port, dbname, and instance password. Change the `UseInfrastructure` flag to `false`.

3. Edit the `ORACLE_HOME/config/iasschema.xml` file to change the DCM repository association entry as follows:

   from this entry:
   
   ```xml
   <SchemaConfigData>
   <ComponentName>DCM</ComponentName>
   <BaseName>DCM</BaseName>
   <Override>true</Override>
   <SchemaName>DCM</SchemaName>
   <DBConnect>iasdb.us.oracle.com</DBConnect>
   <Password></Password>
   <SvcPassword></SvcPassword>
   </SchemaConfigData>
   ```

   to the following entry:

   ```xml
   <SchemaConfigData>
   <ComponentName>DCM</ComponentName>
   <BaseName>DCM</BaseName>
   <Override>false</Override>
   <SchemaName/>
   <DBConnect/DBConnect>
   <Password/>
   <SvcPassword/>
   </SchemaConfigData>
   ```
4. Change the OracleAS JAAS Provider entries, editing the `ORACLE_HOME/j2ee/home/config/jazn.xml` file to replace the following entry:

```xml
<jazn provider="XML" location="./jazn-data.xml"/>
```

with this entry:

```xml
<?xml version="1.0" encoding="UTF-8" standalone='yes'?>
<!DOCTYPE jazn PUBLIC "JAZN Config" "http://xmlns.oracle.com/ias/dtds/jazn-9_04.dtd">
<jazn provider="XML" location="/jazn-data.xml"/>
</jazn>
```

Remove the corresponding Oracle Internet Directory entry also.

5. Modify the LDAP configuration as follows:
   a. Remove the `ORACLE_HOME/network/admin/ldap.ora` file.
   b. Edit the `ORACLE_HOME/network/admin/sqlnet.ora` file to remove the LDAP naming from the list of namings.

### 2.3 Creating and Maintaining OC4J Instances In DCM-Managed OracleAS Clusters

The `dcmctl` utility includes commands for creating and deleting OC4J instances in OracleAS Cluster. It also contains other commands for OC4J application deployment (`validateEarFile`, `deployApplication` and `redployApplication`), but it is generally recommended that in production environments, you use the Oracle Enterprise Manager 10g Application Server Control Console user interface for application deployment. The Application Server Control Console provides configuration change, monitoring, and status functionality as part of the deployment process. For example, the Application Server Control Console will notify you with a message that certain configuration changes will only take effect after you restart the OC4J instance.

#### 2.3.1 Creating an OC4J Instance

Follow these steps to create an OC4J instance:

1. Navigate to the OracleAS Instance that will contain the new OC4J instance.
2. Issue this command:

```bash
dcmctl createComponent -ct oc4j -co nameOfOC4JInstance
```

In the preceding command, `nameOfOC4JInstance` is the identifier for the new OC4J instance. Do not use the host name, the Oracle home or IP address of the computer containing the Oracle Application Server installation as the name for the OC4J instance.

**Tip:** See `createComponent` for syntax and rules, or issue the command `dcmctl help createcomponent` to view help text.
2.3.2 Deleting an OC4J Instance

You can use the `dcmctl` utility to remove an OC4J instance from the local application server instance, if the OC4J instance was created after installation by a user. You cannot remove any OC4J instance that was created by the Oracle Application Server installation process (such as `OC4J_SECURITY`). Follow these steps to remove an OC4J instance:

1. Navigate to the OracleAS Instance that contains the OC4J instance you want to delete.
2. Issue this command:

```
dcmctl removeComponent -co nameOfOC4JInstance
```

In the preceding command, `nameOfOC4JInstance` is the identifier for the new OC4J instance.

**Tip:** See `removeComponent` for syntax and rules, or issue the command `dcmctl help removecomponent` to view help text.

2.4 Managing DCM Log Files

DCM creates log files to track activity and errors that occurred in the operation of the `dcmctl` utility, the DCM daemon, the Enterprise Manager daemon, and the business rule function. You can specify the size of the individual log files and the overall size of the logs by setting the `maxFileSize` and `maxLogSize` parameters.

2.4.1 Log File Location and Naming

The log files are created in subdirectories of the `ORACLE_HOME/dcm/logs` directory. The subdirectories are named for the DCM component or function: `dcmctl_logs`, `daemon_logs`, `busrule_logs`, `emd_logs`. The log files in each directory are named as follows:

```
log.xml, log1.xml, log2.xml ... logN.xml
```

The `log.xml` file is the most recent log file; `log1.xml` is the oldest. When the `log.xml` file size reaches approximately the number of bytes specified by the `maxFileSize` parameter, the `log.xml` file is renamed to `log1-N.xml`, and a new `log.xml` file is created.

**Note:** The actual number of bytes in a log file is not always exactly equal to the `maxFileSize` value. Log file sizes may be slightly more or less when rotation occurs. Individual log entries are not split across files.

The `maxLogSize` parameter specifies the maximum number of bytes of the combined files (`log.xml` through `logN.xml`) and determines when the oldest log file is removed. The actual log size may be slightly larger or smaller than the specified `maxLogSize` when rotation occurs.

2.4.2 Changing Log Size Parameters

Follow these steps to change the `maxFileSize` or `maxLogSize` parameters:

1. Ensure that the DCM daemon process and DCM client processes such as `dcmctl` and Oracle Enterprise Manager 10g Application Server Control Console are stopped.

2. Edit the `ORACLE_HOME/dcm/config/sysmgmtProperties.xml` file and change the `maxFileSize` and `maxLogSize` parameter values to the sizes you want. Example 2–1 shows the parameters as they appear in the file.

   ![Note: An error will occur if the value of `maxFileSize` is larger than the value of `maxLogSize`.](image)

   **Example 2–1 sysmgmtProperties.xml File Contents**

   ```xml
   <Section>
   <Component name="DMSLogging">
   <Property name="maxFileSize" value="200000"/>
   <Property name="maxLogSize" value="1000000"/>
   </Component>
   </Section>
   ```

3. Use these commands to restart:

   - The dcm daemon, by issuing these commands in `ORACLE_HOME/opmn/bin`:
     
     ```bash
     opmnctl stopproc ias-component=dcm-daemon
     opmnctl startproc ias-component=dcm-daemon
     ```

   - The dcmctl shell, by issuing these commands in `ORACLE_HOME/dcm/bin`:
     
     ```bash
     dcmctl shell
     exit
     ```

   - The Enterprise Manager daemon, by issuing these commands in `ORACLE_HOME/bin`:
     
     ```bash
     emctl stop iasconsole
     emctl start iasconsole
     ```
2.5 Best Practices for DCM-Managed OracleAS Cluster Management

Understanding and adhering to the recommendations outlined will help you maximize DCM’s capabilities and safeguard your system configuration. Note the following OracleAS Cluster management practices:

■ OracleAS Instances grouped in a DCM-Managed OracleAS Cluster can be managed using Oracle Enterprise Manager 10g Application Server Control Console or `dcmctl` from a single point of administration on any OracleAS Instance in the DCM-Managed OracleAS Cluster. It is advisable to use one OracleAS Instance as the administrative locus for the entire DCM-Managed OracleAS Cluster.

■ When changing instance-specific configuration such as port numbers, host names or virtual hosts, ensure that no other concurrent administrative changes are made in the DCM-Managed OracleAS Cluster. This should help prevent application of conflicting changes to configuration (which could result in an unusable configuration).

■ Concurrent administration within an DCM-Managed OracleAS Cluster is strongly discouraged. If multiple administrative operations are issued at the same time in an DCM-Managed OracleAS Cluster, errors may occur, and the associated error messages may be confusing. For example, a concurrent attempt to change the configuration of an OracleAS Instance that is being deleted really does not make sense. It is recommended that one OracleAS Instance be used as the administrative locus for the entire DCM-Managed OracleAS Cluster. This will ensure that operations are executed in logical order and are properly serialized.

■ Archiving (automatically or manually) before any administrative operation is recommended. With auto-archiving, you can have `dcmctl` automatically detect that the OracleAS Instance is associated with an DCM-Managed OracleAS Cluster and create auto-archives of the DCM-Managed OracleAS Cluster configuration before administrative operations are performed. Table 3–1, “Automatic Archive Operations” on page 3–4 lists the operations that trigger automatic archiving.
This chapter describes DCM archiving capabilities, and explains how to restore saved configurations from archives.

This chapter covers the following topics:

- DCM Archiving Commands
- Initial Archive Creation
- Archive and Instance Compatibility
- Exporting and Importing Archives
- Using Automatic Archiving
- Preserving Configurations With Archive Commands
- Importing and Applying an Archive to an Instance

3.1 DCM Archiving Commands

The `dcmctl` utility provides commands (listed in Table A–6, "Archive Commands") that enable you to create an archive of the configuration of an Oracle Application Server Instance or OracleAS Cluster, and then apply the archived configuration to the same Oracle Application Server Instance or OracleAS Cluster, or to a different Oracle Application Server Instance or OracleAS Cluster. The archiving feature makes it easy to save configurations before making changes to the system, or to save and restore a particular configuration for specific purposes, such as operating one configuration during the day and another at night.

**Note:** The `saveInstance` and `restoreInstance` commands are deprecated; the archiving feature contains all of the functionality provided by `saveInstance` and `restoreInstance`.

3.2 Initial Archive Creation

The Oracle Universal Installer invokes the archiving function at the end of installation, and archives the initial configuration of Distributed Configuration Management. The name of the initial archive is:

`InstalledImageinstance_name`

In the example above, `name` is the name of the Oracle Application Server Instance.
3.3 Archive and Instance Compatibility

When an archive is created, the configuration and application deployment information associated with the archived object (the Oracle Application Server Instance or OracleAS Cluster) is stored in the repository. This archived image can then be applied to any compatible Oracle Application Server Instance or OracleAS Cluster in the repository, or exported to a file to be applied to an Oracle Application Server Instance or OracleAS Cluster in another repository. The compatibility of an archive with an Oracle Application Server Instance or OracleAS Cluster is similar to the compatibility of Oracle Application Server Instances to be clustered.

Applying archives to Oracle Application Server Instances and OracleAS Clusters is subject to these compatibility rules:

- The source of the archive must have the same installed components configured as the destination. If the destination is an OracleAS Cluster, the archive cannot contain any non-clusterable objects.
- The source of the archive and the instance or cluster to which it is applied must be of the same installation type and version.
- Information specific to an Oracle Application Server Instance, such as host name, can only be applied back to the Oracle Application Server Instance from which it came.
- The configuration from one Oracle Application Server Instance or OracleAS Cluster may be applied directly to another Oracle Application Server Instance or OracleAS Cluster. That is, the apply operation can take, as a source, an archive of an Oracle Application Server Instance or an OracleAS Cluster.

See Also: isClusterable on page A-21

3.4 Exporting and Importing Archives

You can export an archive from the repository to a file, and then import the file back to the same repository, or to a different repository. You can change the name of the archive and associated comments during the import. The original archive name and comments are the defaults.

You can also export from a repository to a file, and import from a file to a repository. The import and export functionality allows an archive to be moved from one repository to another. Archives can be moved from:

- A Database-based repository to another Database-based repository
- A File-based repository to another File-based repository
- A Database-based repository to a File-based repository
- A File-based repository to a Database-based repository

The exported archive file is in a .jar format. Two entries in the jar file encapsulate a description of the archive: Export_Information and Archive_Information. These entries can be extracted from the export file and viewed as text. The other entries in the export file contain the configuration information, one entry for each
component instance. Example 3–1, Example 3–2, and Example 3–3 provide sample archive files.

In Oracle Application Server 10g Release 2 (10.1.2), the files stored in the DCM Repository are not the actual configuration files, but are an internal form with information such as the ORACLE_HOME, host name, and IP address removed.

The current archive files are binary. They are JAR files (ZIP files with additional metadata) that could be stored in a version control system and the used later to apply a prior version of configuration. Considering the configuration data stored within the archive is an internal Oracle form, it makes sense to keep archive files in binary form, rather than exploding the JAR and storing individual files.

---

**Caution:** Do not edit the archive files. If you do, the archive may not function as it should.

---

**Example 3–1  Expanded Export file**

```
Archive_Information
Export_Information
Default_Information
HTTP Server
home
Component1
Component2
```

**Example 3–2  Export_Information File**

```
Export_Information:
Version: 10.1.2.0.0
Comments: "User comments about the export"
Farm: dbmachine.1234.5678
Create Time: Mon Sep 30 12:42:10 PDT 2002
```

**Example 3–3  Archive_Information File**

```
Archive_Information:
Name: archive1
Source: cluster1
Version: 10.1.2.0.0
Configured Component Types: [Apache, J2EE]
Comments: "User comments about the archive"
Create Time: 2002-09-26 16:46:40.0
Clusterable:true
```

3.5 Using Automatic Archiving

When you turn automatic archiving on, DCM automatically creates an archive when you perform any operation listed in Table 3–1. The automatic archives coexist with archives created with the `createArchive` command, and are distinguished by their system-generated name.

Example 3–6 shows several automatically generated archives.
You can specify the number of automatic archive versions you want to save, or disable automatic archiving with the `set` command, using the `-arch` option and an integer, as shown in Example 3–4 and Example 3–5.

**Example 3–4  Setting Automatic Archiving to Save 10 Versions**

```plaintext
dcmctl set -arch 10
```

**Example 3–5  Turning Automatic Archiving Off**

```plaintext
dcmctl set -arch 0
```

Automatic archives have system-generated names that resemble those shown in archive 2 in Example 3–6. Automatic archive names all begin with `dcm.autoarchive`. Appended to this is the IP address of the computer, an identifier for the DCM operation that triggered the archive, and a user ID that is unique to the computer. This ensures that the automatic archive name is unique across the Oracle Application Server Farm. When an Oracle Application Server Instance is a member of an OracleAS Cluster, the Source: identifier is shown as: “cluster: cluster name”.

**Example 3–6  Automatic versus User Generated Archive Names (listArchives Command Output)**

```plaintext
dcmctl listArchives
****** AUTO GENERATED ARCHIVES ******
1
Name:     dcm.autoarchive_138.2.142.2121ff7ae.fcb87f2e6d.-7ffe
Source:   instance: my-sun.us.oracle.com
Version:  10.1.2.0.0
Comments: Automatic archival prior to deployment of application IsWebCacheWkng
Clusterable: true

2
Name:     dcm.autoarchive_138.2.142.21212d3205.fcb88f2b5b.-7fff
Source:   instance: my-sun.us.oracle.com
```

<table>
<thead>
<tr>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add an Oracle Application Server Instance to an OracleAS Cluster</td>
</tr>
<tr>
<td>Change an OracleAS Cluster's attributes</td>
</tr>
<tr>
<td>Change an Oracle Application Server Instance's attributes</td>
</tr>
<tr>
<td>Configure Oracle Application Server Single Sign-On with configuration tool</td>
</tr>
<tr>
<td>Create a new OC4J instance</td>
</tr>
<tr>
<td>Deploy a J2EE application</td>
</tr>
<tr>
<td>Dump or clear instrumentation timer</td>
</tr>
<tr>
<td>Issue the updateConfig command</td>
</tr>
<tr>
<td>Join an OracleAS Cluster</td>
</tr>
<tr>
<td>Perform a configuration change with System Management Interface</td>
</tr>
<tr>
<td>Remove an OC4J instance</td>
</tr>
</tbody>
</table>

Table 3–1  **Automatic Archive Operations**
3.5.1 Tuning Automatic Archiving to Improve Performance

When you install Oracle Application Server, the auto-archiving feature is set to maintain fifteen archives. You can improve system performance by reducing the number of archives, or by turning off auto-archiving.

---

**Note:** Limiting or disabling DCM auto-archiving may reduce the potential for full recovery from a system failure.

---

3.6 Preserving Configurations With Archive Commands

You should save the DCM configuration regularly. You may find it useful to create "before" and "after" snapshots of a configuration when performing extensive configuration changes. Archives that you create are designated as user-generated archives, to differentiate them from archives created automatically (shown in Example 3–6). Use the following steps to create an archive:

1. Issue one of the following commands (depending on the configuration):
   ```
   dcmctl createarchive -arch archive name -cl cluster name
   
   or
   
   dcmctl createarchive -arch archive name -i instance name
   ```

   The archive is created in the repository.

2. (Optional) Export the archive to the file system with this command:
   ```
   dcmctl exportarchive -arch archive name -f file name
   ```

   **Note:** Archives are stored in the repository. If you do not export an archive to the file system, and the repository is destroyed, any archives saved in the repository are also lost. Exporting the archive to the file system provides an extra measure of safety.

---

3.7 Importing and Applying an Archive to an Instance

If an Oracle Application Server Instance is inadvertently destroyed or a repository contains errors, you can restore the configuration from an exported archive. Use the `importArchive` command to bring the archive from the file system to the repository:

```
 dcmctl importarchive -arch archive name -f file name
```
You can restore an archive to the DCM repository with one of the following commands:

dcmctl applyarchiveto -arch archive name -cl cluster name
dcmctl applyarchiveto -arch archive name -i instance name

### 3.8 Benefits of Archiving

The DCM archive feature provides a convenient way to create snapshots of the DCM-managed portions of Oracle Application Server system configuration. Archives are useful for staging changes, recovering from errors, and provisioning DCM-managed Oracle Application Server Instance configuration information from one Oracle Application Server Instance to another. Archiving enables you to:

- Use disk space efficiently. Within an Oracle Application Server Instance, there are many managed objects (including configuration files and EAR or WAR files), but when an archive is created, only one copy of any given version of a managed object is saved in the repository.

- Restore the state of an Oracle Application Server Instance or OracleAS Cluster to a prior state. The DCM system automatically takes an archive when it performs certain administrative operations so that the administrator has the option to roll back undesired administrative changes. The number of automatic archives that are saved is configurable.

- Explicitly create archives to satisfy a site’s change management or staging policy. For example, when staging groups of changes that an administrator may want to collectively rollback, or push to other Oracle Application Server Instances, it is a good idea to explicitly create an archive.

- Use archives to apply the state of any Oracle Application Server Instance to the state of any available archive. An archive can also be applied to another Oracle Application Server Instance in the same Oracle Application Server Farm, or exported and imported to another Oracle Application Server Farm and then applied to an Oracle Application Server Instance in that Oracle Application Server Farm. (A hybrid staging solution is to first stage and test changes to a non-clustered Oracle Application Server Instance, archive the changes, and finally apply the archive to an OracleAS Cluster. These changes are then automatically propagated to all members of the OracleAS Cluster.)

**Note:** After applying an archive to an Oracle Application Server Instance other than the Oracle Application Server Instance from which it was created, some instance-specific configuration data may have to be modified. DCM automates this for IP addresses and hostnames. See Section 2.2.4.2, "Parameters Excluded from the Common Configuration: Instance-Specific Parameters" on page 2-5 for lists of these parameters in OHS, OC4J, and OPMN.

- If you use OracleAS Clusters, DCM assures that any change to the configuration is automatically distributed to all members of the OracleAS Cluster. As an alternative to using OracleAS Clusters, an archive of a staged configuration can be applied manually to non-clustered Oracle Application Server Instances in an Oracle Application Server Farm.
This chapter describes the characteristics of the dcmctl utility, and provides syntax and reference information for each of its commands. It is divided into these sections:

- Notes for Using dcmctl Commands
- Types of dcmctl Commands
- Alphabetical Listing of dcmctl Commands

### A.1 Notes for Using dcmctl Commands

All dcmctl commands have this syntax:

```
ORACLE_HOME/dcm/bin/dcmctl command [options]
```

Before you use the dcmctl utility, be aware of the following:

- Do not run `updateConfig` concurrently with any other dcmctl commands, or perform Oracle Enterprise Manager 10g Application Server Control Console configuration operations from multiple Oracle Application Server Instances in an Oracle Application Server Farm or OracleAS Cluster (see `updateConfig` for details).

- Oracle Application Server supports heterogeneous Oracle Application Server Instances as part of the same OracleAS File-based Farm. For example, an Oracle Application Server Instance running on the Solaris operating system, an Oracle Application Server Instance running on a Linux system, and an Oracle Application Server Instance running on an HP-UX system can reside in the same OracleAS File-based Farm. Oracle Application Server Instances that you want to be part of an OracleAS Cluster must be of the same installation type and version, and be installed on like operating systems (for example, UNIX and Linux).

- To use `dcmctl`, you must log in to the operating system with the user name that was used to install Oracle Application Server, or as the administrative user.

- When administering an OracleAS File-based Farm, you may need to refresh the browser, or restart Application Server Control Console, after issuing the following dcmctl commands:
  - `joinCluster`
  - `joinFarm`
  - `leaveCluster`
  - `leaveFarm`
- Ensure that you issue `dcmctl` commands in the Oracle home of the Oracle Application Server Instance you wish to manage. `dcmctl` commands operate on the Oracle Application Server Instance in which the `dcmctl` executable is located. The value of the `ORACLE_HOME` environment variable does not determine the Oracle Application Server Instance on which `dcmctl` operates. You can issue commands to operate remotely (on a specified Oracle Application Server Instance) by using the `-i` option described in Table A–2.

- All `dcmctl` commands and options are case-insensitive.

- Instance, component, and OracleAS Cluster names are case-sensitive.

### A.1.1 `dcmctl` Options

As shown in the syntax description, `dcmctl` commands may be used with options. An option can be one of following types:

- **Global**: These can be used with all commands (see Table A–1).

- **Scope**: These options indicate the scope of a command. The scope can be an application, OracleAS Cluster, component, type of component, or Oracle Application Server Instance (see Table A–2).

- **Command-specific**: These options apply only to certain commands. Each command description in the following pages includes information about any options the command provides (see Table A–3).

#### Table A–1 `dcmctl` Global Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-d</code></td>
<td>Prints the stack trace if an exception occurs when the command is executed. By default, <code>dcmctl</code> executes with this option on. You can change this behavior with the <code>set</code> command.</td>
</tr>
<tr>
<td><code>-l directory</code></td>
<td>Saves the DCM client error log file <code>log.xml</code> in the named <code>directory</code>. The directory can be a full path name or a path name relative to the current directory. Default value: <code>ORACLE_HOME/dcm/logs/dcmctl_logs</code></td>
</tr>
<tr>
<td><code>-v</code></td>
<td>Prints the long (verbose) version of state and error messages. Setting verbose to <code>off</code> is recommended when using scripts or using the <code>dcmctl</code> shell, since the brief messages are easier to parse. By default, <code>dcmctl</code> executes with this option on. You can change this behavior with the <code>set</code> command.</td>
</tr>
<tr>
<td><code>-sort</code></td>
<td>Sorts the output of list commands by name. For example: <code>dcmctl listComponents -sort</code> By default, this option is set to <code>off</code>. You can make the sorting feature persistent with the <code>set</code> command.</td>
</tr>
</tbody>
</table>
A.1.1.1 Notes on OracleAS Cluster, Instance, Component and Component Type Options

The following rules apply to command-specific options. If you do not use options, by default, the command applies to the local Oracle Application Server Instance.

- If `-cl` is supplied, the command applies to all Oracle Application Server Instances in the OracleAS Cluster.
- If `-ct` is supplied with `-cl` or `-i`, then the command applies to the component type within the OracleAS Cluster, or the component type within the Oracle Application Server Instance within the OracleAS Cluster.
- If `-co` is supplied with `-ct`, `-cl`, `-i` then the command applies to the component within the OracleAS Cluster, or the component type within the Oracle Application Server Instance within the OracleAS Cluster.
- If `-a` is supplied with any of the preceding options, that particular application within the component, component type, Oracle Application Server Instance, or OracleAS Cluster is affected.
A.1.2 Using the dcmctl Shell

You can execute dcmctl commands from within the dcmctl shell. Within the shell, it is not necessary to preface commands with dcmctl (see the following sample session). To start the dcmctl shell, type:

```
dcmctl shell
```

Following is a sample shell session in which the shell is started, commands are executed, and the shell is stopped.

**Example A–1 dcmctl Shell Session**

```
dcmctl shell
dcmctl> createcluster -cl testcluster
dcmctl> joincluster -cl testcluster
dcmctl> createcomponent -ct oc4j -co component1
dcmctl> deployapplication -f /stage/apps/app1.ear -a app1 -co component1
dcmctl> getstate
dcmctl> exit
```

A.2 Types of dcmctl Commands

This section describes types of dcmctl commands and their uses.

Configuration Management: Use these commands to create and manage configuration elements (OracleAS Clusters, Oracle Application Server Instances, components) and the DCM repository.

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>applyClusterTo</td>
<td>applyInstanceTo</td>
<td>createCluster</td>
<td>createComponent</td>
</tr>
<tr>
<td>configRepositorySSL</td>
<td>destroyInstance</td>
<td>exportRepository</td>
<td>fileCleanUp</td>
</tr>
<tr>
<td>getComponentType</td>
<td>getRepositoryId</td>
<td>importRepository</td>
<td>isClusterable</td>
</tr>
<tr>
<td>isCompatible</td>
<td>joinCluster</td>
<td>joinFarm</td>
<td>leaveCluster</td>
</tr>
<tr>
<td>leaveFarm</td>
<td>listClusters</td>
<td>listComponents</td>
<td>listInstances</td>
</tr>
<tr>
<td>removeCluster</td>
<td>removeComponent</td>
<td>repositoryRelocated</td>
<td>resetDCMCachePort</td>
</tr>
<tr>
<td>resetFileTransaction</td>
<td>resetHostInformation</td>
<td>restoreInstance</td>
<td>resyncInstance</td>
</tr>
<tr>
<td>saveInstance</td>
<td>updateConfig</td>
<td>whichCluster</td>
<td>whichFarm</td>
</tr>
<tr>
<td>whichInstance</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**dcmctl Shell:** Use these commands with the dcmctl shell.

<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!</td>
<td>echo</td>
</tr>
<tr>
<td>shell</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td>quit</td>
</tr>
<tr>
<td></td>
<td>setLogLevel</td>
</tr>
</tbody>
</table>

Archive: Use these commands to create archives of configurations.
A.3 Alphabetical Listing of dcmctl Commands

This section lists all commands alphabetically. The command type, syntax, and description is included for each command.

Note: All commands are case-insensitive. Capital letters are used in this guide only to increase readability. Many examples show the commands in all lower case.
A.3.1 !!

Repeats the previous command.

Type
Shell

Syntax
!!

Description
Use this command in the dcmctl shell to repeat the previous command.

Example
!!

A.3.2 addOPMNLink

Creates a Manually-Managed Oracle Application Server Cluster.

Type
Manually-Managed OracleAS Cluster

Syntax
addOPMNLink hostname:port[, hostname:port...]

Description
You can use this command to create a Manually-Managed OracleAS Cluster that includes the local Oracle Application Server Instance and the Oracle Application Server Instances specified as arguments.

- You must run this command in the Oracle home of each Oracle Application Server Instance that is to be a member of the Manually-Managed OracleAS Cluster, using the rest of the Oracle Application Server Instances as arguments.
- All Oracle Application Server Instances must be J2EE and Web Cache Oracle Application Server Instances, and the Oracle Application Server Instances must not be part of an OracleAS File-based Farm. Otherwise, the command will fail.
- To change the ONS remote port for an Oracle Application Server Instance in a Manually-Managed OracleAS Cluster, you must remove the Oracle Application Server Instance from the Manually-Managed OracleAS Cluster using removeOPMNLink, change the remote port, and add it to the Manually-Managed OracleAS Cluster again using addOPMNLink. You must repeat the command in every Oracle home.
- If you create a Manually-Managed OracleAS Cluster and then want to add another Oracle Application Server Instance to the Manually-Managed OracleAS Cluster, you must run the command again in all Oracle homes (essentially creating a new Manually-Managed OracleAS Cluster with the added Oracle Application Server Instance).
Example

HOST1_ORACLE_HOME/dcm/bin/dcmctl getopmnport
host1:6200
HOST2_ORACLE_HOME/dcm/bin/dcmctl getopmnport
host2:6200
HOST3_ORACLE_HOME/dcm/bin/dcmctl getopmnport
host3:6200
HOST1_ORACLE_HOME/dcm/bin/dcmctl addopmnlink host2:6200,host3:6200
HOST2_ORACLE_HOME/dcm/bin/dcmctl addopmnlink host1:6200,host3:6200
HOST3_ORACLE_HOME/dcm/bin/dcmctl addopmnlink host1:6200,host2:6200

A.3.3 applyArchiveTo
Applies an archived configuration to an Oracle Application Server Instance or DCM-Managed OracleAS Cluster.

Type
Archive

Syntax
applyArchiveTo –src archiveName [-cl clusterName | -i instanceName]

Description
When configuration information is stored in the DCM repository, it is recognized as one of the following two types of information:

- Information that is generic to any Oracle Application Server Instance (cluster-wide information).
- Information that is specific to a particular Oracle Application Server Instance (instance-specific information). Instance specific information is defined by the various managed components, including: Oracle HTTP Server, OC4J, OPMN, and JAZN and may include such things as host name or port values.

When an archive is applied to the same Oracle Application Server Instance from which it was created, both the cluster-wide information and instance-specific information is restored to the Oracle Application Server Instance. When the archive is applied to a DCM-Managed OracleAS Cluster or to a different Oracle Application Server Instance than the one it was created for, only the cluster-wide information is restored, the existing instance-specific information is not changed. See Section 2.2.4.2, "Parameters Excluded from the Common Configuration: Instance-Specific Parameters" on page 2-5.

Example
dcmctl applyarchiveto -src archive1 -i instance1

A.3.4 applyClusterTo
Applies the configuration of a DCM-Managed OracleAS Cluster to another Oracle Application Server Instance or DCM-Managed OracleAS Cluster.

Type
Configuration Management
**Syntax**
applyClusterTo -src clusterName [-cl clusterName | -i instanceName]

**Description**
The configuration of the named DCM-Managed OracleAS Cluster (as specified by the -src option) is applied to the named Oracle Application Server Instance or DCM-Managed OracleAS Cluster. The named source DCM-Managed OracleAS Cluster is not affected.

**Example**
dcmctl applyclusterto –src cluster1 –i instance1

**A.3.5 applyInstanceTo**
Applies the configuration of the named Oracle Application Server Instance to another Oracle Application Server Instance or DCM-Managed OracleAS Cluster.

**Type**
Configuration Management

**Syntax**
applyInstanceTo –src instanceName [-cl clusterName | -i instanceName]

**Description**
The configuration of the named Oracle Application Server Instance is applied to the named Oracle Application Server Instance or DCM-Managed OracleAS Cluster. If no Oracle Application Server Instance or DCM-Managed OracleAS Cluster is specified, then the configuration of the named Oracle Application Server Instance is applied to the current Oracle Application Server Instance. The named source Oracle Application Server Instance is not affected. The command will fail if the current Oracle Application Server Instance and the named Oracle Application Server Instance are the same (you cannot apply the configuration of an Oracle Application Server Instance to itself).

**Example**
dcmctl applyinstanceto -src instance1

**A.3.6 configRepositorySSL**
Specifies the location and password of the keystore used to secure the Oracle Application Server Farm.

**Type**
Configuration Management

**Syntax**
cfgRepositorySSL -keystore pathToKeystore -storepass password

**Description**
Specifies the location and password of the keystore to use to provide certificate-based security for the farm. This command applies to a distributed File-based repository only. Configuring the keystore does not automatically enable security.
To use certificate-based security, each Oracle Application Server Instance in the Oracle Application Server Farm must have a Java keystore. It can be shared by other Java applications, or it can be a separate keystore specifically for repository administration. After the keystore is set up, use the configRepositorySSL command in each Oracle Application Server Instance in the Oracle Application Server Farm to tell the system which keystore to use.

To enable this security feature, edit the file ORACLE_HOME/dcm/config/dcmCache.xml, in each Oracle Application Server Instance in the Oracle Application Server Farm to set the value of <useSSL>true</useSSL>. You must restart all DCM daemons and clients in the Oracle Application Server Farm for the security change to take effect. The security setting must be consistent across all Oracle Application Server Instances in the Oracle Application Server Farm, or they will not communicate properly.

Example
configRepositorySSL -keystore /OracleHome/security/files -storepass welcome

A.3.7 createArchive

Creates an archive of the named DCM-Managed OracleAS Cluster or Oracle Application Server Instance.

Type
Archive

Syntax
createArchive -arch archiveName [-cl myCluster | -i myInstance] [-comment "myComments"]

Description
An archive is created of the named Oracle Application Server Instance or DCM-Managed OracleAS Cluster. If you don’t specify a DCM-Managed OracleAS Cluster or Oracle Application Server Instance, the current Oracle Application Server Instance is archived.

The difference is as follows.

Notes for using createArchive with DCM-Managed OracleAS Clusters:

- Cluster-wide archives, created with createArchive -cl, contain only cluster-specific information and do not contain any information specific to the Oracle Application Server Instance the archive is created on.

- Instance-specific archives, created with createArchive -i or with no options, contain cluster-specific information, plus any DCM-managed information pertaining to the Oracle Application Server Instance in which createArchive runs.

Example
createArchive -arch myInstance -comment "my favorite configuration"
A.3.8 createCluster

Creates a DCM-Managed OracleAS Cluster.

**Type**  
Configuration Management

**Syntax**  
createCluster -cl cluster_name

**Description**  
A DCM-Managed OracleAS Cluster is created. "DCM-Managed" means that the DCM configuration can be kept in synch and archived by DCM using the metadata repository of the local farm.

**Notes for using createCluster:**

- When creating a DCM-Managed OracleAS Cluster with the `createCluster` command, use only the following characters in the `cluster_name` argument supplied with the `-cl` option:
  
  ```
  abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_ 
  ```

- You must issue this command in the Oracle home of an Oracle Application Server Instance that belongs to an Oracle Application Server Farm. The DCM-Managed OracleAS Cluster will be created in that farm.

- The DCM-Managed OracleAS Cluster has no members when created. You can add members using `joinCluster`.

- You can create an unlimited number of DCM-Managed OracleAS Clusters.

**Example**  
dcmctl createCluster -cl cluster1

A.3.9 createComponent

Creates an OC4J instance.

**Type**  
Configuration Management

**Syntax**  
createComponent -ct oc4j -co component_name

**Description**  
Creates a new OC4J instance belonging to the local Oracle Application Server Instance. You cannot specify another Oracle Application Server Instance with the `-i` option; the command operates locally. Note that OC4J is currently the only component type allowed for this command.
OC4J instance names and J2EE application names should not contain the host name, the Oracle home or IP address of the computer containing the Oracle Application Server installation. In a clustered environment, this applies to the host name, Oracle home, IP address of any Oracle Application Server installation in the DCM-Managed OracleAS Cluster.

For example, if your computer has the host name of abc.com, you should not create a new OC4J instance or J2EE application that consists of or contains abc.com. This rule also applies to the Oracle home directory path or the IP address of the computer.

The reason for this restriction has to do with the clustering, archival, and cloning features of DCM. When archiving or capturing configuration parameters, DCM scans all configuration files for occurrences of the host name, Oracle home and IP address. When it finds occurrences of these values, it replaces the values with symbolic macros and stores them in the DCM Repository.

When DCM applies these configurations to other installations, the occurrences of host name, Oracle home and IP address are replaced with the values for that installation to which the configuration is being applied. Consequently, if OC4J instance names or J2EE application names contain host names, Oracle home names or IP addresses, they are inappropriately substituted when the configuration is applied to other installations.

Example

dcmctl createComponent -ct oc4j -co OC4J_myapps

A.3.10 deployApplication

Deploys a J2EE application.

Type

Application

Syntax

deployApplication -f file -a app_name [-co comp_name] [-enableIIOp] [-rc rootcontext] [-pa parent_name]

In the preceding command:

- file is the name of the WAR or EAR file to deploy
- app_name is the name of the application specified by the user in original deployment
- comp_name is the name of the OC4J instance to which the application will be deployed. The default is the home instance.
- enable IIOp enables the Internet Inter-Orb Protocol
- rootcontext is the base path used in the URL to access the web module (for example, http://hostname:port/context root). Applies to deployment of WAR files only.
parent_name is the parent application name. The parent application contains common classes used by child applications.

**Description**
The J2EE application is deployed to the specified Oracle Application Server Instance. The deployApplication command does the following:

1. Copies the WAR or EAR file from the specified location to the server.
2. Expands the WAR or EAR file into the runtime directories.
3. Changes the context root in the OHS configuration file, the mount points in the oc4j.conf file, and the server.xml file as necessary.
4. Persists all files into the DCM repository.
5. (If multiple nodes) Copies files from the local node to the remote node(s), and invokes the remote node processing.

**Example**
To deploy an application to the home OC4J instance:

```
dcmctl deployApplication -f app1.ear -a app1
```

To deploy an application to the OC4J_my_apps instance:

```
dcmctl deployApplication -f app1.ear -a app1 -co OC4J_myapps
```

To deploy a WAR file to the home OC4J instance:

```
dcmctl deployApplication -f app2.war -a app1 -rc /myiAS/myWebapps
```

**A.3.11 destroyInstance**
Removes an Oracle Application Server Instance from the DCM Metadata Repository.

---

**Note:** This command is intended for use only with specific guidance from Oracle support, because its effects are irreversible (that is, the Oracle Application Server Instance is unrecoverable). The Oracle Universal Installer uses this command in its deinstallation routine to remove all DCM data associated with the named Oracle Application Server Instance from the Metadata Repository.

---

**Type**
Configuration Management

**Syntax**
destroyInstance -i instance_name

**Description**
Removes all information related to the specified Oracle Application Server Instance from the DCM repository. The need for this command arises when an Oracle Application Server Instance is removed using operating system commands on files or directories, and the repository information about the Oracle Application Server Instance remains. This may cause problems in subsequent installation attempts. The
destroyInstance command clears the repository of all vestiges of an Oracle Application Server Instance that was removed precipitously.

If the command is executed in the Oracle Application Server Instance being destroyed, the dcm.conf file, targets.xml file, and the repository directory will be cleaned up. If it is executed remotely, you should check to ensure that the instance-related information has been removed, and, if not, remove it manually.

Example

dcmctl destroyInstance -i instance1

A.3.12 echo
Displays the specified string to standard output.

Type
Shell

Syntax
echo

Description
Used in a dcmctl command script to display a specified string to standard output.

Example

echo "this is a comment"

A.3.13 exit
Exits a dcmctl shell client.

Type
Shell

Syntax
exit

Description
Exits a dcmctl shell client. This command is only applicable to the shell; it does not affect the dcmctl daemon.

Example

exit
A.3.14 exportArchive

Exports the named archive from the repository to a JAR file.

**Type**
Archive

**Syntax**
exportArchive -arch archiveName -f myFile [-comment myComments]

**Description**
See "Exporting and Importing Archives" on page 3-2.

**Example**
dcmctl exportArchive -arch archive1 -f /exports/testConfig -comment "this is an export of archive1"

A.3.15 exportRepository

Copies the named file-based repository to the specified location.

**Type**
Configuration Management

**Syntax**
exportRepository -f myFile [-force]

**Description**
Copies the file-based repository information to the location specified. Use the -force option to overwrite an existing file. If you do not use the -force option and the named file exists, an exception is thrown.

Note for using exportRepository:

- Typically, when you use the exportRepository command, you also use the importRepository command on another Oracle Application Server Instance. Before executing the importRepository command, stop all DCM daemons in the Oracle Application Server Instances that are part of the Oracle Application Server Farm in which you run importRepository. Use the following command in each Oracle Application Server Instance in the Oracle Application Server Farm to stop the DCM daemon:

  \[
  \text{ORACLE HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon}
  \]

**Example**

This example assumes two Oracle Application Server Instances: instance1 and instance2. To relocate the file-based repository host from instance1 to instance2, perform the following steps:

On instance1, the original file-based repository host issue these commands:

  \[
  \text{ORACLE HOME/dcm/bin> dcmctl exportrepository -f /export/repository_save_file}
  \]

  \[
  \text{ORACLE HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon}
  \]
If there are more than two Oracle Application Server Instances, stop the daemon on all of the other Oracle Application Server Instances.

On instance2, issue these commands:

```
ORACLE_HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon
ORACLE_HOME/dcm/bin> dcmctl importrepository -f /export/repository_save_file
```

On instance1, issue this command:

```
ORACLE_HOME/dcm/bin> dcmctl repositoryrelocated
```

After the `repositoryRelocated` command completes, start the dcm daemons, as follows:

On instance1 issue this command:

```
ORACLE_HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

When this command completes, on instance2, issue the command:

```
ORACLE_HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

If there are more than two Oracle Application Server Instances, on all of the other Oracle Application Server Instances issue the command:

```
ORACLE_HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

### A.3.16 `fileCleanUp`

Purges outdated records found in the file-based repository.

**Type**
Configuration Management

**Syntax**

```
fileCleanUp dump | clean [-force]
```

**Description**

Purges outdated records from the file-based repository. The data tables in a file-based repository are limited in size, and should be purged periodically to make room for new transactions. DCM operations may fail when the tables become full.

You must stop the DCM daemon (`opmnctl stopproc ias-component=dcm-daemon`) before you execute the `fileCleanUp clean` command.

Backing up the `ORACLE_HOME/dcm/repository` directory is recommended before using `fileCleanUp`.

**Example**

To list the contents of the repository tables:

```
dcmctl fileCleanUp dump
```

To purge the outdated records from the repository:

```
dcmctl fileCleanUp clean
```
A.3.17 getComponentType

Returns the type of the component in the local Oracle Application Server Instance or specified Oracle Application Server Instance.

**Type**
Configuration Management

**Syntax**
getComponentType [-i instance_name] -co component_name

**Description**
Returns the type of the component to standard output. By default, it returns the type of the component in the local Oracle Application Server Instance. You can use the -i option to specify a different Oracle Application Server Instance.

**Example**
To obtain the type of the home component in the local Oracle Application Server Instance:

dcmctl getComponentType -co home

OC4J

To obtain the type of the OC4J_SECURITY component in instance1:

dcmctl getComponentType -co OC4J_SECURITY -i instance1

OC4J

A.3.18 getError

Displays descriptions of errors.

**Type**
dcmctl Properties

**Syntax**
getError [error_number | error_name]

**Description**
Displays error descriptions. If you issue this command with no arguments, it displays the error message from the most recent DCM error that occurred. If the debug option is set to on, the stack trace is printed, if there was one. If you provide an error number or error name, it displays the message for that error.

**Example**
To view the description of the error that occurred most recently:

dcmctl getError

You can use the following commands to print the messages for ADMN-906025:

dcmctl getError 906025
dcmctl getError ADMN-906025
A.3.19 getOPMNPort

Returns the hostname and Oracle Notification Service remote port.

Type
Manually-Managed OracleAS Cluster

Syntax
getOPMNPort

Description
This command returns the hostname and the ONS remote port for the local Oracle Application Server Instance. It retrieves this information from the ons.conf file.

Example
dcmctl getopmnport
myhost.example.com:6200

A.3.20 getRepositoryId

Returns the repository ID of an OracleAS File-based Farm repository.

Type
Configuration Management

Syntax
getRepositoryId

Description
Returns the repository identifier of the OracleAS File-based Farm to which the Oracle Application Server Instance belongs. If the Oracle Application Server Instance is a standalone Oracle Application Server Instance, this command returns the repository identifier for the Oracle Application Server Instance. If the standalone Oracle Application Server Instance is to be used to establish a new distributed File-based repository, then use the returned repository identifier to initialize the repository host with joinFarm and the -r option.

Example
dcmctl getRepositoryId
A.3.21 getReturnStatus

Returns the status of the last dcmctl command.

**Type**
dcmctl Properties

**Syntax**
getReturnStatus

**Description**
This command displays the status of the last dcmctl command that performed an asynchronous operation (as opposed to a command that returned information). This command is intended to be used to get the status of a previous command that timed out. You can issue the getReturnStatus command repeatedly until it reports that the previous command has finished.

For synchronous operations, use the getError command to retrieve more information on the last failed command.

**Example**
In this example, the start command times out and the getReturnStatus command is used to check for status:

dcmctl getReturnStatus

ADMN-906005

The specified command, "start", is being executed asynchronously. The maximum wait time of, 120 seconds, has been reached. This operation will continue to execute to completion. Use the "getReturnStatus" command to determine if/when the operation completes successfully.

A.3.22 getState

Returns the state of the components in the indicated scope. The getState command only displays the state of the following components: OC4J, Oracle HTTP Server, and JAZN.

**Type**
Configuration Management

**Syntax**
g getState [-i instance_name] [-cl cluster_name] [-co component_name]

**Description**
Without any arguments, this command returns the state of all components in the local Oracle Application Server Instance. The state includes the following indicators:

- **Up Status** — indicates whether a component is running
- **In Sync Status** — indicates whether the component’s configuration is synchronized with the configuration in the DCM repository

You can use arguments with the command to narrow the scope of the command by Oracle Application Server Instance, DCM-Managed OracleAS Cluster, or component.
If you execute the `getState` command with `verbose` option set to `off`, the output (which is intended to facilitate parsing for scripting purposes) appears in the form:

\[x:y:z\]

where:
- \(x\) is the number of components running
- \(y\) is the number of components in the process of starting or stopping
- \(z\) is the total number of components

**Example**
To get the state of the local Oracle Application Server Instance:

```
dcmctl getState
```

To get the state of the component HTTP_Server:

```
dcmctl getState -co HTTP_Server
```

To get the state of a managed OracleAS Cluster:

```
dcmctl getState -cl cluster1
```

### A.3.23 help

Returns a listing of `dcmctl` commands or help for a specific command.

**Type**

dcmctl Properties

**Syntax**

```
help [commandName]
```

**Description**

Returns a listing of `dcmctl` commands. If a command name is specified after the `help` command, description and syntax information on that command is returned.

**Example**

To list all commands:

```
dcmctl help
```

To get help for the `createComponent` command:

```
dcmctl help createcomponent
```
A.3.24 importArchive

Imports an archive file to the current repository.

Type
Archive

Syntax
importArchive [-arch archiveName] -f myFile [-comment "myComments"]

Description
Imports the named archive file from the file system to the current repository. Use -arch to change the name and -comment to change the comment during the import.

Example
dcmctl importArchive -arch Archive1 -f /exports/testConfig -comment "this is an import"

A.3.25 importRepository

Moves an OracleAS File-based Farm repository from one Oracle Application Server Instance to another.

Type
Configuration Management

Syntax
importRepository -f file_name [-force]

Description
Moves a OracleAS File-based Farm repository from one Oracle Application Server Instance to another, based on a saved file from the exportRepository command. The repository may be restored to any Oracle Application Server Instance in the farm. If the current Oracle Application Server Instance is not hosting a repository, dcmctl prompts for confirmation of the action, unless the -force option is used.

If the repository has been moved, and the former repository host is still a member of the farm, then the repositoryRelocated command must be run in the former repository host Oracle Application Server Instance to notify it that it is no longer the host.

Note for using importRepository:

- Before executing importRepository, stop all DCM daemons in the Oracle Application Server Instances that are part of the OracleAS File-based Farm in which you are running importRepository. Use the following command in each Oracle Application Server Instance in the Oracle Application Server Farm to stop the DCM daemon:

  ORACLE_HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon
Example
This example assumes two Oracle Application Server Instances: instance1 and instance2. To relocate the OracleAS File-based Farm repository host from instance1 to instance2, perform the following steps:

On instance1, the original OracleAS File-based Farm repository host, issue these commands:

```
ORACLE HOME/dcm/bin> dcmctl exportrepository -f /export/repository_save_file
ORACLE HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon
```

If there are more than two Oracle Application Server Instances, issue the `opmnctl stopproc` command on all of the other Oracle Application Server Instances.

On instance2, issue these commands to stop the DCM daemon, and then import the saved repository:

```
ORACLE HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon
ORACLE HOME/dcm/bin> dcmctl importrepository -f /export/repository_save_file
```

On instance1, issue this command:

```
ORACLE HOME/dcm/bin> dcmctl repositoryrelocated
```

After the `repositoryrelocated` command completes, sequentially, start the DCM daemons, as follows:

On instance1, issue this command:

```
ORACLE HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

When this command completes, on instance2, issue this command:

```
ORACLE HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

If there are more than two Oracle Application Server Instances, on all of the other Oracle Application Server Instances issue this command:

```
ORACLE HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

A.3.26 isClusterable

Identifies whether an Oracle Application Server Instance can become a member of a DCM-Managed OracleAS Cluster.

Note for using DCM-Managed OracleAS Clusters:

- Oracle Application Server supports heterogeneous Oracle Application Server Instances as part of the same farm. For example, an Oracle Application Server Instance running on Solaris operating system, an Oracle Application Server Instance running on a Linux system, and an Oracle Application Server Instance running on a Windows system can reside in the same farm. Oracle Application Server Instances that you want to join to a DCM-Managed OracleAS Cluster must be of the same installation type and version and be installed on a like operating system. For example, Solaris and Linux systems may be clustered.

Type
Configuration Management

Syntax

```
isClusterable [-i instance_name | -arch archive_name]
```
Description
Identifies whether an Oracle Application Server Instance is eligible to become a member of a DCM-Managed OracleAS Cluster. By default, this command uses the local Oracle Application Server Instance. You can use the -i option to specify a different Oracle Application Server Instance. In order for an Oracle Application Server Instance to be eligible, all components in the Oracle Application Server Instance must be clusterable.

If the Oracle Application Server Instance is ineligible, and the verbose option is on, the reason is included with the false answer returned by the command.

Example
dcmctl isClusterable

A.3.27 isCompatible
Identifies whether an Oracle Application Server Instance is compatible with other members of an DCM-Managed OracleAS Cluster.

Type
Configuration Management

Syntax
isCompatible -cl cluster_name [-i instance_name | -arch archive_name]

Description
Identifies whether an Oracle Application Server Instance is compatible with other members of an DCM-Managed OracleAS Cluster. By default, this command uses the local Oracle Application Server Instance. You can use the -i option to specify a different Oracle Application Server Instance. An Oracle Application Server Instance is compatible if it has the same components configured and is of the same version.

Example
dcmctl isCompatible -cl cluster1

A.3.28 joinCluster
Adds an Oracle Application Server Instance to the named DCM-Managed OracleAS Cluster.

Notes for using DCM-Managed OracleAS Clusters:

- Oracle Application Server supports heterogeneous Oracle Application Server Instances as part of the same farm. For example, an Oracle Application Server Instance running on Solaris operating system, an Oracle Application Server Instance running on a Linux system, and an Oracle Application Server Instance running on a Windows system can reside in the same farm. Oracle Application Server Instances that you want to be part of a OracleAS Cluster must be of the same installation type and version, and be installed on a like operating system. For example, Solaris and Linux systems may be clustered.

- If you are using Oracle Enterprise Manager 10g Application Server Control Console, then after issuing joinCluster command, you must stop and then start
Oracle Enterprise Manager 10g Application Server Control Console using the commands:

```
dcmctl stop iasconsole
emctl start iasconsole
```

**Type**
Configuration Management

**Syntax**

```
joinCluster -cl cluster_name [-i instance_name]
```

**Description**

Adds an Oracle Application Server Instance to the DCM-Managed OracleAS Cluster specified with the `-cl` option. By default, this command uses the local Oracle Application Server Instance. You can specify a different Oracle Application Server Instance with the `-i` option. The Oracle Application Server Instance must be a member of the same Oracle Application Server Farm as the DCM-Managed OracleAS Cluster. There is no limit to the number of Oracle Application Server Instances you can add to a DCM-Managed OracleAS Cluster. An Oracle Application Server Instance is stopped after being added to an DCM-Managed OracleAS Cluster, so you must manually start it.

**Example**

To add the local Oracle Application Server Instance to cluster1 and restart it:

```
dcmctl joinCluster -cl cluster1
opmnctl @cluster startproc
```

To add instance1 to cluster1 and restart it:

```
dcmctl joinCluster -cl cluster1 -i instance1
opmnctl @instance startproc
```

**A.3.29 joinFarm**

Associates an Oracle Application Server Instance with an OracleAS Database-based Farm or an OracleAS File-based Farm.

Notes for using Oracle Application Server Farms and the `joinFarm` command:

- Oracle Application Server supports heterogeneous Oracle Application Server Instances as part of the same farm. For example, an Oracle Application Server Instance running on Solaris operating system, an Oracle Application Server Instance running on a Linux system, and an Oracle Application Server Instance running on a Windows system can reside in the same farm.

- If you are using Oracle Enterprise Manager 10g Application Server Control Console, then after issuing `joinFarm` command, you must stop and then start Oracle Enterprise Manager 10g Application Server Control Console using the commands:

```
emctl stop iasconsole
emctl start iasconsole
```
**Alphabetical Listing of dcmctl Commands**

### joinFarm

**Type**
Configuration Management

**Syntax**
joinFarm [-r repository_ID]

**Description**
With the -r option and a repository ID, it associates an Oracle Application Server Instance with the named OracleAS File-based Farm repository. The repository ID is a hostname and port.

If the Oracle Application Server Instance was originally associated with a OracleAS Database-based Farm, using Infrastructure database information, you can issue this command without arguments to reassociate the Oracle Application Server Instance with the original database.

You can obtain the repository ID by issuing the getRepositoryId command on any Oracle Application Server Instance that is a member of the OracleAS File-based Farm associated with the repository.

**Example**
To reassociate the Oracle Application Server Instance with a OracleAS Database-based Farm:

dcmctl joinFarm

To associate the Oracle Application Server Instance with a OracleAS File-based Farm:

dcmctl joinFarm -r myhost.mycompany.com:1899

### leaveCluster

**Type**
Configuration Management

**Syntax**
leaveCluster [-i instance_name]

**Description**
Removes an Oracle Application Server Instance from its DCM-Managed OracleAS Cluster. By default, this command uses the local Oracle Application Server Instance. You can specify a different Oracle Application Server Instance with the -i option. The Oracle Application Server Instance being removed is stopped, so you must restart it after using this command.

**Note:** When an Oracle Application Server Instance joins a farm, all archives for the Oracle Application Server Instance are removed. If you want to preserve the archives on the Oracle Application Server Instance that is joining the farm, export each of the archives with the exportArchive command prior to using the joinFarm command.
Note for using `leaveCluster`:

- If you are using Oracle Enterprise Manager 10g Application Server Control Console, then after issuing the `dcmctl leaveCluster` command, you must stop and then start Oracle Enterprise Manager 10g Application Server Control Console using the commands:

  ```
  emctl stop iasconsole
  emctl start iasconsole
  ```

**Example**

To remove the local Oracle Application Server Instance from the DCM-Managed OracleAS Cluster:

`dcmctl leaveCluster`

To remove instance1 from its DCM-Managed OracleAS Cluster:

`dcmctl leaveCluster -i instance1`

### A.3.31 `leaveFarm`

Removes an Oracle Application Server Instance from a farm.

**Type**
Configuration Management

**Syntax**

```
leaveFarm
```

**Description**

Removes an Oracle Application Server Instance from a farm. This command affects only the relationship between DCM and a repository, and has no impact on other components. Specific implications for this command on the repository and other components are as follows:

- Only the metadata for the DCM-managed configuration is moved from the centralized DCM repository to a local Oracle Application Server Instance.

- The Oracle Application Server Instance is removed from the repository, but it is not removed from the Oracle Internet Directory. Therefore, if you have a complete Infrastructure installation (the Metadata Repository and Identity Management), the `leaveFarm` command removes the Oracle Application Server Instance the Metadata Repository, but it remains in Identity Management. To remove the Oracle Application Server Instance entirely, see Section 2.2.12, "Removing an Instance From the Infrastructure After Executing the leaveFarm Command" on page 2-15.

- When an Oracle Application Server Instance is removed from a farm, any associated archives are deleted from the centralized repository. For this reason, it is a good idea to issue the `createArchive` command immediately after issuing the `leaveFarm` command. This creates a new baseline archive for the Oracle Application Server Instance.

- The `leaveFarm` command does not remove connections to the infrastructure database for other components, such as Oracle Application Server Single Sign-On or JAZN.
Note for using `leaveFarm`:

- Using Oracle Enterprise Manager 10g Application Server Control Console, after issuing the `dcmctl leaveFarm` command, you must stop and then start Oracle Enterprise Manager 10g Application Server Control Console using the commands:
  
  ```
  emctl stop iasconsole
  emctl start iasconsole
  ```

**Example**

To remove an Oracle Application Server Instance from the farm:

```
dcmctl leaveFarm
```

### A.3.32 listApplications

Lists the applications deployed in an OC4J instance within a farm, DCM-Managed OracleAS Cluster, Oracle Application Server Instance, or archive.

**Type**

Application

**Syntax**

`listApplications [-farm] [-cl cluster_name] [-i instance_name] [-arch archive_name] [-co component_name] [-sort on | off] [pattern]`

**Description**

List the applications deployed in the specified domain. The default is the home OC4J instance in the local Oracle Application Server Instance.

When using the `dcmctl` shell, you can limit the output with the pattern option, which enables you to list only the applications with names matching the pattern. Specify the pattern using the wildcard character `*` as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, `*string, string*, str*ing, *trin*, *tr*ng` and `*` are all valid uses of the wildcard to specify a pattern.

**Note:** If the asterisk character is part of the string you want to match, you must prepend it with the escape character `\` (backward slash).

**Example**

To list the applications in the `home` OC4J instance in the local application server instance:

```
dcmctl listApplications
1
Application:    ADFBC
Component Name: home
Component Type: OC4J
Cluster:        cluster1

2
Application:    ADFBCManager
Component Name: home
Component Type: OC4J
```
Cluster:    cluster1

3
Application: IsWebCacheWorking
Component Name: home
Component Type: OC4J
Cluster:    cluster1

To list the applications in myInstance:

dcmctl listApplications -i myInstance
1
Application:    ADFBC
Component Name: home
Component Type: OC4J
Instance:       myInstance

2
Application:    ADFBCManager
Component Name: home
Component Type: OC4J
Instance:       myInstance

3
Application:    IsWebCacheWorking
Component Name: home
Component Type: OC4J
Instance:       myInstance

To list only the ADFBC applications when applications ADFBC, ADFBCManager and IsWebCacheWorking are present, the command and its resulting output would be:

dcmctl shell
dcmctl> listApplications A*
1
Application:    ADFBC
Component Name: home
Component Type: OC4J
Cluster:    cluster1

2
Application:    ADFBCManager
Component Name: home
Component Type: OC4J
Cluster:    cluster1
A.3.33 listArchives

Returns a list of archive names.

**Type**
Archive

**Syntax**
listArchives [–arch archive_name][–sort on| off]

**Description**
Returns a list of automatically generated archives (in ascending order by timestamp), followed by a list of user generated archives (in ascending order by archive name). If the -arch (archive name) option is used, only the named archive is listed.

**Example**
To list information about archives:

dcmctl listArchives

****** AUTO GENERATED ARCHIVES ******

1
Name: dcm.autoarchive_138.2.142.2121ff7a1e.fcb87f2e6d.-7ffe
Source: instance: my-sun.us.oracle.com
Version: 10.1.2.0.0
Comments: Automatic archival prior to deployment of application IsWebCacheWkng
Created: 2004-08-24 12:34:24.972
Clusterable: true

2
Name: dcm.autoarchive_138.2.142.21212d3205.fcb882fb5b.-7fff
Source: instance: my-sun.us.oracle.com
Version: 10.1.2.0.0
Comments: Automatic archival prior to hand-editing of configuration files OC4J OHS opmn jazn
Created: 2004-08-24 12:35:59.168
Clusterable: true

****** USER GENERATED ARCHIVES ******

1
Name: initial_archive_my-sun.us.oracle.com
Source: instance: my-sun.us.oracle.com
Version: 10.1.2.0.0
Comments: The initial archive after joining the farm for my-sun.us.oracle.com
Created: 2004-08-24 14:01:54.158
Clusterable: true
A.3.34 listClusters

Lists the DCM-Managed OracleAS Clusters in the local Oracle Application Server Farm.

Type
Configuration Management

Syntax
listClusters

Description
This command lists the DCM-Managed OracleAS Clusters in the Oracle Application Server Farm that is associated with the local Oracle Application Server Instance.

When using the dcmctl shell, you can limit the output with the pattern option, which enables you to list only the DCM-Managed OracleAS Clusters whose names match the pattern. Specify the pattern using the wildcard character * as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, *string, string*, *STRING, *TRIN*, *TR*NG and * are all valid uses of the wildcard to specify a pattern.

Note: If the asterisk character is part of the string you want to match, you must prepend it with the escape character \ (backward slash).

Example
To list DCM-Managed OracleAS Clusters in the local farm, which contains cluster1 and cluster2:

dcmctl listClusters
1  cluster2
2  cluster1

To list the DCM-Managed OracleAS Clusters whose name ends in 1:

dcmctl shell
dcmctl> listClusters *1
1  cluster1
A.3.35 listComponents

Lists components within the named scope.

Type
Configuration Management

Syntax
listComponents [-i instance_name] [-cl cluster_name] [-farm] [-arch archive_name] [pattern] [-sort on | off]

Description
Returns a list of the components in the specified scope. Without arguments, this command returns a list of components in the local Oracle Application Server Instance.

When using the dcmctl shell, you can limit the output with the pattern option, which enables you to list only the components whose names match the pattern. Specify the pattern using the wildcard character * as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, *string, string*, st*ing, *trin*, *tr*ng and * are all valid uses of the wildcard to specify a pattern.

Note: If the asterisk character is part of the string you want to match, you must prepend it with the escape character \ (backward slash).

The listcomponents command may display components that OPMN manages. Use the opmnctl command to manage, start, stop, and restart these components.

Example
To list the components in the local Oracle Application Server Instance:

dcmctl listComponents
1
Component Name: home
Component Type: OC4J
Cluster: cluster1

2
Component Name: HTTP_Server
Component Type: HTTP_Server
Cluster: cluster1

To list only the components whose name contains the underscore character:

dcmctl shell
dcmctl> listcomponents _*
1
Component Name: HTTP_Server
Component Type: HTTP_Server
Cluster: cluster1
A.3.36 listComponentTypes

Lists supported component types.

**Type**
dcmctl Properties

**Syntax**
listComponentTypes

**Description**
Lists the component types that DCM supports.

**Example**
dcmctl listComponentTypes

A.3.37 listInstances

Lists the Oracle Application Server Instances in a Oracle Application Server Farm or DCM-Managed OracleAS Cluster.

**Type**
Configuration Management

**Syntax**
listInstances -cl cluster_name [pattern] -farm

**Description**
With no options, this command lists the Oracle Application Server Instances that belong to the same Oracle Application Server Farm as the local Oracle Application Server Instance, but are not part of an DCM-Managed OracleAS Cluster. If you use the -cl option, it lists only the Oracle Application Server Instances that are part of the specified DCM-Managed OracleAS Cluster. If you use the -farm option, it lists all Oracle Application Server Instances in the farm, identifying DCM-Managed OracleAS Cluster membership (if any) for each.

When using the dcmctl shell, you can limit the output with the pattern option, which enables you to list only the Oracle Application Server Instances whose names match the pattern. Specify the pattern using the wildcard character * as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, *string,string*, string*, string* and * are all valid uses of the wildcard to specify a pattern.

---

**Note:** If the asterisk character is part of the string you want to match, you must prepend it with the escape character \ (backward slash).

---

**Example**
Assume an Oracle Application Server Farm that includes five Oracle Application Server Instances. Two belong to the UNIX DCM-Managed OracleAS Cluster cluster1, two belong to the Windows DCM-Managed OracleAS Cluster cluster2, and one is a non-clustered Windows Oracle Application Server Instance.
Alphabetical Listing of dcmctl Commands

To list all non-clustered Oracle Application Server Instances in the Oracle Application Server Farm associated with the local Oracle Application Server Instance:

dcmctl listInstances
1
Instance name: 10gTwo.my-pc.us.oracle.com
Cluster: 
Hostname: my-pc.us.oracle.com
Oracle Home: C:\10gHome2

To list the Oracle Application Server Instances in cluster1:

dcmctl listInstances -cl cluster1
1
Instance name: 1012AS01.my-sun.us.oracle.com
Cluster: cluster1
Hostname: my-sun.us.oracle.com
Oracle Home: /private2/1012AS01

2
Instance name: AS1012.my-sun.us.oracle.com
Cluster: cluster1
Hostname: my-sun.us.oracle.com
Oracle Home: /private2/1012AS

To list the Oracle Application Server Instances in cluster 2:

dcmctl listInstances -cl cluster2
1
Instance name: M28.my-pc.us.oracle.com
Cluster: cluster2
Hostname: my-pc.us.oracle.com
Oracle Home: C:\M28Home

2
Instance name: 10gOne.my-pc.us.oracle.com
Cluster: cluster2
Hostname: my-pc.us.oracle.com
Oracle Home: C:\10gHome1

To list all Oracle Application Server Instances in the farm:

dcmctl listInstances -farm
1
Instance name: 10gTwo.my-pc.us.oracle.com
Cluster: 
Hostname: my-pc.us.oracle.com
Oracle Home: C:\10gHome2

2
Instance name: M28.my-pc.us.oracle.com
Cluster: cluster2
Hostname: my-pc.us.oracle.com
Oracle Home: C:\M28Home

3
Instance name: 10gOne.my-pc.us.oracle.com
Cluster: cluster2
Hostname: my-pc.us.oracle.com
Oracle Home: C:\10gHome1
Instance name: 1012AS01.my-sun.us.oracle.com
Cluster: cluster1
Hostname: my-sun.us.oracle.com
Oracle Home: /private2/1012AS01

Instance name: AS1012.my-sun.us.oracle.com
Cluster: cluster1
Hostname: my-sun.us.oracle.com
Oracle Home: /private2/1012AS

To list Oracle Application Server Instances within the Oracle Application Server Farm whose name contains pc:

```bash
dcmctl shell
dcmctl> listinstances -farm *pc*
1
Instance name: 10gTwo.my-pc.us.oracle.com
Cluster: 
Hostname: my-pc.us.oracle.com
Oracle Home: C:\10gHome2

2
Instance name: M28.my-pc.us.oracle.com
Cluster: cluster2
Hostname: my-pc.us.oracle.com
Oracle Home: C:\M28Home

3
Instance name: 10gOne.my-pc.us.oracle.com
Cluster: cluster2
Hostname: my-pc.us.oracle.com
Oracle Home: C:\10gHome1
```

**A.3.38 listOPMNLinks**

Lists the Oracle Application Server Instances that are in a Manually-Managed OracleAS Cluster with the local Oracle Application Server Instance.

**Type**
Manually-Managed OracleAS Cluster

**Syntax**
listOPMNlinks

**Description**
This command lists all Oracle Application Server Instances that are in a Manually-Managed OracleAS Cluster with the local Oracle Application Server Instance. These Oracle Application Server Instances could have been added to the Manually-Managed OracleAS Cluster using the addOPMNLink command.

**Example**
dcmctl listopmnlinks
host1:6200
host2:6200
A.3.39  quit

Ends a dcmctl shell client session.

**Type**
Shell

**Syntax**
quit

**Description**
Ends a dcmctl shell client session. This command is only applicable to the shell; it does not affect the dcmctl daemon.

**Example**
quit

A.3.40  redeployApplication

Redeploys a J2EE application.

**Type**
Application

**Syntax**
redeployApplication -f file -a app_name [-co comp_name] [-enableIIOP] [-rc rootcontext]

In the preceding command:
- *file* is the name of the WAR or EAR file to deploy
- *app_name* is the name of the application specified by the user in original deployment
- *comp_name* is the name of the OC4J instance to which the application will be deployed. The default is the home instance.
- *enableIIOP* enables the Internet Inter-Orb Protocol
- *rootcontext* is the base path used in the URL to access the web module (for example, http://hostname:port/context root). Applies to deployment of WAR files only.

**Description**
This command redeploy a J2EE application (WAR or EAR file) to the local Oracle Application Server Instance.
Example

To redeploy app1.ear to the home OC4J instance:

dcmctl redeployApplication -f app1.ear -a app1

To redeploy app1.ear to the OC4J_myapps instance:

dcmctl redeployApplication -f app1.ear -a app1 -co OC4J_myapps

To redeploy app2.war to the home OC4J instance:

dcmctl redeployApplication -f app2.war -a app1 -rc /myiAS/myWebapps

A.3.41 removeArchive

Removes one or more archive files.

Type

Archive

Syntax

removeArchive -arch [archive_name] [pattern]

Description

Removes the specified archive files from the DCM repository.

When using the dcmctl shell, you can specify multiple archives to remove with the pattern option, which removes all archives with names matching the pattern. Specify the pattern using the wildcard character * as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, *string, string*, st*ing, *trin*, *tr*ng and * are all valid uses of the wildcard to specify a pattern.

Note: If the asterisk character is part of the string you want to match, you must prepend it with the escape character \ (backward slash).

Example

To remove archive3:

dcmctl removeArchive -arch archive3

To remove all archives whose name begins with init*:

dcmctl shell
dcmctl> removeArchive init*
A.3.42 removeCluster

Removes one or more DCM-Managed OracleAS Clusters from the farm.

**Type**
Configuration Management

**Syntax**
removeCluster -cl [cluster_name] [pattern]

**Description**
Removes the specified DCM-Managed OracleAS Cluster from its farm, or all DCM-Managed OracleAS Clusters whose name matches the pattern string. The DCM-Managed OracleAS Cluster must contain no Oracle Application Server Instances when it is removed. This command destroys all information about the DCM-Managed OracleAS Cluster in the DCM repository.

When using the dcmctl shell, you can specify multiple DCM-Managed OracleAS Clusters to remove with the pattern option, which removes all DCM-Managed OracleAS Clusters with names matching the pattern. Specify the pattern using the wildcard character * as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, *string, string*, string*, string*, string* and * are all valid uses of the wildcard to specify a pattern.

**Note:** If the asterisk character is part of the string you want to match, you must prepend it with the escape character \ (backward slash).

**Example**
To remove cluster1 from a Oracle Application Server Farm containing cluster1, cluster2 and cluster3:

dcmctl removeCluster -cl cluster1

To remove all DCM-Managed OracleAS Clusters from a Oracle Application Server Farm containing cluster1, cluster2, cluster3:

dcmctl shell
dcmctl> removeCluster -cl *


A.3.43 removeComponent

Destroy one or more OC4J instances.

Type
Configuration Management

Syntax
removeComponent -co [component_name] [pattern]

Description
Destroy the named OC4J instance, or all OC4J instances in the local Oracle Application Server Instance whose name matches the pattern string. Note that OC4J is the only component type allowed for this command. You cannot remove an OC4J instance that was created by the installation process (such as OC4J_SECURITY).

When using the dcmctl shell, you can specify multiple components to remove with the pattern option, which removes all components with names matching the pattern. Specify the pattern using the wildcard character * as a leading, trailing or embedded character (or in combination) in the string you want to match, or by itself to specify all. For example, *string, string*, st*ing, *trin*, *tr*ng and * are all valid uses of the wildcard to specify a pattern.

Note: If the asterisk character is part of the string you want to match, you must prepend it with the escape character \ (backward slash).

Example
To destroy the instance OC4J_myapps:

dcmctl removeComponent -co OC4J_myapps

To destroy all OC4J instances whose names begin with my:

dcmctl shell
dcmctl> removeComponent -co my*
A.3.44 removeOPMNLInk

Removes Oracle Application Server Instances from a Manually-Managed OracleAS Cluster.

**Type**
Manually-Managed OracleAS Cluster

**Syntax**
removeOPMNLInk hostname:port[, hostname:port...]

**Description**
Removes one or more Oracle Application Server Instances from a Manually-Managed OracleAS Cluster. You must run this command in the Oracle home of each Oracle Application Server Instance in the Manually-Managed OracleAS Cluster.

**Example**
This example shows a Manually-Managed OracleAS Cluster with three Oracle Application Server Instances on host1, host2, and host3. The command removes the host2 Oracle Application Server Instance.

```
HOST2_ORACLE_HOME/dcm/bin/dcmctl listopmnlinks
host1:6200
host2:6200
host3:6200
HOST1_ORACLE_HOME/dcm/bin/dcmctl removeopmnlink host2:6200
HOST2_ORACLE_HOME/dcm/bin/dcmctl removeopmnlink host2:6200
HOST3_ORACLE_HOME/dcm/bin/dcmctl removeopmnlink host2:6200
```

A.3.45 repositoryRelocated

Notifies an Oracle Application Server Instance that it is no longer hosting a repository.

**Note:** Usually, you use the `repositoryRelocated` command after an associated `exportRepository` and `importRepository`. Before running `importRepository` and `repositoryRelocated`, stop all DCM daemons in the Oracle Application Server Instances that are part of the Oracle Application Server Farm in which you run `repositoryRelocated`.

Use the following command in each Oracle Application Server Instance in the Oracle Application Server Farm to stop the DCM daemon:

```
ORACLE_HOME/opmn/bin> opmncctl stopproc ias-component=dcm-daemon
```

**Type**
Configuration Management

**Syntax**
repositoryRelocated
Description
Notifies an Oracle Application Server Instance that the repository it formerly hosted has been imported to another Oracle Application Server Instance. This command is issued in the Oracle Application Server Instance that formerly hosted the repository.

This command is used in conjunction with the `importRepository` command. An Oracle Application Server Farm can only be associated with one repository at a time. When the repository is imported to a new Oracle Application Server Instance of the Oracle Application Server Farm, the old Oracle Application Server Instance must be notified, with the `repositoryRelocated` command, that it is no longer the repository host.

Example
This example assumes two Oracle Application Server Instances: instance1 and instance2.

To relocate the file-based repository host from instance1 to instance2, perform the following steps:

On instance1, the original file-based repository host, issue these commands:

```
ORACLE HOME/dcm/bin> dcmctl exportrepository -f /export/repository_save_file
ORACLE HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon
```

If there are more than two Oracle Application Server Instances, stop the daemon on all of the other Oracle Application Server Instances.

On instance2, issue these commands:

```
ORACLE HOME/opmn/bin> opmnctl stopproc ias-component=dcm-daemon
ORACLE HOME/dcm/bin> dcmctl importrepository -f /export/repository_save_file
```

On instance1, issue this command:

```
ORACLE HOME/dcm/bin> dcmctl repositoryrelocated
```

After the `repositoryrelocated` command completes, start the DCM daemons as follows:

On instance1, issue this command:

```
ORACLE HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

When the command completes, on instance2, issue this command:

```
ORACLE HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```

If there are more than two Oracle Application Server Instances, on all of the other Oracle Application Server Instances issue this command:

```
ORACLE HOME/opmn/bin> opmnctl startproc ias-component=dcm-daemon
```
A.3.46 resetDCMCachePort

Updates the port used by the DCM cache for Oracle Application Server Instance discovery in DCM-Managed OracleAS Clusters that are managed using a file-based repository.

**Type**
Configuration Management

**Syntax**
resetDCMCachePort [new_port_number] [-r]

**Description**
Changes the port value used by the DCM cache in DCM-Managed OracleAS Clusters in an OracleAS File-based Farm. To find the current host and port, issue the command without arguments, as follows:

dcmctl resetDCMCachePort

To update the port associated with the current Oracle Application Server Instance, issue the command with the new port number, as follows:

dcmctl resetDCMCachePort 12345

If the port value is changed on the Oracle Application Server Instance that is hosting the repository, other Oracle Application Server Instances in the OracleAS File-based Farm may not be able to locate the repository. If this occurs, issue the following command in the Oracle Application Server Instance that cannot locate the repository:

dcmctl resetDCMCachePort -r 12345

In the preceding command, 12345 is the port number set at the repository host.

**Example**
To find the current host and port:

dcmctl resetDCMCachePort

To update the port associated with the current Oracle Application Server Instance:

dcmctl resetDCMCachePort 12345

To notify an Oracle Application Server Instance of the location of the repository:

dcmctl resetDCMCachePort -r 12345
A.3.47 resetFileTransaction

Resets a file-based repository to its pre-transaction state after an interrupted operation.

**Type**

Configuration Management

**Syntax**

resetFileTransaction

**Description**

Resets a file-based repository. If an operation on a File-based repository is interrupted with `control-c`, uncommitted information may be left in the repository. This command blocks all subsequent updates to the repository, cleans up uncommitted data, and reopens the repository for update.

**Example**

dcmctl resetfiletransaction

A.3.48 resetHostInformation

Updates IP address or hostname information.

**Type**

Configuration Management

**Syntax**

resetHostInformation [-r repository_hostname]

**Description**

If the IP address or hostname information has changed for an Oracle Application Server Instance in the farm, this command updates the repository and the `ons.conf` file with the new information. This command is used in the local Oracle Application Server Instance. If the host information changes for the repository host of an OracleAS File-based Farm, it is best to have all Oracle Application Server Instances running, so that all Oracle Application Server Instances can locate the repository.

For Oracle Application Server Instances that were not running during the change, it may be necessary to update the repository host information directly. Use the `–z` option and the name of the repository host to do this.

**Example**

resethostinformation
A.3.49 restart

This command is deprecated in Oracle Application Server 10g. Use opmnctl to manage processes in Oracle Application Server 10g. See the Oracle Process Manager and Notification Server Administrator’s Guide for information on using opmnctl.

A.3.50 restoreInstance

This command is deprecated in Oracle Application Server 10g. Use archiving commands to save and restore instances in Oracle Application Server 10g. To move configuration between Oracle Application Server Instances in a farm, use the applyClusterTo and applyInstanceTo commands. To move configuration between farms, use the exportArchive and importArchive commands.

A.3.51 resyncInstance

Resynchronizes Oracle Application Server Instance configuration files with the DCM repository.

Type
Configuration Management

Syntax
resyncInstance [-force] [-i instance_name]

Description
Resynchronizes the Oracle Application Server Instance configuration files with the contents of the DCM repository. This command takes all data from the repository that is not yet propagated and writes it out to the configuration files for the specified Oracle Application Server Instance. It updates the Oracle HTTP Server, OC4J, and OPMN configuration files, as well as the targets.xml file. It may also deploy or undeploy applications and add or remove components, as required. This could involve lengthy processing; use this command judiciously.

This command operates on the local Oracle Application Server Instance, unless you specify a different Oracle Application Server Instance with the -i option.

By default, the command only updates the configuration files for components whose In Sync Status is false (see getState). You can use the -force option to force it to update all configuration information.

Example
To resynchronize instance1 with the contents of the DCM repository:

dcmctl resyncinstance -i instance1

To force all files in the local Oracle Application Server Instance to be updated with that is in the DCM repository:

dcmctl resyncinstance -force
A.3.52  saveInstance

This command is deprecated in Oracle Application Server 10g. Use createArchive to save an instance's configuration.

A.3.53  set

Sets dcmctl options, timeout value, sorting preference, and number of versions for automatically archived Oracle Application Server Instances.

**Type**
dcmctl Properties

**Syntax**
dcmctl set [-v off | on] [-d off | on] [-t timeout_value] [-arch number_of_auto_archive_versions] [-sort on]

**Description**
Sets properties of the dcmctl utility. The set command enables you to set the -verbose and -debug flags on a persistent basis, and the default time out value to be set on a persistent basis. You can also specify the number of versions to be automatically archived, and whether to sort the output of list commands.

When set is run without any arguments, it displays the current settings.

**Example**
To turn verbose and debug off, and set the timeout value to 200:

dcmctl set -v off -d off -t 200

To view current settings:

dcmctl set
  Verbose:  true
  Sort:  false
  Debug:  true
  Default Timeout:  120
  Auto Archive Count:  10

To set name sorting of ‘list’ command output on:

set -sort on

To automatically archive 45 versions of an Oracle Application Server Instance:

set -arch 45

To turn off automatic archiving:

set -arch 0
A.3.54 setLogLevel

Sets the logging level.

**Type**
Shell

**Syntax**
Level [-admin] [error] [notification] [debug] [trace]

**Description**
Sets the logging level for the dcmctl client shell, or for the daemon (with the -admin option).

**Example**
To set the DCM daemon logging level:
```
setLogLevel -admin notification
```

To set the dcmctl shell log level:
```
setLogLevel notification
```

A.3.55 shell

Starts the dcmctl shell.

**Type**
Shell

**Description**
Starts the dcmctl shell, and, optionally, executes the commands in a named file.

**Syntax**
dcmctl shell [-f file_name]

**Example**
To start the shell:
```
dcmctl shell
```

To start the shell and execute the commands in the file myDCMCommands:
```
dcmctl shell -f myDCMCommands
```
A.3.56 shutdown
This command is deprecated in Oracle Application Server 10g. Use opmnctl to manage processes in Oracle Application Server 10g. See the Oracle Process Manager and Notification Server Administrator’s Guide for information on using opmnctl.

A.3.57 start
This command is deprecated in Oracle Application Server 10g. Use opmnctl to manage processes in Oracle Application Server 10g. See the Oracle Process Manager and Notification Server Administrator’s Guide for information on using opmnctl.

A.3.58 stop
This command is deprecated in Oracle Application Server 10g. Use opmnctl to manage processes in Oracle Application Server 10g. See the Oracle Process Manager and Notification Server Administrator’s Guide for information on using opmnctl.

A.3.59 updateConfig
Updates the repository with information from local configuration files.

The purpose of the updateConfig operation is to take the configuration that is currently stored in the local file system and place it into the DCM repository. This is a coarse grained operation with minimal validation of the content of the configuration file.

The dcmctl updateConfig command should be used in limited and controlled situations. It is recommended that when changing DCM-managed configuration for Oracle HTTP Server, OC4J, OC4J applications, OPMN, or JAZN that you use the Application Server Control Console. If you use the Application Server Control Console, then you do not need to use updateConfig.

If you need to manually edit configuration files for a component, you must use updateConfig to place these changes into the DCM repository. If you make manual changes and you do not run updateConfig the changes will be overwritten the next time that the configuration is resynchronized.

---

**Caution:** Do not run updateConfig concurrently with any other dcmctl commands or while performing Application Server Control Console configuration operations across multiple Oracle Application Server Instances in a Oracle Application Server Farm or DCM-Managed OracleAS Cluster.

---

**Type**
Configuration Management

**Syntax**
updateConfig [-ct component_type [, component_type...]] [-force]

**Description**
Updates the DCM repository with the information in local configuration files.

To do this, DCM obtains all configuration documents from the OHS, OC4J, OPMN and JAZN plug-ins and copies them to the repository. The updated documents are given
back to the plug-ins and each plug-in decides whether or not to notify the respective component.

With no arguments, this command updates all DCM managed components, configuration files, as well as the targets.xml file. It does not cause all applications to be redeployed, but if an EAR or an expanded EAR file was changed manually since the last deployment, it will redeploy the application.

You can specify which component’s configuration files to update with the -ct option.

Valid component types are:

- ohs: Oracle HTTP Server
- oc4j: Oracle Application Server Containers for J2EE
- opmn: Oracle Process Manager and Notification Server
- jazn: Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider

When you make manual configuration file changes in conjunction with the updateConfig command, use this approach:

1. Verify that the Oracle Application Server Instance has the most current configuration from the DCM repository. Issue the resyncInstance command to resync the configuration.
2. Create an explicit archive for the Oracle Application Server Instance or the OracleAS Cluster using createArchive.
3. Ensure that there are no other administrative operations taking place in the Oracle Application Server Farm that may alter the configuration while you are making the manual change to the configuration. This includes Oracle Enterprise Manager 10g Application Server Control Console changes, deployments, or other dcmctl shell or dcmctl commands running in the farm.
4. Make the manual configuration file change and test the change, if possible.
5. Execute dcmctl updateConfig to place the updated configuration files into the DCM repository.

If safety precautions outlined in step 3 are not followed, there is a risk that conflicting changes will be placed in the repository. This could leave the configuration stored in the repository in a non-functional state, and could require a restore from the archive created in step 2. If you restore from the archive, you will need to restart at step 1.

**Example**

To register all configuration files with the Distributed Configuration Management repository and restart the Oracle HTTP Server:

```
ORACLE_HOME/dcm/bin> dcmctl updateConfig
ORACLE_HOME/opmn/bin> opmnctl restartproc ias-component=HTTP_Server
```

If you have just updated only an Oracle HTTP Server configuration file, you can register the change with the Distributed Configuration Management repository and restart Oracle HTTP Server as follows:

```
dcmctl updateConfig -ct ohs
ORACLE_HOME/opmn/bin> opmnctl restartproc ias-component=HTTP_Server
```
A.3.60 undeployApplication

Undeploys applications.

**Type**
Application

**Syntax**
undeployApplication -a application_name -co instance_name

**Description**
Undeploys the named application in the named OC4J instance.

**Example**
To undeploy the application testApp in the home OC4J instance:

```
undeployApplication -a testApp -co home
```

A.3.61 validateEarFile

Checks an EAR file for J2EE compliance.

**Type**
Application

**Syntax**
validateEarFile -f file [-noproxy]

**Description**
Examines the named EAR file and lists characteristics that are not compliant with the J2EE specification.

You may need to set up a proxy to enable access to DTDs on the Web. You can pass a parameter to the JVM using the ORACLE_DCM_JVM_ARGS environment variable to specify the proxy host and port.

**Example**
To validate the Pet Store EAR file:

```
dcmctl validateEarFile -f petstore.ear
Warning: J2EE/DTD validation errors were foundADMN-906001
{0}   Base Exception:
oracle.ias.sysmgmt.deployment.j2ee.exception.J2eeDeploymentException:Cannot get
xml document by parsing /var/tmp/jar50152.tmp: Invalid element 'servlet' in
content of 'web-app', expected elements '[servlet-mapping, session-config,
mime-mapping, welcome-file-list, error-page, taglib, resource-ref,
security-constraint, login-config, security-role, env-entry, ejb-ref]'.
```
A.3.62 whichCluster

Returns the DCM-Managed OracleAS Cluster name for the named Oracle Application Server Instance.

**Type**
Configuration Management

**Syntax**
whichCluster [-i instance_name]

**Description**
Returns the name of the DCM-Managed OracleAS Cluster that contains the named Oracle Application Server Instance. If no Oracle Application Server Instance is specified, returns the name of the DCM-Managed OracleAS Cluster that contains the local Oracle Application Server Instance.

**Example**
dcmctl whichCluster

A.3.63 whichFarm

Identifies the type and location of the farm.

**Type**
Configuration Management

**Syntax**
dcmctl whichFarm

**Description**
Returns Oracle Application Server Farm name, host Oracle Application Server Instance, host name, and repository type, and SSL usage. If the repository is hosted by a third-party database, the hosting Oracle Application Server Instance name and host name are not available.

**Example**
dcmctl whichFarm
Farm Name: .private2.1012AS.dcm.repository
Host Instance: AS1012.my-sun.us.oracle.com
Host Name: my-sun.us.oracle.com
Repository Type: Distributed File-based (host)
SSL In Use: false
A.3.64 whichInstance

Returns the Oracle Application Server Instance name.

Type
Configuration Management

Syntax
whichInstance

Description
Returns the name of the local Oracle Application Server Instance.

Example
dcmctl whichInstance
1012AS01.my-sun.us.oracle.com
This appendix describes common problems that you might encounter when using DCM and explains how to solve them. It also gives detailed instructions on how to diagnose problems. It contains the following topics:

- Problems and Solutions
- Diagnosing Problems with Components Related to DCM
- Diagnosing DCM Problems

B.1 Problems and Solutions

This section describes problems reported by, or encountered while using, Distributed Configuration Management and provides instructions for resolving these problems. It contains the following topics:

- OPMN Cannot Start
- DCM Daemon Cannot Start
- DCM Daemon Does Not Start As Expected
- Invalid username/password
- BOM (Bill of Materials) Not Updated
- In Sync Status is False
- Unable to Connect to the Directory
- Cannot Access the Infrastructure Database
- The dcm.conf File is Unreadable
- The dcm.conf or dcmCache.xml File is Empty
- dcmctl getState Hangs
- getState Cannot Be Used With (9.0.2 or 9.0.3) Instances
- dcmctl joinFarm Produces Error
- repositoryRelocated Command Causes Repository to be Unreachable
- Unreachable IP Address
- resyncInstance Command Does Not Restore Properly
B.1.1 OPMN Cannot Start

**Problem**
During a configuration operation, the opmn.xml file was corrupted or removed. This error may be accompanied by one of the following error messages:

**ADMN-202001**
An exception has occurred while accessing DCM repository. Please refer to the accompanying base exception for more details.
Base Exception:
 oracle.ons.ONSException:Unable to open file /private/10g/opmn/conf/opmn.xml

or

**ADMN-705002**
The most likely causes are:
1. Configuration error; TaskMaster cannot read configuration file;
2. Cannot connect to the repository.
Base Exception:
 oracle.ons.SubscriptionException:Subscription request timed out after 120000 milliseconds. Possible causes: OPMN is not running, you may have OPMN running in an alternate ORACLE_HOME using duplicate port values, OPMN may be misconfigured.

**Solution 1**
Restart OPMN using the last saved version of the opmn.xml configuration file and then perform a dcmctl resyncinstance command to update the configuration files with the version last saved to the repository.

To do this, perform the following steps:

1. Perform a opmnctl validate operation on the saved opmn.xml.bak file.
   
   % cd $ORACLE_HOME/opmn/conf
   % $ORACLE_HOME/opmn/bin/opmnctl validate opmn.xml.bak

   **Note:** The opmn.xml.bak file is created upon installation. It is re-created before any change (through DCM or Oracle Enterprise Manager 10g) is committed.

2. If step 1 returns the following, then continue to the next step:
   
   opmnctl: opmn validation succeeded!

   If step one returns with the message:

   opmnctl: opmn validation failed!

   Then, if you know how to correct the error in opmn.xml.bak then correct it and try step 1 again. Otherwise, try Solution 2.

3. Replace the opmn.xml file with the file that was saved in the last DCM operation.
   The saved version of opmn.xml should be in the file $ORACLE_HOME/opmn/conf/opmn.xml.bak.

   For example:

   % cd $ORACLE_HOME/opmn/conf
   % cp opmn.xml.bak opmn.xml
4. Start the DCM shell and run the resyncinstance command.
   ```
   % cd $ORACLE_HOME/dcm/bin
   % dcmctl shell
   dcmctl> resyncinstance -force
   All OC4J components are stopped during a resyncinstance -force.
   They will need to be restarted after the command completes.
   Current State for Instance:10gM21.tvanraal-sun.us.oracle.com
   ```

5. Use opmnctl to check the status of all components, and if any components that you expect to be running are not running, then start them:
   ```
   % cd $ORACLE_HOME/opmn/bin
   % opmnctl status
   ```

Solution 2

Modify the last saved version of the opmn.xml file to allow DCM to run, and then perform a dcmctl resyncinstance command to update the configuration files with the version last saved to the repository.

To do this, perform the following steps:

1. Perform a opmnctl validate operation on the saved opmn.xml.bak file.
   ```
   % cd $ORACLE_HOME/opmn/conf
   % $ORACLE_HOME/opmn/bin/opmnctl validate opmn.xml.bak
   ```

2. If step one returns the following:
   ```
   opmnctl: opmn validation failed!
   ```

   Then, you need to correct the error in opmn.xml.bak to allow you to start the dcm daemon. First, backup your current version of opmn.xml.bak, in case you have problems, so you can revert to your backup copy. For example, use the following command:
   ```
   cp opmn.xml.bak opmn.xml.bak.orig
   ```

   Edit the $ORACLE_HOME/opmn/conf/opmn.xml.bak file and add the attribute status="disabled" for each <ias-component> element in opmn.xml.bak. For only the component dcm-daemon, set the status = "enabled".

   For example:
   ```
   <ias-component id="OC4J" status="disabled">
   <ias-component id="dcm-daemon" status="enabled" id-matching="true">
   ```

3. Replace the opmn.xml file with the updated version of opmn.xml.bak.
   ```
   % cd $ORACLE_HOME/opmn/conf
   % cp opmn.xml.bak opmn.xml
   ```

4. Start the DCM shell and run the resyncinstance command.
   ```
   % cd $ORACLE_HOME/dcm/bin
   % dcmctl shell
   dcmctl> resyncinstance -force
   All OC4J components are stopped during a resyncinstance -force.
   They will need to be restarted after the command completes.
   Current State for Instance:10gM21.tvanraal-sun.us.oracle.com
5. Use opmnctl to check the status of all components; if any components that you expect to be running are not running, start them:

```
% cd $ORACLE_HOME/opmn/bin
% opmnctl status
```

**Solution 3**

If Solution 1 and Solution 2 do not solve this problem, then both the opmn.xml and the opmn.xml.bak files have been corrupted. In this case, you need to manually create an opmn.xml file that enables you to start the dcm-daemon and then perform a dcmctl resyncinstance command to update the configuration files with the version last saved to the repository.

To create a version of opmn.xml that is sufficient for starting the dcm-daemon, perform the following steps:

1. Copy and paste the text shown in Example B–1 to $ORACLE_HOME/opmn/conf/opmn.xml.
2. Edit the port element such that the attributes local, remote, and request have the correct values for your environment. These values are shown as XXXX, YYYY, and ZZZZ in Example B–1.
3. Using this file, the OPMN Server should start up, which should allow the DCM Daemon to start so that you can restore the opmn.xml file from the DCM repository using the dcmctl resyncInstance command.

---

**Example B–1  Basic opmn.xml File For Emergency Startup - Replace Port Values Before Using**

```
<opmn xmlns="http://www.oracle.com/ias-instance">
<notification-server>
  <port local="XXXX" remote="YYYY" request="ZZZZ"/>
  <log-file path="$ORACLE_HOME/opmn/logs/ons.log" level="4"
    rotation-size="1500000"/>
  <ssl enabled="true" wallet-file="$ORACLE_HOME/opmn/conf/ssl.wlt/default"/>
</notification-server>
<process-manager>
  <log-file path="$ORACLE_HOME/opmn/logs/ipm.log" level="4"
    rotation-size="1500000"/>
  <process-modules>
    <module path="$ORACLE_HOME/opmn/lib/libopmndcmdaemon">
      <module-id id="DCMDaemon"/>
    </module>
  </process-modules>
  <ias-instance id="foo">
    <environment>
      <variable id="TMP" value="/tmp"/>
    </environment>
    <ias-component id="dcm-daemon" status="enabled" id-matching="true">
      <process-type id="dcm-daemon" module-id="DCMDaemon">
        <process-set id="dcm" numprocs="1">
          <module-data>
            <category id="start-parameters">
              <data id="java-parameters" value="-Xmx256m -Doracle.ias.sysmgmt.logging.loglevel=ERROR -Djava.net.preferIPv4Stack=true -Djava.io.tmpdir=$TMP"/>
              <data id="jar-file" value="$ORACLE_HOME/dcm/lib/dcm.jar"/>
              <data id="application-parameters" value="daemon -logdir $ORACLE_HOME/dcm/logs/daemon_logs -o $ORACLE_HOME"/>
            </category>
          </module-data>
        </process-set>
      </process-type>
    </ias-component>
  </ias-instance>
</opmn>
```
**B.1.2 DCM Daemon Cannot Start**

**Problem**
The DCM daemon does not start.

**Solution**
This can occur due to insufficient memory problem or an OPMN problem. You can investigate this problem as follows:

2. Review the error messages in the log files in `ORACLE_HOME/opmn/logs/*`.
3. Ensure that the ports OPMN uses are free of conflicts. These ports are listed in `ORACLE_HOME/opmn/conf/opmn.xml` under the `<notification-server>` element.

**B.1.3 DCM Daemon Does Not Start As Expected**

**Problem**
After a successful execution of the `opmnctl startall` command, the `opmnctl status` shows that `dcm-daemon` is down.

**Solution**
Start the `dcm-daemon` component explicitly, if necessary, by issuing this command:

```
opmnctl startproc ias-component=dcm-daemon
```

By default, the `dcm-daemon` component is not started by the `opmnctl startall` command. Instead, the `dcm-daemon` component is started during initialization by `dcmctl` or Oracle Enterprise Manager 10g Application Server Control Console.

Starting the `dcm-daemon` explicitly is required in some cases. For example, if you are performing a cluster-wide DCM operation, such as deploying an application to the cluster, then the `dcm-daemon` must be running on all of the OracleAS Instances in the cluster. A `dcmctl` command on the local OracleAS Instance will start the `dcm-daemon` on the local OracleAS Instance, but not in remote OracleAS Instances of the cluster.

**Note:** For more information, see: Section 1.2.1, "The Distributed Configuration Management Bootstrap Sequence" on page 1-5.
B.1.4 DCM Operations Fail

**Problem**
Occasionally, while using `dcmctl` commands with a file-based repository, unexpected messages or failures may occur.

**Solution 1**
Purge outdated records from the repository data tables using the `fileCleanUp` command on page A-15.

**Solution 2**
Restart all `dcmctl`, `dcm` daemons, and Application Server Control Console on all the OracleAS Instances in the farm. Restarting these processes may fix the problem, and will not result in a loss of configuration information.

B.1.5 Invalid username/password

**Problem**
`dcmctl` returns "invalid username/password; logon denied"

**Solution**
This could mean that DCM is having problems accessing Oracle Internet Directory or the DCM schema in the metadata repository. Refer to "Troubleshooting Oracle Internet Directory" on page B-12 for the steps for troubleshooting this problem.

B.1.6 BOM (Bill of Materials) Not Updated

**Problem**
The DCM log file contains the entry "BOM Not Updated". Usually, this error is accompanied by a detailed message resembling the following:

```
Error writing file path to file for plugin plugin name.
```

**Solution**
This message is logged when configuration files for the OracleAS Instance are not synchronized with the configuration information stored in the DCM repository. DCM was not able to update the configuration files due to a file system full condition, file permissions problem, or an operating system problem. Examine the following log files to find the reasons for which DCM could not write to the file system:

```
ORACLE_HOME/dcm/logs/dcmctl_logs/log.xml
ORACLE_HOME/dcm/logs/emd_logs/log.xml
ORACLE_HOME/j2ee/home/log/log.xml
```

Another way to pinpoint a problem writing to the file system is to log in to the operating system as the user that installed OracleAS, and attempt to write to a configuration file. The operating system will return an error message that describes the problem precisely.

After you resolve the file system problem, you can resynchronize the OracleAS Instance with this command:

```
ORACLE_HOME/dcm/bin/dcmctl resyncinstance
```
B.1.7 In Sync Status is False

Problem 1

dcmctl getState displays an In Sync status of "False: Configuration" for one or more components.

Solution 1

This indicates that the configuration files for the OracleAS Instance are out of sync with the configuration information stored in the DCM repository. This can happen if a previous operation was not able to complete properly. To resolve this problem, you need to restore the configuration information from the DCM repository to the OracleAS Instance as follows:

1. Stop Oracle Enterprise Manager 10g Application Server Control Console to ensure that no attempts are made to change the configuration while you are synchronizing it.

2. Resynchronize the OracleAS Instance with this command:

   \texttt{ORACLE_HOME/dcm/bin/dcmctl resyncinstance}

   This will usually fix the problem. If not, the \texttt{resyncinstance} operation will return error information that will help you determine what is wrong and how to proceed.

Problem 2

dcmctl updateconfig returns the message "The configuration files for this OracleAS Instance are inconsistent with the configuration stored in the repository."

Solution 2

This indicates that the configuration files for the OracleAS Instance are out of sync with the configuration information stored in the DCM repository. Stop and resynchronize the OracleAS Instance. After the configuration files for the OracleAS Instance are synchronized, you can make your original modifications. Be sure to run \texttt{dcmctl updateconfig} again after making the changes.

\textbf{See Also:} Problem 1 and Solution 1 in this section

Problem 3

A DCM configuration file has zero bytes after an operation.

Solution 3

A file size of zero bytes indicates there was a problem writing to the configuration file during a DCM operation. To resolve the problem:

1. Log on to the operating system as the user that installed the OracleAS Instance, and:
   a. Check the file permissions.
   b. Attempt to write to the configuration file, noting the operating system error message after any failure. Resolve the condition that caused the failure.
   c. Verify file system quotas.
d. Ensure that there are no problems with the operating system that are preventing writing to the configuration file.

2. Resolve any problems found.

3. Restore the file from the DCM repository with the resyncInstance command.

B.1.8 Unable to Connect to the Directory

**Problem**

dcmctl returns ADMN-100999 and the base exception is "Unable to connect to Directory."

**Solution**

This error occurs when DCM cannot access Oracle Internet Directory. DCM needs to connect to the directory in order to retrieve the passwords it uses to connect to the Database-based repository.

See if restarting Oracle Internet Directory solves the problem. If it does not, refer to "Troubleshooting Oracle Internet Directory" on page B-12.

B.1.9 Cannot Access the Infrastructure Database

**Problem**

dcmctl returns ADMN-202026 and the base exception is "Unable to connect to Directory."

**Solution**

This error occurs when DCM cannot access Oracle Internet Directory. DCM needs to connect to the directory in order to retrieve the passwords it uses to connect to the Database-based repository.

See if restarting Oracle Internet Directory solves the problem. If it does not, refer to "Troubleshooting Oracle Internet Directory" on page B-12.

B.1.10 The dcm.conf File is Unreadable

**Problem**

An error occurs when attempting login to Application Server Control Console or invoking a dcmctl command; permission is denied when attempting to read the dcm.conf file. This error occurs when a user other than the user who installed the OracleAS Instance invokes a dcmctl command, or when the file permissions on the dcm.conf file have been altered from the default set by the installer.

**Solution**

Change the ownership/permissions on the ORACLE_HOME/dcm/config/dcm.conf file to the permissions set by the installer when the OracleAS Instance was installed. In addition, ensure that the permissions are set correctly on ORACLE_HOME/opmn/conf/ons.conf and all files under ORACLE_HOME/dcm/logs/dcmctl_logs.
B.1.11 The dcm.conf or dcmCache.xml File is Empty

Problem
Occasionally, during configuration management operations, configuration files in are written out as zero length files. This is usually due to disk write problems, such as lack of space on the disk.

Solution
Copy the corresponding .bak file, dcm.conf.bak or dcmCache.xml.bak, to dcm.conf or dcmCache.xml respectively, and start the OracleAS Instance.

B.1.12 dcmctl getState Hangs

Problem
When the dcmctl getState -d command is executed, it returns the following and then stops:

Current state for Instance:myinstance.example.com
Component Type Up Status In Sync Status
=============================

Solution
Determine which process has actually stopped. It could be that a process associated with one of the related components (that is, OHS, OPMN, OC4J) has stopped.

1. Determine the PID of the dcm-daemon process, using the following command:

   ORACLE_HOME/opmn/bin/opmnctl status

2. Send a kill -3 signal to the dcm-daemon process with the following command:

   kill -3 PID

   This tells the JVM to dump an execution stack trace.

3. Open the ORACLE_HOME/opmn/logs/dcm-daemon~dcm-daemon~dcm~1 file and examine the execution stack trace to identify the process that caused DCM to stop.

B.1.13 getState Cannot Be Used With (9.0.2 or 9.0.3) Instances

Problem
Invoking the command dcmctl getstate-i instance name from an OracleAS 10g Release 2 (10.1.2) Oracle home results in an ADMN-604104 exception. This problem occurs when the instance name is a 9.0.2 or 9.0.3 OracleAS Instance.

Solution
When trying to retrieve the state of a 9.0.2 or 9.0.3 OracleAS Instance, invoke the dcmctl utility from a 9.0.2 or 9.0.3 Oracle home, and not from the OracleAS 10g Release 2 (10.1.2) Oracle home.
B.1.14 dcmctl joinFarm Produces Error

**Problem**
When the `dcmctl joinfarm` command is executed, it returns the following:

```
ADMN-202026
A problem has occurred accessing the OracleAS 10g infrastructure database.
Base Exception:
oracle.ias.repository.schema.SchemaException:DCM Repository is not Configured
Please refer to the base exception for resolution or call Oracle support.
```

**Solution**
If you are attempting to join a File-based repository, issue the command again, using the `-r` option to specify the repository ID for the File-based repository.

The `-r` option is required to specify the ID of a File-based repository. If you erroneously omit the `-r` option, `dcmctl` attempts to join an Infrastructure database farm with the repository ID you provided.

B.1.15 repositoryRelocated Command Causes Repository to be Unreachable

**Problem**
You executed `repositoryRelocated` and now the repository is unreachable.

The `repositoryRelocated` command is intended to be used in conjunction with the `importRepository` command. A OracleAS Farm can only be associated with one repository at a time. When the repository is imported to a new OracleAS Instance of the OracleAS Farm, the old OracleAS Instance must be notified, with the `repositoryRelocated` command, that it is no longer the repository host. If the repository was not imported to a new OracleAS Instance, then there is currently no OracleAS Instance managing the repository.

**Solution**
The solution depends on the circumstances under which the command was issued.

If the intent was to import an exported repository image to a new OracleAS Instance, simply issue the `importRepository` command. After the repository is imported to a new OracleAS Instance, all OracleAS Instances will be able to locate the new repository host.

If the command was run in error, and you wish to continue to use the current repository:

1. Stop the OracleAS Instance.
2. Navigate to the `ORACLE_HOME/dcm/config` directory.
3. Compare the files with the command:
   ```
   diff dcm.conf dcm.conf.bak
   ```
   The result should be:
   ```
   < repository_type=cache
   > repository_type=cache rep
   ```
4. Restore the dcm.conf file to the original version with the command:
   
   
   cp dcm.conf.bak dcm.conf

5. Start the OracleAS Instance.

   The repository is now available to the farm.

B.1.16 Unreachable IP Address

Problem

The following Application Server Control Console or dcmctl errors may occur if the configuration file ORACLE_HOME/dcm/config/dcmCache.xml includes a
<communication> element with a <coordinator> subelement that specifies an ip
attribute for an IP address that is not reachable.

The exception 202504, has occurred in the cache layer of the persistence manager. "Network port is in use.". Resolution: Please, refer to the base exception for the details.

And the following dcmctl error is also associated with this problem:

Local Stack:

oracle.ias.cache.group.GroupException: Network port is in use
at oracle.ias.cache.group.Transport.ssinit(Unknown Source)

at oracle.ias.cache.group.Transport.init(Unknown Source)

Solution

The IP address could become unreachable if the IP address for the machine changes, for example, in a DHCP environment.

Use the following procedure to workaround this problem:

1. Change the ip attribute specified in the <coordinator> element of the ORACLE_HOME/dcm/config/dcmCache.xml configuration file. There are several ip attributes and <coordinator> elements specified in this file; change the <coordinator> element with attribute "original=true". In this element change the ip attribute to the correct value for the machine. Use the hostname and nslslookup commands or similar commands to identify the correct IP address.

2. Change the VirtualHostName value in ORACLE_HOME/config/ias.properties to the corresponding correct hostname.

3. Run the command dcmctl resetHostInformation.

   This should correct the problem.

B.1.17 resyncInstance Command Does Not Restore Properly

Problem

A configuration change was not reverted as expected in the following sequence of steps:

1. Update of DCM-managed configuration file
2. Execute the updateConfig command
3. Manually remove change to configuration file
4. Execute the resyncinstance command
Solution
Make DCM aware of the removal of the configuration change in Step 3 by issuing the updateConfig command after Step 3. You can use the -force option to force DCM to resynchronize the OracleAS Instance whether it thinks it necessary or not:

```
ORACLE_HOME/dcm/bin/dcmctl resyncinstance -force
```

B.2 Diagnosing Problems with Components Related to DCM

Some problems reported by DCM have root causes in a related component or system (that is, OPMN, OHS, OC4J, Oracle Internet Directory, the metadata repository, or the file system). This section contains tips for recognizing, verifying, and fixing these types of problems. It contains the following topics:

Troubleshooting Oracle Process Manager and Notification Server
Troubleshooting Oracle Internet Directory
Troubleshooting the Metadata Repository

B.2.1 Troubleshooting Oracle Process Manager and Notification Server

A problem with starting or stopping could be due to OPMN, or to one of the components it is attempting to start and stop. The OPMN logs are the starting point for determining the source of the problem. The OPMN logs are located in the ORACLE_HOME/opmn/logs/. In addition to the main logs, opmn.log, ipm.log, and ons.log, there are log files for each component managed by OPMN.

B.2.2 Troubleshooting Oracle Internet Directory

If DCM returns an error referring to the directory, or a problem with a password, problems with Oracle Internet Directory may be the cause. This section provides instructions on how to resolve these types of problems.

B.2.2.1 How DCM Uses Oracle Internet Directory

DCM uses Oracle Internet Directory only if the OracleAS Instance is configured to use Identity Management and a metadata repository. In this case:

1. DCM logs in to Oracle Internet Directory using an internal, randomly generated password that is stored with the OracleAS Instance.
2. It obtains the metadata repository password from Oracle Internet Directory.
3. It uses this password to access the metadata repository.

B.2.2.2 How to Troubleshoot Oracle Internet Directory Problems

1. Determine the host and port the OracleAS Instance is using to access Oracle Internet Directory. Examine the ORACLE_HOME/config/ias.properties file in the Oracle home of the middle tier OracleAS Instance that is reporting the DCM failure. In the file, the Oracle Internet Directory host is OIDhost. The Oracle Internet Directory port is OIDport.
2. Ping the Oracle Internet Directory host from another computer to see if the network is running. If the ping is unsuccessful, then the network is stopped.
3. If the ping is successful, run the following command to see if you can access Oracle Internet Directory:

```
ORACLE_HOME/bin/ldapbind -p oid_port -h oid_host
```
4. If this command fails, then Oracle Internet Directory is stopped. Try to restart Oracle Internet Directory using Oracle Process Manager and Notification Server. If it does not start, then contact the Oracle Internet Directory administrator with a request to restart Oracle Internet Directory.

Use one of the following commands, on the OracleAS Instance that contains the OID component to start or restart Oracle Internet Directory:

- `opmnctl startall`
- `opmnctl startproc ias-component=OID`

5. Try the DCM command again. If it fails, continue with the next step.

6. Ensure that the metadata repository is running. DCM accesses the metadata repository differently, depending on installation type and configuration, as follows:

   - **For J2EE and Web Cache installations using Identity Management and a metadata repository:** The metadata repository the OracleAS Instance is using is the DBConnect field corresponding to ComponentNameDCM in the `ORACLE_HOME/config/iasschema.xml` file.

   - **For Portal and Wireless installations:** The metadata repository the OracleAS Instance is using is the InfrastructureDBCommonName field in the `ORACLE_HOME/config/ias.properties` file.

7. Ensure that the DCM schema password stored in Oracle Internet Directory matches the DCM schema password in the metadata repository.

   a. Obtain the DCM schema password from Oracle Internet Directory by following the instructions in Oracle Application Server Installation Guide, Section 7.5, “How to Determine the Password for the DCM Schema”.

   b. Try to log in to the DCM schema in the metadata repository using the password:

   ```sqlplus dcm/password@connectstring```

   If the passwords do not match, you have two options:

   **B.2.2.2.1 Option 1:** Reset the password in Oracle Internet Directory and the metadata repository using the Oracle Enterprise Manager 10g Application Server Control Console (Application Server Control Console):

   1. Log in to the Application Server Control Console and navigate to the Instance Home page for the OracleAS Instance that is reporting the problem.

   2. Click Infrastructure.

   3. Select the DCM schema.


   5. Click Change Password.

   6. Set the new password.

   7. Try the DCM command again.

   **See Also:** *Oracle Internet Directory Administrator’s Guide* for more information on troubleshooting Oracle Internet Directory.

   **B.2.2.2.2 Option 2:** Reset the DCM password in the metadata repository to match that which is stored in Oracle Internet Directory. To do this, log in to the metadata repository as the SYS user and issue the command:
sql> ALTER DCM IDENTIFIED BY password in oid

In the preceding command, password in oid is the DCM schema password retrieved from Oracle Internet Directory.

B.2.3 Troubleshooting the Metadata Repository

If DCM is reporting problems with the repository, and you are using the metadata repository (not a File-based repository), you can troubleshoot the repository as described in this section.

B.2.3.1 How DCM Uses the Metadata Repository

DCM stores configuration information in the DCM schema in the metadata repository.

B.2.3.2 How to Troubleshoot Problems with the Metadata Repository

1. Ensure that the metadata repository is running and that you can access it from your system. The metadata repository for the OracleAS Instance is listed in the DBConnect field in the ORACLE_HOME/config/iasschema.xml file.

2. If the error message states that an OracleAS Instance is inconsistent with the repository, issue the resyncInstance command:

   ORACLE_HOME/dcm/bin/dcmctl resyncinstance

3. Examine the log files for errors. The repository could be corrupted in some way, due to database or system errors. These will be captured in the following log files (in the following example, the asdb directory is named for the default SID):

   METADATA_REPOSITORY_ORACLE_HOME/admin/asdb/bdump
   METADATA_REPOSITORY_ORACLE_HOME/admin/asdb/udump

   If there is evidence of database failure or corruption, refer to database documentation for information on restoring and recovering the database.

B.2.4 Troubleshooting the File-based Repository

If DCM is reporting problems with a File-based repository, you can troubleshoot the repository as described in this section.

B.2.4.1 How DCM Uses the File-based Repository

DCM stores configuration information in ORACLE_HOME/dcm/repository.

B.2.4.2 How to Troubleshoot Problems with the File-based Repository

Most File-based repository problems are due to file system problems.

1. Ensure that:
   - The file system is not full.
   - File system usage is within quotas.
   - The operating system user that installed Oracle Application Server has write permission on the repository.

2. If the error message states that an OracleAS Instance is inconsistent with the repository, issue the resyncInstance command:

   ORACLE_HOME/dcm/bin/dcmctl resyncinstance
3. If the repository files contain errors or some or all have been removed, you can restore them as follows:

   If the OracleAS Instance is in a cluster:
   a. In the Oracle home of the OracleAS Instance you are trying to recover:
      \[\text{ORACLE_HOME/dcm/bin/dcmctl resyncinstance}\]

   If you have a backup of the Oracle home:
   a. Restore the repository files in \text{ORACLE_HOME/dcm/repository}.
   b. In the Oracle home of the OracleAS Instance you are trying to recover, issue this command:
      \[\text{ORACLE_HOME/dcm/bin/dcmctl resyncinstance}\]

B.3 Diagnosing DCM Problems

This section presents a step-by-step approach for diagnosing problems with DCM.

Step 1: Analyze the verbose and debug output from the dcmctl utility
When an error occurs, \text{dcmctl} usually provides a detailed error message and contextual information on a possible cause and solution. If an exception occurs, the \text{-d} option provides a stack trace. Sometimes this information may allow you to resolve the problem. The stack trace must be supplied if you contact Oracle support. By default, \text{dcmctl} utility sets the \text{-v} (verbose) option and the \text{-d} (debug) option on. Change these defaults with the \text{set} command.

To view the error message and stack trace that most recently occurred, use this command:

\[\text{ORACLE_HOME/dcm/bin/dcmctl getError}\]

Step 2: Examine the log files
If the information provided by the \text{dcmctl} verbose and debug output is not sufficient, try examining the log files. Often, they will contain error messages that will help you determine the problem. Table B–1 describes the log files related to DCM.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{ORACLE_HOME/dcm/logs/dcmctl_logs/log.xml}</td>
<td>DCM client log for \text{dcmctl} shell</td>
</tr>
<tr>
<td>\text{ORACLE_HOME/dcm/logs/emd_logs/log.xml}</td>
<td>DCM client log for Application Server Control Console</td>
</tr>
<tr>
<td>\text{ORACLE_HOME/dcm/logs/daemon_logs/log.xml}</td>
<td>DCM daemon log</td>
</tr>
<tr>
<td>\text{ORACLE_HOME/dcm/logs/busrule_logs/log.xml}</td>
<td>DCM business rule execution log</td>
</tr>
<tr>
<td>\text{ORACLE_HOME/opmn/logs/*}</td>
<td>Oracle Process Manager and Notification Server logs</td>
</tr>
</tbody>
</table>

Step 3: Review the Problems and Solutions section
If you have determined the error but are not sure how to proceed, try searching for the problem, or one similar, in "Problems and Solutions“ on page B-1.
Step 4: Determine whether the problem was caused by a related component

Many problems reported by DCM have root causes in the operation of a related component, such as OPMN, Oracle HTTP Server, Oracle Application Server Containers for J2EE, Oracle Internet Directory, and the metadata repository. Table B–2 provides tips on recognizing these types of problems and pointers to more information.

Table B–2 Problems with Related Technologies

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>May be due to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting and stopping</td>
<td>Oracle Process Manager and Notification Server, Oracle HTTP Server, or Oracle Application Server Containers for J2EE problems</td>
</tr>
<tr>
<td></td>
<td>See Also: Oracle Process Manager and Notification Server Administrator’s Guide</td>
</tr>
<tr>
<td></td>
<td>See Also: Oracle Application Server Containers for J2EE User’s Guide</td>
</tr>
<tr>
<td>Application deployment or clustering</td>
<td>Oracle Application Server Containers for J2EE problems</td>
</tr>
<tr>
<td>Directory or password problems</td>
<td>Oracle Internet Directory problems</td>
</tr>
<tr>
<td></td>
<td>See Also: Troubleshooting Oracle Internet Directory</td>
</tr>
<tr>
<td></td>
<td>See Also: Oracle Internet Directory Administrator’s Guide, Appendix I, Troubleshooting</td>
</tr>
<tr>
<td>Repository problems</td>
<td>Metadata Repository problems</td>
</tr>
<tr>
<td>(metadata repository)</td>
<td>See Also: Troubleshooting the Metadata Repository</td>
</tr>
<tr>
<td>Repository problems</td>
<td>File system problems</td>
</tr>
<tr>
<td>(File-based repository)</td>
<td>See Also: Troubleshooting the File-based Repository</td>
</tr>
</tbody>
</table>

Step 5: Get assistance from Oracle Support

If none of the preceding steps led to a solution, report the problem to Oracle Support. Have the following information ready:

- Steps to duplicate the problem, or a summary of the conditions and settings in effect when the problem occurred
- The verbose and debug output from the dcmctl command
- Log file entries

B.4 Need More Help?

You can find more solutions on Oracle MetaLink, http://metalink.oracle.com. If you do not find a solution for your problem, log a service request.

See Also:

addopmnlink command, A-6
ADMN-100999 error message, B-8
ADMN-202001 error message, B-2
ADMN-202026 error message, B-8, B-10
ADMN-604104 error message, B-9
ADMN-705002 error message, B-2
AJP port, 2-5
ApacheVirtualHost, 2-6
Application commands, A-5
Application Server Control, 2-12
applyarchiveto command, A-7
applyinstanceto command, A-8
archive
best practices, 3-6
commands, A-4
compatibility with instances and clusters, 3-2
creating with createArchive command, A-9
exporting and importing, 3-2
importing, 3-2
installed image, 3-1
leaveFarm and, A-25
naming, 3-4
setting number of versions, A-43
Archive commands, A-4
archive, staged configuration vs. clustering, 3-6
asynchronous operation, A-18
automatic archive settings, 3-4
automatic archiving, performance and, 3-5
removing, A-36
clustered environment, names of instances and applications, 2-8
command, previous, A-6
command-line options, OC4J, 2-5
command-specific options, A-2
comments, A-3
common configuration, how established, 2-5
created by installation process, A-37
In Sync Status and, A-42
removing, A-37
supported by DCM, A-31
components in clustered instance, 2-7
configrepositoryssl command, A-8
collection
common, 2-5
restoring, 3-5
saving, 3-5
storage location, 1-2
updating, A-45
with the applyarchiveto command, 3-5
collection file
out of sync, B-6, B-7
updateConfig and, B-11
zero bytes, B-7
Configuration Management commands, A-4
createarchive command, A-9
createcluster command, A-10
createcomponent command, A-10
D
daemon
PID, B-9
start failure, B-5
daemon_logs directory, 2-19
DBConnect, B-14
DCM daemon, stopping, 1-6
DCM schema password, B-6
dcmCache.xml file, 2-13, 2-15
unreachable IP address and, B-11
dcm.conf file, instance removal and, A-13
dcmctl, 2-11
command options, A-2
operating instance, A-2
Properties commands, A-5
shell, A-4
Shell commands, A-4
dcmctl_logs directory, 2-19
debug option, B-15
deinstallation, destroyInstance and, A-12
deployapplication command, A-11
destroyinstance command, A-12
destructive action, A-3
DHCP and resethostinformation command, B-11
disksize parameter, EAR files and, 2-15
disksize parameter, tuning, 2-15
DTD (Document Type Definition), A-47

E
EAR files and disk size parameter, 2-15
echo command, A-13
dmLogs directory, 2-19
environment variable, ORACLE_HOME, A-2
errors, CacheFullException or
RegionFullException, 2-15
exit command, A-13
export archive, 3-2
exportarchive command, A-14
exportrepository command, A-14

F
farm
creating with joinFarm command, A-23
definition of, 1-2
securing, A-8
file
dcmCache.xml, 2-13, 2-15
ias.properties, 2-16, 2-17
iasschema.xml, 2-16, 2-17, B-13, B-14
jazn.xml, 2-16, 2-18
ldap.ora, 2-16, 2-18
log.xml, 2-19
ons.conf, A-17
opmn.xml, 2-6, 2-7
sqlnet.ora, 2-16, 2-18
ssl.conf, 2-13
sysmgmtProperties.xml, 2-20
targets.xml, A-42, A-46
file system problems, B-14
file-based repository, 1-2
force option, A-3

G
getcomponenttype command, A-15, A-16
geterror command, A-16
getopmnport command, A-17
getrepositoryid command, A-17
getreturnstatus command, A-18
getstate command, A-18
global command options, A-2
Group, 2-6

H
help command, A-19
host name, restrictions for naming clusters, 2-8, A-11

I
ias.properties file, 2-16, 2-17
iasschema.xml file, 2-16, 2-17, B-13, B-14
importarchive command, A-20
importing archives, 3-2
importrepository command, A-20
In Sync Status, A-18
instance
cluster and, A-23
compatible, A-22
dcmctl effect on, A-2
hosted repository and, A-39
in non-managed cluster, A-33
in sync, 1-4
IP address and hostname, A-41
leaving a cluster, A-24, A-38
removing, A-12
restoring, 3-5
resynchronizing, A-42
standalone, 1-3
instance-specific
configuration, 2-21
parameters, 2-5
insufficient memory, B-5
Internet Inter-Orb Protocol (IIOP), enabling, A-11,
A-34
invalidated objects, cached, 2-15
IP address
restrictions for naming clusters, 2-8
unreachable, B-11
isclusterable command, A-21
iscompatible command, A-22

J
J2EE compliance, EAR file J2EE compliance, A-47
JAAS, 2-12
Java keystore, A-9
jazn.xml file, 2-16, 2-18
JDK keytool, 2-12
JMS port, 2-5
joincluster command, A-22
joinfarm command, A-23

K
keystore, 2-12, A-9
keytool, 2-12

L
LDAP configuration, removing instance and, 2-16,
2-18
ldapbind command, B-12
ldap.ora file, 2-16, 2-18
<table>
<thead>
<tr>
<th>Command/Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>leavecluster command</td>
<td>A-24</td>
</tr>
<tr>
<td>leavefarm command</td>
<td>A-25</td>
</tr>
<tr>
<td>leaveFarm, archives and</td>
<td>A-25</td>
</tr>
<tr>
<td>listapplications command</td>
<td>A-26</td>
</tr>
<tr>
<td>listarchives command</td>
<td>A-28</td>
</tr>
<tr>
<td>listclusters command</td>
<td>A-29</td>
</tr>
<tr>
<td>listcomponents command</td>
<td>A-30</td>
</tr>
<tr>
<td>listcomponenttypes command</td>
<td>A-31</td>
</tr>
<tr>
<td>Listen</td>
<td>2-6</td>
</tr>
<tr>
<td>listinstances command</td>
<td>A-31</td>
</tr>
<tr>
<td>listopmnlinks command</td>
<td>A-33</td>
</tr>
<tr>
<td>log file directory</td>
<td>2-19</td>
</tr>
<tr>
<td>log files</td>
<td>B-15</td>
</tr>
<tr>
<td>log.xml file</td>
<td>2-19</td>
</tr>
</tbody>
</table>

**M**

- manually configured OracleAS cluster, creating                        | A-6 |
- maxFileSize parameter                                                  | 2-19 |
- maxLogSize parameter                                                   | 2-19 |
- max-objects parameter, tuning                                          | 2-15 |
- memory, insufficient                                                    | B-5  |
- metadata repository, Distributed Configuration Management                | 1-2  |
- MissingLocalValuePolicy flag, OPMN                                      | 2-6  |
- instance-specific parameters and                                        | 2-6  |

**N**

- NameVirtualHost                                                         | 2-6  |
- naming applications and instances                                       | 2-8  |
- Non-managed OracleAS cluster commands                                   | A-5  |
- number of OC4J processes                                                | 2-5  |

**O**

- OC4J command-line options                                              | 2-5  |
- OC4J instance, removing                                                | A-37 |
- OC4J island definitions                                                | 2-5  |
- ONS port                                                               | 2-5  |
- remote                                                                | A-6  |
- A-17                                                                  |
- ons.conf file                                                          | A-17 |
- operating system of clustered instances                                | 2-7  |
- OpmnHostPort                                                          | 2-6  |
- opmn.xml file                                                          | 2-7  |
- options, dcmctl command                                                | A-2  |
- Oracle Application Server                                              | 2-12 |
- Infrastructure                                                         | 2-12 |
- Single Sign-On                                                         | 2-12 |
- Oracle home, restrictions for naming clusters                          | 2-8  |
- Oracle Internet Directory                                              | 2-12 |
- Oracle Internet Directory entry, removing instance and                 | 2-16 |
- 2-18                                                                  |
- ORACLE_DCM_JVM_ARGS variable                                           | A-47 |
- ORACLE_HOME environment variable                                       | A-2  |
- OracleAS JAAS Provider entries, removing instance and                  | 2-16 |
- 2-18                                                                  |
- override confirmation prompt                                           | A-3  |

**P**

- parent application                                                      | A-12 |
- password, DCM schema                                                   | B-6  |
- performance, automatic archiving and                                   | 3-5  |
- performance, file-based farm                                          | 2-15 |
- PerlBlob                                                               | 2-6  |
- Port                                                                   | 2-6  |
- port, DCM cache                                                        | A-40 |
- Process Management commands                                            | A-5  |
- processes                                                              | 2-6  |
- managing                                                               | A-5  |
- proxy settings                                                          | A-47 |

**Q**

- quit command                                                          | A-34 |

**R**

- redeploy, updateConfig and                                              | A-46 |
- redeployapplication command                                            | A-34 |
- removearchive command                                                  | A-35 |
- removecluster command                                                  | A-36 |
- removecomponent command                                                | A-37 |
- removeopmnlink command                                                 | A-38 |
- repeat previous shell command                                          | A-6  |
- repository host, standalone instance and                               | A-17 |
- repositoryrelocated command                                           | A-38 |
- resetdcmcacheport command                                             | A-40 |
- resetfiletransaction command                                          | A-41 |
- resethostinformation command                                         | A-41 |
- restoring configuration                                               | 3-5  |
- resyncinstance command                                                | A-42 |
- RMI port                                                               | 2-5  |
- root context                                                          | A-11 |
- A-34                                                                  |

**S**

- saving configuration                                                    | 3-5  |
- scope command options                                                  | A-2  |
- security, certificate-based                                            | A-8  |
- ServerName                                                            | 2-6  |
- set command                                                            | A-43 |
- setloglevel command                                                    | A-44 |
- shell                                                                  | A-4  |
- shell command                                                          | A-44 |
- sorting preference                                                     | A-43 |
- SQLnet.ora file                                                        | 2-16 |
- 2-18                                                                  |
- SSL                                                                     | 2-13 |
- .ssl.conf file                                                         | 2-13 |
- stateful OC4J applications, clustering and                             | 2-5  |
- synchronous operations                                                | A-18 |
- sysmgmtProperties.xml file                                            | 2-20 |

**T**

- targets.xml file                                                        | A-42 |
- A-46                                                                  |
- targets.xml file, instance removal and                                | A-13 |
- timeout value                                                         | A-43 |
undeployapplication command, A-47
unrecoverable command actions, A-3
Up Status, A-18
updateconfig command, A-45
User, 2-6
validateearfile command, A-47
verbose option, A-2, B-15
whichcluster command, A-48
whichfarm command, A-48
whichinstance command, A-49