Distributed Configuration Management

Reference Guide

10*g* (9.0.4)

Part No. B12052-02

March 2004

A reference guide for Distributed Configuration
Management command syntax, usage, and error resolution.

ORACLE®

Distributed Configuration Management Reference Guide 10*g* (9.0.4)

Part No. B12052-02

Contents

3   Troubleshooting

4   Archiving Configurations

Index

Send Us Your Comments

Distributed Configuration Management Reference Guide, 10g (9.0.4)

Part No. B12052-02

Oracle 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.

Preface

This reference 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 topologies or 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 brief 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 architecture and functionality.

Chapter 2, "dcmctl Commands"

This chapter contains an alphabetical listing of dcmctl commands. For each command, a description, type, syntax and example usage are provided.

Chapter 3, "Troubleshooting"

This chapter describes common problems and provides possible solutions.

Chapter 4, "Archiving Configurations"

This chapter describes the Distributed Configuration Management archiving capabilities, and discusses best practices for archiving.

Related Documents

These documents may also be of interest to the system administrator using Distributed Configuration Management:

- *Oracle Application Server 10g Concepts*

- *Oracle Application Server 10g Administrator's Guide*

- *Oracle Process Manager and Notification Server Administrator's Guide*

- *Oracle Application Server 10g High Availability Guide*

Conventions

The following conventions are also used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
...	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
monospace text	File names, path names, command names, code, URLs
monospace bold text	Typed user input.
monospace italic text	Variables in text or code.

1

Distributed Configuration Management Overview

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
- Distributed Configuration Management Functionality
- Distributed Configuration Management Best Practices

1.1 What is Distributed Configuration Management?

Distributed Configuration Management is a management framework that enables you to manage the configurations of multiple Oracle Application Server instances.

Distributed Configuration Management (DCM) features enable you to:

- Keep multiple configurations synchronized
- Archive and restore versions of configurations
- Export and import configurations between Oracle Application Server instances

DCM manages configuration information for the following Oracle Application Server components:

- Oracle HTTP Server
- Oracle Application Server Containers for J2EE (OC4J)
- OC4J Applications
- Oracle Process Manager and Notification Server (OPMN)
- Oracle Application Server Java Authentication and Authorization Service (JAZN)

1.2 Distributed Configuration Management Architecture

Distributed Configuration Management consists of clients, a daemon, and a metadata repository. Figure 1–1 shows the relationship between these and other Oracle Application Server components.

Figure 1–1 Distributed Configuration Management Architecture
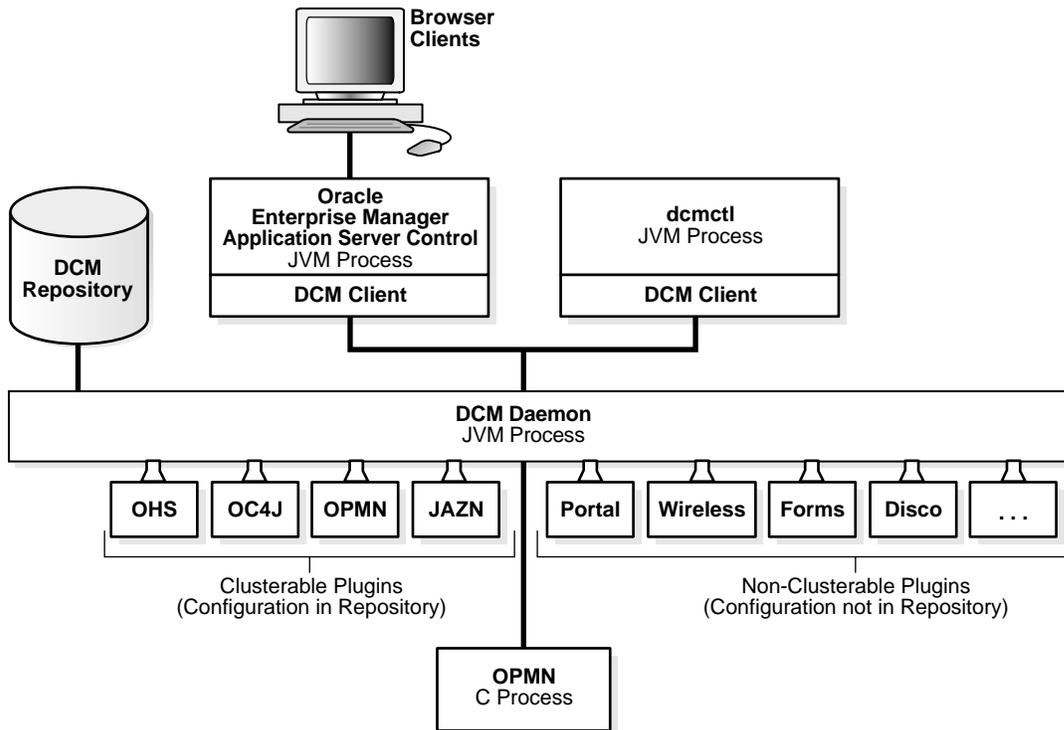
The Oracle Enterprise Manager Application Server Control servlet and `dcmctl` contain the Distributed Configuration Management client JAR file. The Distributed Configuration Management bootstrap sequence is:

1. The Distributed Configuration Management 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 Distributed Configuration Management client checks to determine whether the Distributed Configuration Management daemon is running.

   a. If not, it starts the daemon.

   b. If yes, it discovers and uses it.

3. The Distributed Configuration Management daemon checks the configuration file version of Oracle HTTP Server, Oracle Application Server Containers for J2EE, Oracle Process Manager and Notification Server, and Java Authentication and Authorization Service (using the configurable plug-ins shown in Figure 1–1.

4. The Distributed Configuration Management daemon updates the configuration file versions, if required.

5. The Distributed Configuration Management daemon restarts Oracle Process Manager and Notification Server, if required.

1.3 Distributed Configuration Management Functionality

This section covers the following topics:

- Understanding the Distributed Configuration Management Tools and Scope
- Understanding the Distributed Configuration Management Metadata Repository

1.3.1 Understanding the Distributed Configuration Management Tools and Scope

Distributed Configuration Management enables you to:

- Manage clusters and farms of Oracle Application Server instances.
- Manage the configuration of individual components, such as OC4J instances, Oracle HTTP Server, OPMN, or the Java Authentication and Authorization Service (JAZN).
- Perform cluster-wide Oracle Application Server Containers for J2EE application deployment.
- Manage versions of configurations with archive, save and restore, and import and export functions.

The `dcmctl` command is the Distributed Configuration Management command-line utility. You can use `dcmctl` to manage configurations and to deploy applications. Chapter 2, "dcmctl Commands", contains instructions on using `dcmctl` and descriptions of the available `dcmctl` commands.

All configuration and topology data is stored in the Distributed Configuration Management metadata repository, which is optionally stored as part of the Oracle Application Server Metadata Repository. The Distributed Configuration Management metadata repository is a distinct metadata repository that is not dependent on the Oracle Application Server Metadata Repository.

> **See Also:** *Oracle Application Server 10g High Availability Guide* for information on working with Oracle Application Server instances, farms, and clusters.

1.3.2 Understanding the Distributed Configuration Management Metadata Repository

The Distributed Configuration Management metadata repository contains the following:

- Configuration files for Oracle HTTP Server, Oracle Application Server Containers for J2EE, Oracle Process Manager and Notification Server, and Java Authentication and Authorization Service.
- Topology information describing a farm and the instances and clusters that are part of the farm
- Deployed J2EE applications

1.3.2.1 Types of Distributed Configuration Management Repositories

An Oracle Application Server farm is a collection of instances that share the same configuration management metadata repository. A farm can use either a file-based repository or database repository.

There are three types of metadata repository configuration: File-based (standalone instance), File-based (with repository host) and Database:

- File-based repository (standalone instance) – Every instance includes a local file-based repository. When an instance is in standalone mode, this repository stores the configuration metadata for the instance. When the instance is associated with a farm, either database or file-based, and the instance is not the repository host, the local file-based repository contains the Bill of Materials (BOM) that DCM uses to validate that the instance is in sync with the configuration metadata in the repository.

- File-based repository (with repository host) – When an instance is defined as the repository host for a file-based farm, the file-based repository for the instance contains the configuration metadata for all instances in the farm.

- A Database repository – comprised of Distributed Configuration Management schema. Storing the metadata repository in a database may be useful as part of a site's high availability and backup strategy. Using a database repository, the database serves as the repository host.

For all three types of metadata repository: database repository, file-based repository in standalone mode, or file-based repository host mode, an instance always has a local file based repository. In cases where the instance is not included in a farm this is the sole storage for the configuration metadata for the instance.

You can access each configuration management repository using either Oracle Enterprise Manager Application Server Control (Application Server Control) or the `dcmctl` utility.

> **See Also:** *Oracle Application Server 10g High Availability Guide* for information on setting up the repository and working with clusters.

1.3.2.2 Understanding Synchronization of the Metadata Repository and the Instance

The DCM repository is viewed as the definitive source for configuration information that DCM manages. If there is a difference in the configuration stored in the repository and what is in the associated ORACLE_HOME file system for an instance, the configuration in the file system is updated with the configuration in the repository. When the DCM repository and the file system configuration information have no differences, the configuration is synchronized or "In Sync".

DCM attempts to resynchronize the members of a cluster automatically and immediately after a configuration change. If an instance in a cluster is not available, the resynchronization occurs the next time the DCM daemon on the instance is started. The DCM daemon is started when an application server restarts, or manually using the `dcmctl start -admin` command.

In a farm or in a cluster, when you make a configuration change, DCM attempts to assure that a configuration change will be successful by applying the change to the local instance before attempting to propagate the change to other instances. If the local configuration change fails, its affects are automatically rolled back. There are cases where a configuration change may be successful on one instance but fail on other instances in the farm or cluster. There are many reasons why this could occur, including issues related to disk space, file system security, or connectivity from the Instance to a dependent services (for example, OID, or the database). In these cases an instance may be marked with the "In Sync" status set to `False`.

When the "In Sync" status is set to `False`, this is noted, with details in the DCM log. In this case, when the problem associated with this condition is resolved, you should

resynchronize the instance using the `dcmctl resyncinstance` command. This command instructs DCM to copy the configuration stored in the repository for an instance to the file system for the instance (see resyncInstance).

The `updateConfig` command is a special kind of 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. Read the guidelines for using the `updateConfig` command carefully before using this command (see updateConfig).

> **See Also:**
>
> "'In Sync' Status is False" on page 3-7

1.4 Distributed Configuration Management Best Practices

This section covers the following topics:

- Naming OC4J Instances and J2EE Applications
- Using Distributed Configuration Management Archiving
- Managing Oracle Application Server Clusters with DCM

1.4.1 Naming OC4J Instances and J2EE Applications

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 cluster.

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

The reason for this restriction is the following. In order to deliver clustering, archival and cloning features, DCM must scan configuration files for occurrences of the host name, Oracle home and IP Address. When DCM finds occurrences of these values, it replaces these 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 specific for that installation. Therefore, if OC4J Instance names or J2EE application names contain host names, Oracle homes or IP Addresses, they will be inappropriately substituted when applied to other installations.

1.4.2 Using Distributed Configuration Management Archiving

The DCM archive feature provides a convenient and easy means of managing "snapshots" of the DCM managed portions of Oracle Application Server system configuration. Archives are useful for staging changes, recovering from errors, and to provision DCM managed configuration information associated with one Oracle Application Server instance to another.

DCM managed system configuration includes configuration for a farm, clusters, Oracle HTTP Server, OPMN, OC4J, and Oracle Application Server Java Authentication and Authorization Service. For OC4J, in addition to configuration information related to the container itself, DCM manages all deployed J2EE applications.

Note that it is often not expensive to take an archive in terms of disk space. Within an Oracle Application Server instance, there are many managed objects (including configuration files and EAR or WAR files). When an archive is taken only one copy of any given version of a managed object is saved in the repository.

You can use archives to restore the state of an Oracle Application Server instance or 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 "rollback" undesired administrative changes. The number of automatic archives that are saved is configurable.

Administrators also have the option to explicitly create archives to satisfy the 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.

The state of an Oracle Application Server instance can be rolled back in place to the state of any available archive. An archive can also be applied to another Oracle Application Server Instance in the same farm, or exported and imported to another farm and then applied to an instance in that farm.

If you use DCM clusters, DCM assures that any change to the configuration is automatically distributed to all members of the cluster. As an alternative to using clusters, an archive of a staged configuration can be applied manually to non-clustered instances in a farm.

A hybrid staging solution is to first stage and test changes to a non-clustered instance, archive the changes, and finally apply the archive to a DCM cluster. These changes are then automatically propagated to all members of the cluster.

Note that after applying an archive to an instance other than the one from which it was created, some instance specific configuration data may have to be modified. DCM automates this for IP addresses and hostnames.

> **See Also:** Chapter 4, "Archiving Configurations" on page 4-1

1.4.3 Managing Oracle Application Server Clusters with DCM

Oracle Application Server instances grouped in a cluster can be managed using Oracle Enterprise Manager Application Server Control (Application Server Control) or `dcmctl` from a single point of administration on any instance in the cluster. It is recommended that one instance be used as the administrative point for the entire cluster at any point in time.

When changing instance specific configuration, for example port numbers, host names or virtual hosts, on a particular instance in the cluster, you must ensure that there are no other administrative changes are being made concurrently in the cluster to avoid conflicting changes to configuration resulting in an unusable configuration.

Concurrent administration within a cluster is strongly discouraged. If multiple administrative operations are issued at the same time in a cluster, this can lead to errors and associated confusing error messages. For example, a concurrent attempt to change the configuration of an instance being deleted really does not make sense. Specifying a single instance in a cluster as the management point ensures that operations are executed in the correct order and are properly serialized.

Archiving before any administrative operation is also recommended. Auto-archiving automatically detects that the Oracle Application Server instance is associated with a cluster and auto-archives are created of the cluster configuration prior to administrative operations.

See Also: "Using Distributed Configuration Management Archiving" on page 1-5

2

dcmctl Commands

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

2.1 Notes for Using dcmctl Commands

All `dcmctl` commands have this syntax:

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

Before you use the `dcmctl` utility, note the following:

- Do not run updateConfig concurrently with any other `dcmctl` commands or Oracle Enterprise Manager Application Server Control (Application Server Control) configuration operations from multiple Oracle Application Server instances in a farm or cluster (see updateConfig for details).

- Oracle recommends that Oracle Application Server Clusters using a file based repository contain four (4) or less than four instances.

- Oracle Application Server supports heterogeneous instances as part of the same farm. For example, an instance running on Solaris Operating System, an instance running on a Linux system, and an instance running on an HP-UX system can reside in the same farm. Oracle Application Server instances that you want to be part of a cluster must be installed on identical operating systems.

- You must log in to the operating system with the user name that was used to install Oracle Application Server in order to use `dcmctl`.

- When using a file based farm, you may need to refresh or restart Application Server Control after issuing the following `dcmctl` commands:

  - joinCluster
  - joinFarm
  - leaveCluster
  - leaveFarm

- Ensure that you issue `dcmctl` commands in the Oracle home of the instance you wish to manage. `dcmctl` commands operate on the instance in which the `dcmctl` executable is located. The value of the `ORACLE_HOME` environment variable does not determine the instance on which `dcmctl` operates.

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

- Instance, component, and cluster names are case-sensitive.

- If you use dcmctl set commands while multiple DCM clients are active, the clients will not act upon the changed values until they are restarted. You must restart the dcmctl and Enterprise Manager clients after using dcmctl set when multiple DCM clients are active.

2.1.1 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 2–1).

- Scope: These options indicate the scope of a command. The scope can be an application, application server cluster, component, type of component, or application server instance (see Table 2–2).

- 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 2–3).

Table 2–1   dcmctl Global Options

Option	Description
−d	Prints the stack trace if an exception occurs when the command is executed. By default, `dcmctl` executes with this option on. You can change this behavior with the set command.
−l *directory*	Saves the DCM client error log file `log.xml` in the named *directory*.
	The directory can be a full path name or a path name relative to the current directory.
	Default value: `ORACLE_HOME/dcm/logs/dcmctl_logs`
−v	Prints the long (verbose) version of state and error messages. Setting verbose to `off` is recommended when using scripts or using the `dcmctl` shell, since the brief messages are easier to parse.
	By default, `dcmctl` executes with this option on. You can change this behavior with the set command.

Table 2–2   dcmctl Scope Options

Option	Description
−a *app_name*	Applies the command to the named application *app_name*, or designates the name of an application during initial deployment.
−admin	Applies the command to the DCM daemon. For example:
	`start -admin` `stop -admin`
−cl *cluster_name*	Applies the command to the named application server cluster, *cluster_name*, or designates the name of a cluster during creation.
−co *comp_name*	Applies the command to the named component, *comp_name*, or designates the name of a component during creation
−ct *type*	Applies the command to components of the named component *type*.
	Component type can be of type: `ohs`, `oc4j`, `opmn`, or `jazn`.
−i *inst_name*	Applies the command to the named instance, *inst_name*.
−r *repository_info*	Applies the command to the repository name or port specified with *repository_info*.

Table 2–3   dcmctl Command-Specific Options

Option	Description
-arch	Specifies an archive name. Use with archive commands. For example: `dcmctl createArchive -arch myArchive`
-c	Creates a comment. Use with archive commands and 'import' and 'export' configuration management commands.
-force	Performs an operation without performing checks or issuing a confirmation. This option should be used judiciously, since in some cases, it can result in lengthy processing, and in others, it can perform a destructive action that might be unrecoverable.
-sort	Sorts listed results by name. Use with all 'list' commands. For example: `dcmctl listComponents -sort` You can make sorting persistent with the set command.
-src	Indicates the source of the archive. Use with archive commands. For example, to apply an archive to the current instance: `dcmctl applyArchiveTo -src myArchive`

2.1.1.1 Notes on 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 instance.

- If -cl is supplied, the command applies to all instances in the cluster.

- If -ct is supplied with -cl or -i, then the command applies to the component type within the cluster, or the component type within the instance within the cluster.

- If -co is supplied with -ct, -cl, -i then the command applies to the component within the cluster, or the component type within the instance within the cluster.

- If -a is supplied with any of the preceding options, that particular application within the component, component type, instance, or cluster is used.

2.1.2 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.

```
dcmctl shell

dcmctl> createcluster -cl testcluster

dcmctl> joincluster -cl testcluster

dcmctl> createcomponent -ct oc4j -co component1

dcmctl> start -co component1
```

```
dcmctl> deployapplication -f /stage/apps/app1.ear -a app1 -co
component1

dcmctl> start -cl testcluster

dcmctl> getstate

dcmctl> exit
```

2.2 Types of dcmctl Commands

This section describes types of dcmctl commands and their uses.

Configuration Management: Use these commands to create and associate configuration elements (clusters, instances, components)

Table 2–4   Configuration Management Commands

Command			
applyClusterTo	applyInstanceTo	createCluster	createComponent
configRepositorySSL	destroyInstance	exportRepository	getComponentType
getRepositoryId	importRepository	isClusterable	isCompatible
joinCluster	joinFarm	leaveCluster	leaveFarm
listClusters	listComponents	listInstances	removeCluster
removeComponent	repositoryRelocated	resetDCMCachePort	resetFileTransaction
resetHostInformation	restoreInstance	resyncInstance	saveInstance
updateConfig	whichCluster	whichFarm	whichInstance

dcmctl Shell: Use these commands with the dcmctl shell.

Table 2–5   dcmctl Shell Commands

Command			
!!	echo	exit	quit
shell	setLogLevel		

Archive: Use these commands to create archives of configurations.

Table 2–6   Archive Commands

Command			
applyArchiveTo	createArchive	exportArchive	importArchive
listArchives	removeArchive		

Application: Use these commands to deploy and manage applications.

Table 2–7   Application Commands

Command			
deployApplication	listApplications	redeployApplication	undeployApplication

Table 2–7 (Cont.) Application Commands

Command
validateEarFile

dcmctl Properties: Use these commands to administer the dcmctl utility.

Table 2–8   dcmctl Properties Commands

Command			
getError	getReturnStatus	getState	help
listComponentTypes	set		

Non-managed Clusters: Use these commands to work with non-managed clusters.

Table 2–9   Non-Managed Cluster Commands

Command			
addOPMNLink	getOPMNPort	listOPMNLinks	removeOPMNLink

Process Management: These commands allow you to start and stop processes within clusters and instances. These commands are deprecated in Oracle Application Server 10*g*. Use `opmnctl` to manage processes in Oracle Application Server 10*g*.

Table 2–10   Process Management Commands

Command			
restart	shutdown	start	stop

2.3 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.

2.3.1 !!

Repeats the previous command.

Type
Shell

Syntax
!!

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

Example
!!

2.3.2 addOPMNLink

Creates a non-managed Oracle Application Server cluster.

Type
Configuration Management

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

Description
You can use this command to create a non-managed Oracle Application Server cluster that includes the local application server instance and the instances specified as arguments.

- You must run this command in the Oracle home of each instance you would like to put into the cluster, using the rest of the instances as arguments.

- All instances must be J2EE and Web Cache instances and the instances must not be part of a farm (associated with a repository); otherwise, the command will fail.

- If you would like to change the ONS remote port for an instance in a cluster, you must remove the instance from the cluster using removeOPMNLink, change the remote port, and add it to the cluster again using addOPMNLink. You must repeat the command in every Oracle home.

- If you create a cluster and then want to add another instance to the cluster, you must run the command again in all Oracle homes (essentially creating a new cluster with the added 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
```

2.3.3 applyArchiveTo

Applies an archived configuration to an instance or 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 which is generic to any instance (cluster-wide information).

- Information that is specific to a particular 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 instance that it was created for, both the cluster-wide information and instance specific information is restored to the instance. When the archive is applied to a cluster or to a different instance than the one it was created for, only the cluster-wide information is restored, the existing instance specific information is not changed.

Example

```
dcmctl applyarchiveto –src archive1 –i instance1
```

2.3.4 applyClusterTo

Applies the configuration of a cluster to an instance or cluster.

Type

Configuration Management

Syntax

applyClusterTo -src clusterName [-cl clusterName | -i instanceName]

Description

The configuration of the named cluster (as specified by the `-src` option) is applied to the named instance or cluster. The named source cluster is not affected.

Example
```
dcmctl applyclusterto -src cluster1 -i instance1
```

2.3.5 applyInstanceTo

Applies the configuration of the named instance to another instance or cluster.

Type
Configuration Management

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

Description
The configuration of the named instance is applied to the named instance or cluster. If no instance or cluster is specified, then the configuration of the named instance is applied to the current instance. The named source instance is not affected. The command will fail if the current instance and the named instance are the same (you cannot apply the configuration of an instance to itself).

Example
```
dcmctl applyinstanceto -src instance1
```

2.3.6 configRepositorySSL

Specifies the location and password of the keystore used to secure the farm.

For more information on using this command, see the *Oracle Application Server 10g High Availability Guide.*

Type
Configuration Management

Syntax
configRepositorySSL -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 instance in the farm must have a Java keystore. It 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 instance in the 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 instance in the farm to set the value of `<useSSL> true </useSSL>`. You must restart all DCM daemons and clients in the farm for the security change to take effect. The security setting must be consistent across all instances in the farm, or they will not communicate properly.

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

2.3.7 createArchive

Creates an archive of the named cluster or instance.

Type
Archive

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

Description
An archive is created of the named instance or cluster. If you don't specify a cluster or instance, the current instance is archived.

The difference is as follows.

Notes for using `createArchive` with Oracle Application Server clusters:

- Cluster-wide archives, created with `createArchive -cl`, contain only cluster-specific information and do not contain any information specific to the 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 instance where `createArchive` runs.

Example
```
createArchive -arch myInstance -comment "my favorite
configuration"
```

2.3.8 createCluster

Creates a managed Oracle Application Server cluster.

> **Note:** Oracle recommends that Oracle Application Server Clusters using a file based repository contain four (4) or less than four instances.

Type
Configuration Management

Syntax
createCluster -cl cluster_name

Description
A managed cluster is created.

Notes for using `createCluster`:

- When creating a 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 instance that belongs to a farm (that is, is associated with a metadata repository). The cluster will be created in that farm.

- The cluster has no members when created. You can add members using joinCluster.

- You can create an unlimited number of clusters.

Example

```
dcmctl createCluster -cl cluster1
```

2.3.9 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 application server instance. You cannot specify another instance with the -i option; the command operates locally. Note that OC4J is currently the only component type allowed for this command.

> **Note:** When creating a component with the createComponent command, use only the following characters in the *component_name* argument supplied with the -co option:
>
> ```
> abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_-
> ```

Example

```
dcmctl createComponent -ct oc4j -co OC4J_myapps
```

2.3.10 deployApplication

Deploys a J2EE application.

Type

Application

Syntax

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

where

- 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

- –rc 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.

- –pa 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 local application server instance.

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
```

2.3.11 destroyInstance

Removes an instance from the DCM repository.

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

Type

Configuration Management

Syntax

destroyInstance -i instance_name

Description

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

If the command is executed in the 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
```

2.3.12 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"
```

2.3.13 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
```

2.3.14 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 4-7.

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

2.3.15 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:

■ Usually if you use the exportRepository command, you also use importRepository on another instance. Before running importRepository , stop all DCM daemons in the instances that are part of the farm where you run importRepository. Use the following command at each instance in the farm to stop DCM daemons:

```
dcmctl shell
dcmctl> shutdown
```

Example
This example assumes that you have two 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,

```
dcmctl shell
dcmctl> exportrepository -f /export/repository_save_file
dcmctl> shutdown
```

If there are more than two instances, issue the shutdown command on all the other instances.

On instance2, shutdown and import the saved repository,

```
dcmctl shell
dcmctl> shutdown
dcmctl> importrepository -f /export/repository_save_file
```

On instance1,

```
dcmctl> repositoryrelocated
```

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

On instance1,

```
dcmctl> start -admin
```

When this command completes, on instance2 issue the command,

```
dcmctl> start -admin
```

If there are more than two instances, issue the following command sequentially on all of the other instances:

```
dcmctl> start -admin
```

2.3.16 getComponentType

Returns the type of the component in the local instance or specified 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 application server instance. You can use the $-i$ option to specify a different instance.

Example
To obtain the type of the `home` component in the local 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
```

2.3.17 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 most recently occurred:

```
dcmctl getError
```

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

```
dcmctl getError 906025
dcmctl getError ADMN-906025
```

2.3.18 getOPMNPort

Returns the hostname and ONS remote port.

Type
Non-managed Cluster

Syntax
```
getOPMNPort
```

Description
This command returns the hostname and the ONS remote port for the local application server instance. It retrieves this information from the ons.conf file.

Example
```
dcmctl getopmnport
```

```
myhost.example.com:6200
```

2.3.19 getRepositoryId

Returns the repository ID of a File based repository.

Type
Configuration Management

Syntax
getRepositoryId

Description
Returns the File based repository identifier of the farm to which the instance belongs. If the instance is a standalone instance, this command returns the repository identifier for the instance. If the standalone 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.

See the *Oracle Application Server 10g High Availability Guide* for a complete description of using `getRepositoryId` and `joinFarm -r`.

Example
```
dcmctl getrepositoryid
```

2.3.20 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 the 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.
```

2.3.21 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

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 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 instance, cluster, or component.

Example

To get the state of the local 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 Oracle Application Server cluster:

```
dcmctl getState -cl cluster1
```

2.3.22 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
```

2.3.23 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 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"
```

2.3.24 importRepository

Moves a File based repository.

Type

Configuration Management

Syntax

importRepository -f file_name [-force]

Description

Moves a File based repository from one instance to another, based on a saved file from the exportRepository command. The repository may be restored to any instance in the farm. If the current 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 repository host is still a member of the farm, then the repositoryRelocated command must be run in the repository host instance to notify it that it is no longer the host.

Note for using importRepository:

■   Before running importRepository, stop all DCM daemons in the instances that are part of the farm where you are running importRepository. Use the following command at each instance in the farm to stop DCM daemons:

```
dcmctl shell
dcmctl> shutdown
```

Example

This example assumes that you have two 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,

```
dcmctl shell
dcmctl> exportrepository -f /export/repository_save_file
dcmctl> shutdown
```

If there are more than two instances, issue the shutdown command on all the other instances.

On instance2, shutdown and then import the saved repository,

```
dcmctl shell
dcmctl> shutdown
dcmctl> importrepository -f /export/repository_save_file
```

On instance1,

```
dcmctl> repositoryrelocated
```

After the repositoryrelocated command completes, start the dcm daemons, as follows:

On instance1,

```
dcmctl> start -admin
```

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

```
dcmctl> start -admin
```

If there are more than two instances, issue the following command sequentially on all of the other instances:

```
dcmctl> start -admin
```

2.3.25 isClusterable

Identifies whether an application server instance can become a member of a managed Oracle Application Server cluster.

Notes for using Oracle Application Server clusters:

1. Oracle Application Server supports heterogeneous instances as part of the same farm. For example, an instance running on Solaris Operating System, an instance running on a Linux system, and an instance running on an HP-UX system can reside in the same farm. Oracle Application Server instances that you want to be part of a cluster must be installed on identical operating systems

2. Oracle recommends that Oracle Application Server Clusters using a file based repository contain four (4) or less than four instances.

Type
Configuration Management

Syntax
isClusterable [-i instance_name | -arch archive_name]

Description
Identifies whether an application server instance is eligible to become a member of a managed Oracle Application Server cluster. By default, this command uses the local instance. You can use the $-i$ option to specify a different instance. In order for an instance to be eligible, all components in the instance must be clusterable.

If the instance is ineligible, and the verbose option is enabled, the reason is included with the false answer returned by the command.

Example
```
dcmctl isClusterable
```

2.3.26 isCompatible

Identifies whether an application server instance is compatible with other members of a managed Oracle Application Server cluster.

Type
Configuration Management

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

Description

Identifies whether an application server instance is compatible with other members of a managed Oracle Application Server cluster. By default, this command uses the local instance. You can use the -i option to specify a different instance. An instance is compatible if it has the same components configured and is of the same version.

Example

```
dcmctl isCompatible -cl cluster1
```

2.3.27 joinCluster

Adds an Oracle Application Server instance to the named managed cluster.

Notes for using Oracle Application Server clusters:

1. Oracle Application Server supports heterogeneous instances as part of the same farm. For example, an instance running on Solaris Operating System, an instance running on a Linux system, and an instance running on an HP-UX system can reside in the same farm.

2. Oracle Application Server instances that you want to be part of a cluster must be installed on identical operating systems

3. Oracle recommends that Oracle Application Server Clusters using a file based repository contain four (4) or less than four instances.

4. If you are using Oracle Enterprise Manager Application Server Control, then, after issuing joinCluster command, you must stop and then start Oracle Enterprise Manager Application Server Control using the commands:

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

Type

Configuration Management

Syntax

joinCluster -cl cluster_name [-i instance_name]

Description

Adds an application server instance to the managed Oracle Application Server cluster specified with the -cl option. By default, this command uses the local instance. You can specify a different instance with the -i option. The instance must be a member of the same farm as the cluster. There is no limit to the number of instances you can add to a cluster. An instance is stopped after being added to a cluster, so you must manually start it.

Example

To add the local application server instance to cluster1 and restart it:

```
dcmctl joinCluster -cl cluster1
```

```
dcmctl start
```

To add instance1 to cluster1 and restart it:

```
dcmctl joinCluster -cl cluster1 -i instance1
```

```
dcmctl start -i instance1
```

2.3.28 joinFarm

Associates an instance with a database repository or a File based repository.

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

1. Oracle Application Server supports heterogeneous instances as part of the same farm. For example, an instance running on Solaris Operating System, an instance running on a Linux system, and an instance running on an HP-UX system can reside in the same farm.

2. Oracle Application Server instances that you want to be part of a cluster must be installed on identical operating systems.

3. Oracle recommends that Oracle Application Server Clusters using a file based repository contain four (4) or less than four instances.

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

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

> **Caution:** Using `joinFarm`, when an instance joins a farm all archives for the instance are removed. If you want to preserve the archives on the instance that is joining the farm, export each of the archives with the exportArchive command prior to using the `joinfarm` command.

Type
Configuration Management

Syntax
joinFarm [-r repository_ID]

Description
With the `-r` option and a repository ID, it associates an instance with the named file-based repository. The `repositoryId` is a hostname and port.

If the instance was originally associated with a database repository, using infrastructure database information, you can issue this command without arguments to reassociate the instance with the original database.

You can obtain the file-based repository ID by issuing the getRepositoryId command on any instance that is a member of the farm associated with the repository. (To establish a distributed file-based repository, execute getRepositoryId on the instance that will host the repository, then issue the `joinFarm` command with the identifier returned by `getRepositoryId`). See the *Oracle Application Server 10g High Availability Guide* for a complete description of using `getRepositoryId` and `joinFarm -r`.

Example

To reassociate the instance with a repository:

1. Get the repository ID from the instance that hosts the repository (the instance could be file-based or stored in a database).

    ```
    % dcmctl getRepositoryID
    ```

2. Using the repository ID that you obtain in the previous step, issue the command:

    ```
    % dcmctl joinFarm -r repository_ID
    ```

2.3.29 leaveCluster

Removes an instance from a managed Oracle Application Server cluster.

Type

Configuration Management

Syntax

leaveCluster [-i instance_name]

Description

Removes an application server instance from its managed Oracle Application Server cluster. By default, this command uses the local instance. You can specify a different instance with the –i option. The instance being removed is stopped, so you must restart it after using this command.

Note the following when using leaveCluster:

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

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

Example

To remove the local instance from the cluster:

```
dcmctl leaveCluster
```

To remove instance1 from its cluster:

```
dcmctl leaveCluster -i instance1
```

2.3.30 leaveFarm

Removes an application server instance from a farm.

Type

Configuration Management

Syntax

leaveFarm

Description

Removes an 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 instance.

- When an 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 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 Application Server Control, after issuing the `dcmctl leaveFarm` command, you must stop and then start Oracle Enterprise Manager Application Server Control using the commands:

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

Example

To remove an instance from the farm:

```
dcmctl leaveFarm
```

2.3.31 listApplications

Lists the applications deployed in an OC4J instance.

Type

Application

Syntax

listApplications [-cl cluster_name | -i instance_name | -arch archive_name] [-co component_name]   [-sort]

Description

List the applications deployed in the named OC4J instance. The default is the home OC4J instance in the local application server instance.

Example

To list the applications in `home`:

```
dcmctl listApplications

1   BC4J

2   BC4JManager

3   transtrace
```

To list the applications in `OC4J_SECURITY` in `instance1`:

```
dcmctl listApplications -i instance1 -co OC4J_SECURITY

1   wirelesssso

2   oiddas

3   sso
```

2.3.32 listArchives

Returns a list of archive names.

Type

Archive

Syntax

listArchives [–arch archive_name] [-**sort**]

Description

Returns a list of archive names, and, with the –v (verbose) option, the display includes the name, source, version, and other information about the archive. If the –arch (archive name) option is used, only the named archive is listed.

Example

To list information about archives:

```
dcmctl listarchives -v
1
Name:        InstalledImage_M21.example.com
Source:      instance: M21.example.com
Version:     9.0.4.0.0
Comments:    This is an archive of the initial installed configuration.
Created:     2003-10-27 12:01:55.327
Clusterable: true


2
Name:        initial_archive_M21.example.com
Source:      instance: M21.example.com
Version:     9.0.4.0.0
Comments:    The initial archive after leaving the farm for M21.example.com
Created:     2003-10-30 09:08:22.591
Clusterable: true
```

2.3.33 listClusters

Lists the managed Oracle Application Server clusters in the local farm.

Type
Configuration Management

Syntax
listClusters

Description
This command lists the managed Oracle Application Server clusters in the farm that is associated with the local application server instance.

Example
```
dcmctl listClusters

1     cluster1

2     cluster2
```

2.3.34 listComponents

Lists components within the named scope.

Type
Configuration Management

Syntax
listComponents [-i instance_name] [-cl cluster_name] [-arch archive_name] [-sort]

Description
Returns a list of the components in the specified scope. Without arguments, this command returns a list of components in the local application server instance.

Components are listed in the format component type:component name.

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

Example
```
dcmctl listComponents

1  HTTP_Server:HTTP_Server

2  OC4J:OC4J_SECURITY

3  OC4J:home

4  OID:OID
```

2.3.35 listComponentTypes

Lists supported component types.

Type

dcmctl Properties

Syntax

listComponentTypes

Description

Lists the component types that DCM supports.

Example

```
dcmctl listComponentTypes
```

2.3.36 listInstances

Lists the application server instances in the farm.

Type

Configuration Management

Syntax

listInstances [-cl cluster_name]

Description

With no options, this command lists the application server instances that belong to the same farm as the local instance, but are not part of a cluster. If you use the -cl option, it lists only the instances that are part of the specified cluster.

Example

To list all non-clustered instances in the farm associated with the local instance:

```
dcmctl listInstances
```

To list the instances in cluster1:

```
dcmctl listInstances -cl cluster1
```

2.3.37 listOPMNLinks

Lists the instances that are in a non-managed Oracle Application Server cluster with the local instance.

Type

Configuration Management

Syntax

listOPMNlinks

Description

This command lists all instances that are in a non-managed Oracle Application Server cluster with the local instance. These instances could have been added to the cluster using the addOPMNLink command.

Example

```
dcmctl listopmnlinks

host1:6200

host2:6200
```

2.3.38 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
```

2.3.39 redeployApplication

Redeploys a J2EE application.

Type

Application

Syntax

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

where

- 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
- –rc 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 redeploys a J2EE application (WAR or EAR file) to the local 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
```

2.3.40 removeArchive

Deletes an archive file.

Type

Archive

Syntax

removeArchive -arch archive_name

Description

Deletes the named archive file.

Example

```
dcmctl removeArchive –arch archive3
```

2.3.41 removeCluster

Removes a cluster from the farm.

Type

Configuration Management

Syntax

removeCluster -cl cluster_name

Description

Remove the specified cluster from its farm. The cluster must contain no instances when it is removed. This command destroys all information about the cluster in the DCM repository.

Example

```
dcmctl removeCluster -cl cluster1
```

2.3.42 removeComponent

Destroys an OC4J instance.

Type

Configuration Management

Syntax

removeComponent -co component_name

Description

Destroy an OC4J instance. By default, the OC4J instance must belong to the local application server instance. Note that OC4J is the only component type allowed for this command. You cannot remove any component that was created by the installation process (such as OC4J_SECURITY).

Example

```
dcmctl removeComponent -co OC4J_myapps
```

2.3.43 removeOPMNLink

Removes instances from a non-managed Oracle Application Server cluster.

Type

Configuration Management

Syntax

removeOPMNLink hostname:port[, hostname:port...]

Description

Removes one or more instances from a non-managed Oracle Application Server cluster. You must run this command in the Oracle home of each instance in the cluster.

Example

This example shows a non-managed Oracle Application Server cluster with three instances on `host1`, `host2`, and `host3`. The command removes the `host2` 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`

2.3.44 repositoryRelocated

Notifies an instance that it is no longer hosting a repository.

> **Note:** Usually you use of the `repositoryRelocated` command, after an associated `exportRepository` and `importRepository`. Before running `importRepository` and `repositoryRelocated`, stop all DCM daemons in the instances that are part of the farm where you run `repositoryRelocated`.
>
> Use the following command in each instance in the farm to stop DCM daemons:
>
> ```
> dcmctl shell
> dcmctl> shutdown
> ```

Type

Configuration Management

Syntax

repositoryRelocated

Description

Notifies an instance that the repository it formerly hosted has been imported to another instance. This command is issued in the instance that formerly hosted the repository.

This command is used in conjunction with the importRepository command. A farm can only be associated with one repository at a time. When the repository is imported to a new instance of the farm, the old instance must be notified, through the `repositoryRelocated` command, that it is no longer the repository host.

Example

This example assumes that you have two 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,

```
dcmctl shell
dcmctl> exportrepository -f /export/repository_save_file
dcmctl> shutdown
```

If there are more than two instances, issue the `shutdown` command on all the other instances.

On instance2, shutdown and import the saved repository using these commands:

```
dcmctl shell
dcmctl> shutdown
dcmctl> importrepository -f /export/repository_save_file
```

On instance1, issue this command:

```
dcmctl> repositoryrelocated
```

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

On instance1, issue this command:

```
dcmctl> start -admin
```

When this command completes, on instance2 issue this command:

```
dcmctl> start -admin
```

If there are more than two instances, issue the following command sequentially on all of the other instances:

```
dcmctl> start -admin
```

2.3.45 resetDCMCachePort

Updates the port used by the DCM cache for instance discovery in Oracle Application Server 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 File based clusters. To find the current host and port, issue the command without arguments, as follows:

```
dcmctl resetDCMCachePort
```

To update the port associated with the current instance, issue the command with the new port number, as follows:

```
dcmctl resetDCMCachePort 12345
```

If the port value is changed on the instance that is hosting the repository, other instances in the farm may not be able to locate the repository. If this occurs, issue the following command in the instance that cannot locate the repository:

```
dcmctl resetDCMCachePort -r 12345
```

where `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 instance:

```
dcmctl resetdcmcacheport 12345
```

To notify an instance of the location of the repository:

```
dcmctl resetdcmcacheport -r 12345
```

2.3.46 resetFileTransaction

Resets a file 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
```

2.3.47 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 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 instance. If the host information changes for the repository host of a File based farm repository, it is best to have all instances running, so that all instances can locate the repository.

For instances that were not running during the change, it may be necessary to update the repository host information directly. Use the `-r` option and the name of the repository host to do this.

Example

```
resethostinformation
```

2.3.48 restart

Restarts processes. This command is deprecated in Oracle Application Server 10*g* and is provided for backward compatibility only. Use `opmnctl` to manage processes in Oracle Application Server 10*g*.

Type

Process Management

Syntax

restart [[-cl cluster_name] | [-i instance_name] | [-co component_name] | [-ct component_type]] | [-admin]

Description

Restarts running processes in the specified scope. This command does not restart OPMN or the DCM daemon, it leaves them running. Only the processes that were running when the command was issued are restarted. If the `-admin` option is used, then the DCM daemon is restarted.

Example

To restart the local instance:

```
restart
```

To restart a remote instance:

```
restart -i myInstance
```

To restart a component across a cluster:

```
restart -cl myCluster -co myComponent
```

2.3.49 restoreInstance

Restores configuration and application information to the local instance.

Type

Configuration Management

Syntax

restoreInstance [-dir directory_name]

Description

This command is deprecated; use archiving commands instead. This command restores the configuration and application information for the local instance from the specified directory. The directory must have been created with saveInstance; instance information can only be restored to the instance from which it was saved. To move configuration between instances in a farm, use the applyClusterTo and applyInstanceTo commands. To move configuration between farms, use the exportArchive and importArchive commands.

This command stops the instance, you must restart the instance after this command.

If the instance is part of a managed Oracle Application Server cluster, it will be removed from the cluster before the configuration information is restored. This operation does not affect other members of the cluster.

Example

To restore an instance to the configuration saved in `/private/config1`:

```
dcmctl restoreInstance -dir /private/config1
dcmctl start
```

2.3.50 resyncInstance

Resynchronizes instance configuration files with the DCM repository.

Type

Configuration Management

Syntax

resyncInstance [-force]|[-i instance_name]

Description

Resynchronizes the 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 instance. It updates the Oracle HTTP Server, OC4J, and OPMN configuration files, as well as `targets.xml`. 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 application server instance, unless you specify a different 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 re synchronize instance1 with the contents of the DCM repository:

```
dcmctl resyncinstance -i instance1
```

To force all files in the local instance to be updated with that is in the DCM repository:

```
dcmctl resyncinstance -force
```

2.3.51 saveInstance

Saves configuration and application information.

Type

Configuration Management

Syntax

saveInstance -dir directory_name

Description

This command is deprecated; use createArchive instead. This command saves the configuration and application information of the local instance to the designated directory. Creates the directory if it does not exist. If it does exist, then the specified directory must be empty. This command can be used to save the current configuration settings and installed J2EE applications before making configuration changes. You can then undo the changes, if necessary, using restoreInstance.

Example

```
saveInstance -dir instance1/config
```

2.3.52 set

Sets dcmctl options, timeout value, sorting preference, and number of versions for automatically archived instances.

Type

dcmctl Properties

Syntax

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

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

> **Note:** If you use dcmctl set commands while multiple DCM clients are active, the clients do not recognize the changed values until they are restarted. You must restart the dcmctl and Enterprise Manager clients after using dcmctl set with multiple active DCM clients.

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 instance:

```
set -arch 45
```

To turn off automatic archiving:

```
set -arch 0
```

2.3.53 setLogLevel

Sets the logging level.

Type
Shell

Syntax
setLogLevel [-admin] [error] [notification] [debug] [trace]

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

Example
Set the DCM daemon log level,

```
setloglevel -admin notification
```

Sets the dcmctl shell log level,

```
setloglevel notification
```

2.3.54 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
```

2.3.55 shutdown

Stops processes.

Type
Process Management

Description
Stops running processes in the local instance, including OPMN and the DCM daemon. To stop processes selectively, use stop.

Syntax
dcmctl shutdown

Example
```
dcmctl shutdown
```

2.3.56 start

Starts the named instance, cluster, or component. This command is deprecated in Oracle Application Server 10*g* and is provided for backward compatibility only. Use `opmnctl` to manage processes in Oracle Application Server 10*g*.

Type
Process Management

Syntax
dcmctl start [[-cl cluster_name] | [-i instance_name] | [-co comp_name] | [-ct comp_type]] | [-admin]

Description

Starts all components indicated with the scope options. If no options are supplied, the command starts all components in the local application server instance. If the −admin option is used, this command starts the DCM daemon.

Example

To start the cluster named myCluster:

```
dcmctl start -cl myCluster
```

To start all HTTP servers:

```
dcmctl start -ct HTTP_Server
```

2.3.57 stop

Stops processes. This command is deprecated in Oracle Application Server 10*g* and is provided for backward compatibility only. Use opmnctl to manage processes in Oracle Application Server 10*g*.

Type

Process Management

Syntax

stop [[-cl cluster_name] | [-i instance_name] | [-co component_name] | [-ct component_type]] | [-admin]

Description

Stops the specified processes. This command does not stop OPMN or the DCM daemon unless the −admin option is used; in that case, it stops the daemon. To stop everything, including the daemon, use shutdown.

Example

To stop the cluster named myCluster:

```
dcmctl stop -cl myCluster
```

To stop all HTTP servers:

```
dcmctl stop -ct ohs
```

To stop a component across a cluster:

```
dcmctl stop -cl myCluster -co myComponent
```

2.3.58 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 either `dcmctl` commands or use Application Server Control. If you use these tools, 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 configuration operations across multiple instances in a farm or cluster.

Type
Configuration Management

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

Description
Updates the DCM repository with the information in local configuration files. With no arguments, this command updates all DCM managed components, configuration files, as well as `targets.xml`. 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

When you make manual configuration file changes and then use `updateConfig`, follow these guidelines:

1. Before making a manual change requiring `updateConfig`, you should verify that your 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 instance or the cluster using createArchive.

3. While a configuration, make sure that there are no other administrative operations happening in the farm that may alter the configuration while you are making the manual change to the configuration. This includes Oracle Enterprise Manager Application Server Control 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. Finally, run `dcmctl updateConfig` to place the updated configuration files into the DCM repository.

If the precautions outlined in step 3 are not observed, there is a risk of conflicting changes being 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 DCM configuration files with the Distributed Configuration Management repository and restart them:

```
dcmctl updateConfig
```

```
dcmctl restart
```

If you have just updated 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
```

```
dcmctl restart -ct ohs
```

2.3.59 undeployApplication

Undeploys applications.

Type

Application

Syntax

undeployApplication -a application_name -co instance_name

Description

Undeploys the named application in the named Oracle Application Server Containers for J2EE instance.

Example

To undeploy the application testApp in the home instance:

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

2.3.60 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 petstore EAR file:

```
dcmctl validateEarFile -f petstore.ear

Warning: J2EE/DTD validation errors were foundADMN-906001

{0}   Base Exception:
oracle.ias.sysmgmt.deployment.j2ee.exception.J2eeDeploymentExce
ption: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]'.
```

2.3.61 whichCluster

Returns the cluster name for the named instance.

Type
Configuration Management

Description
Returns the name of the cluster that contains the named instance. If no instance is specified, returns the name of the cluster that contains the local instance.

Syntax
whichCluster [-i instance_name]

Example
```
dcmctl whichCluster
```

2.3.62 whichFarm

Identifies the type and location of the farm.

Type

Configuration Management

Identifies the type and location of the farm.

Type

Configuration Management

Description

Returns farm name, farm type (database or distributed file based repository), the hosting instance, and the host name. If the repository is hosted by a third-party database, the hosting instance name and host name are not available.

Syntax

dcmctl whichFarm

2.3.63 whichInstance

Returns the instance name.

Type

Configuration Management

Syntax

whichInstance

Description

Returns the name of the local application server instance.

Example

```
dcmctl whichInstance
```

3

Troubleshooting

This chapter contains information on troubleshooting problems you may encounter when using DCM. Usually, the problems are reported by the `dcmctl` utility.

This chapter contains the following major sections:

- Troubleshooting Guidelines
- DCM Problems and Solutions
- Troubleshooting Problems with Related Components

3.1 Troubleshooting Guidelines

This section presents a step-by-step approach for troubleshooting problems with DCM. It contains the following steps:

Step 1: Analyze the verbose and debug output from the dcmctl utility

When an error occurs, `dcmctl` usually provides a detailed error message and contextual information on a possible cause and solution. If an exception occurs, the `-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, the `dcmctl` utility sets the `-v` (verbose) option and the `-d` (debug) option `on`. Change these defaults with the set command.

To view the error message and stack trace that most recently occurred, use this command:

ORACLE_HOME/dcm/bin/dcmctl getError

Step 2: Examine the log files

If the information provided by the 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 3–1 describes the log files related to DCM.

Table 3–1   Distributed Configuration Management Log Files

File	Description
ORACLE_HOME/dcm/logs/dcmctl_logs/log.xml	DCM client log for dcmctl shell
ORACLE_HOME/dcm/logs/emd_logs/log.xml	DCM client log for Application Server Control
ORACLE_HOME/dcm/logs/daemon_logs/log.xml	DCM daemon log
ORACLE_HOME/dcm/logs/busrule_logs/log.xml	DCM business rule execution log

Table 3–1 (Cont.) Distributed Configuration Management Log Files

File	Description
`ORACLE_HOME/opmn/logs/*`	Oracle Process Manager and Notification Server logs

Step 3: Review the DCM 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 "DCM Problems and Solutions" on page 3-3.

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 3–2 provides tips on recognizing these types of problems and pointers to more information.

Table 3–2 Problems with Related Technologies

Type of Problem	May be due to:
Starting and stopping	Oracle Process Manager and Notification Server, Oracle HTTP Server, or Oracle Application Server Containers for J2EE problems
	See Also: *Oracle Process Manager and Notification Server Administrator's Guide*
	See Also: *Oracle Application Server Containers for J2EE User's Guide*
Application deployment or clustering	Oracle Application Server Containers for J2EE problems
	See Also: *Oracle Application Server Containers for J2EE User's Guide*
Directory or password problems	Oracle Internet Directory problems
	See Also: Troubleshooting Oracle Internet Directory
	See Also: *Oracle Internet Directory Administrator's Guide*, Appendix I, Troubleshooting
Repository problems (metadata repository)	Metadata Repository problems
	See Also: Troubleshooting the Metadata Repository
Repository problems (File based repository)	File system problems
	See Also: Troubleshooting the File based Repository

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

3.2 DCM Problems and Solutions

This section describes problems reported by Distributed Configuration Management and suggested techniques for resolving these problems.

It contains the following topics:

- OPMN is Unable to Start Due to Corrupted or Missing opmn.xml File
- BOM (Bill of Materials) Not Updated
- 'In Sync' Status is False
- Unable to Connect to the Directory
- Problem Accessing the Infrastructure Database
- DCM Daemon Cannot Start
- resyncInstance Command Does Not Restore Properly
- Invalid username/password
- dcmctl Takes a Long Time to Start a Component
- dcmctl Returns a SubscriptionException
- dcmctl getState Hangs
- dcmctl joinFarm Produces Error
- Repository Unreachable After repositoryRelocated Command
- The dcm.conf or dcmCache.xml file is empty
- Handling Unexpected Messages or Failures With a File Based Repository
- resetHostInformation Workaround for Unreachable IP Address
- DCM Repository Unavailable With Start Stop and Restart Commands
- The getState Command Cannot Be Used With Older Instances (9.0.2 or 9.0.3)
- Registered Plugin Named Discoverer Could Not Be Loaded Found in log.xml

3.2.1 OPMN is Unable to Start Due to Corrupted or Missing opmn.xml File

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
millseconds. 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
    ```

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, back up 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`.

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; 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 allows 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 3–1 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 3–1.

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 3–1   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>
  </process-manager>
</opmn>
```

> **See Also:**
>
> "dcmctl Returns a SubscriptionException" on page 3-10
>
> *Oracle Process Manager and Notification Server Administrator's Guide*

3.2.2 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 Oracle Application Server 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 Oracle Application Server, 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 instance with this command:

ORACLE_HOME/dcm/bin/dcmctl resyncinstance

3.2.3 ′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 Oracle Application Server 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 instance as follows:

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

2. Resynchronize the instance with this command:

   ORACLE_HOME/dcm/bin/dcmctl resyncinstance

   This will usually fix the problem. If not, the `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 Oracle Application Server instance are inconsistent with the configuration stored in the repository."

Solution 2

This indicates that the configuration files for the Oracle Application Server instance are out of sync with the configuration information stored in the DCM repository. Stop and resynchronize the instance. After the configuration files for the Oracle Application Server instance are in sync, synchronized, you can make your original modifications. Be sure to run `dcmctl updateconfig` again after making the changes.

> **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 Oracle Application Server 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.

3.2.4 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 repository.

See if restarting Oracle Internet Directory solves the problem. If it does not, refer to "Troubleshooting Oracle Internet Directory" on page 3-15.

3.2.5 Problem Accessing 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 repository.

See if restarting Oracle Internet Directory solves the problem. If it does not, refer to "Troubleshooting Oracle Internet Directory" on page 3-15.

3.2.6 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:

1. Review the error messages in `ORACLE_HOME/opmn/logs/dcm-daemon~dcm-daemon~dcm~1`.

2. Review the error messages in the log files in `ORACLE_HOME/opmn/logs/*`.

   > **See Also:** *Oracle Process Manager and Notification Server Administrator's Guide*

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.

3.2.7 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 instance whether it thinks it necessary or not:

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

3.2.8 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 3-15 for the steps for troubleshooting this problem.

3.2.9 dcmctl Takes a Long Time to Start a Component

Problem

`dcmctl start` times out while trying to start a component.

Solution

First, verify whether the component does eventually start. To do this, issue the following command repeatedly until the component starts successfully, or fails to start and returns an error:

`ORACLE_HOME/dcm/bin/dcmctl getreturnstatus`

If the component eventually starts, then it may be that the component needs more time than allowed by the current timeout setting to start. Some reasons for this are:

- The first time an Oracle Application Server Containers for J2EE (OC4J) instance is started, it takes more time than in subsequent starts.

- When all OC4J instances are started with one command, DCM starts them sequentially. If there are many OC4J instances, the time needed to start them will probably exceed the timeout.

  You can increase the timeout value for all DistributedDCM operations with the following command:

  `ORACLE_HOME/dcm/bin/dcmctl set -t timeout_in_seconds`

If the component fails to start, and returns an error, refer to the documentation for the component to resolve the problem, then try `dcmctl start` again.

3.2.10 dcmctl Returns a SubscriptionException

Problem

The `dcmctl start` command returns:

`ADMN-705002. Examining`

`ORACLE_HOME/dcm/logs/dcmctl_logs/log.xml`

`shows SubscriptionException: Subscription request timed out after 30000 milliseconds.`

Solution

This indicates that DCM is trying to communicate with Oracle Process Manager and Notification Server. This is usually due to a port conflict with one of the Oracle Notification Server (ONS) ports. Check the `ORACLE_HOME/opmn/conf/opmn.xml` file, examine the ports used for the `<notification-server>` element (local, remote, request), and verify that they are free.

> **See Also:**
>
> "OPMN is Unable to Start Due to Corrupted or Missing opmn.xml File" on page 3-3
>
> *Oracle Process Manager and Notification Server Administrator's Guide*

3.2.11 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.

3.2.12 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.
oracle.ias.repository.schema.SchemaException: DCM Repository is not Configured
.
.
.
```

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.

3.2.13 Repository Unreachable After repositoryRelocated Command

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 farm can only be associated with one repository at a time. When the repository is imported to a new instance of the farm, the old instance must be notified, through the `repositoryRelocated` command, that it is no longer the repository host. If the repository was not imported to a new instance, then there is currently no 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 instance, simply issue the importRepository command. After the repository is imported to a new instance, all 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 instance.

2. Navigate to the *ORACLE_HOME*/dcm/config directory.

3. Compare the files using this command:

   UNIX: `diff dcm.conf dcm.conf.bak`

   Windows: `fc dcm.conf.bak dcm.conf`

   The result should be:

   `< repository_type=cache`

   `> repository_type=cache rep`

4. Restore the `dcm.conf` file to the original version using this command.

   UNIX: `cp dcm.conf.bak dcm.conf`

   Windows: `copy dcm.conf.bak dcm.conf`

5. Start the instance.

   The repository is now available to the farm.

3.2.14 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 instance.

3.2.15 Handling Unexpected Messages or Failures With a File Based Repository

Problem

Occasionally, while using `dcmctl` commands with file based clustering, unexpected messages or failures may be seen.

Solution

Restarting all `dcmctl` processes, `dcm` daemons, and Application Server Control on all the instances in the farm may allow you to recover from these problems. Restarting these processes will not result in a loss of configuration information.

3.2.16 resetHostInformation Workaround for Unreachable IP Address

Problem

The following Application Server Control 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 `nlslookup` 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.

3.2.17 DCM Repository Unavailable With Start Stop and Restart Commands

Problem

If the DCM repository is not available, the `dcmctl start`, `stop`, and `restart` commands fail.

For example if `dcmctl` start returns the exception,

```
ADMN-202037

The DCM repository is unavailable and there is currently insufficient information
cached locally for DCM to initialize or the operation to be completed.
```

Solution

Work around: use `opmnctl` to start, stop, or restart processes.

3.2.18 The getState Command Cannot Be Used With Older Instances (9.0.2 or 9.0.3)

Problem

Invoking the command `dcmctl getstate -i` *instance_name* from an Oracle Application Server 10*g* (9.0.4) Oracle Home results in an ADMN-604104 exeption. This problem occurs when the *instance_name* is a 9.0.2 or 9.0.3 instance.

Solution

When trying to retrieve the state of a 9.0.2 or 9.0.3 instance, invoke the `dcmctl` utility from a 9.0.2 or 9.0.3 Oracle Home, and not from the Oracle Application Server 10*g* (9.0.4) Oracle Home.

3.2.19 Registered Plugin Named Discoverer Could Not Be Loaded Found in log.xml

Problem

After installing BI and Forms, the `$ORACLE_HOME/dcm/logs/daemon_logs/log.xml` file includes messages similar to the following:

```
<MSG_TEXT>The registered plugin named discoverer could not be loaded and will
be not be functional within the iAS Home: /private1/iasinst/OraHome1 until
this problem is resolved, due to the following exception: </MSG_TEXT>
<SUPPL_DETAIL><![CDATA[java.io.FileNotFoundException:
... /j2ee/OC4J_BI_Forms/applications/discoverer/web/WEB-I
NF/configuration.xsd (No such file or directory)
at java.io.FileInputStream.open(Native Method)
at java.io.FileInputStream.<init>(FileInputStream.java:103)
at java.io.FileInputStream.<init>(FileInputStream.java:66)
.
.
.
```

Solution

This problem causes no loss of service and requires no user or administrator action.

Explanation

This is an Oracle Application Server installation configuration tool timing issue that the system recovers from, with no action required by the user or the administrator.

When the OracleAS Discoverer component registers its SMI Plugin with DCM, the XML Schema files have not been placed in the Oracle Home. When DCM attempts to initialize the Plugin, the Plugin throws exceptions because the xsd files are not available. DCM correctly logs this exception and continues processing without the OracleAS Discoverer Plugin.

In the OC4J Configuration Assistant, the OracleAS Discoverer component deploys J2EE Applications. OracleAS Discoverer XML Schema files are contained within the application. At the end of the OC4J Configuration Assistant (OCA), the OCA calls an explicit API that causes DCM to reinitialize failed Plugins. Because the XML Schema documents, required by the OracleAS Discoverer Plugin, now exist on disk, the Plugin will correctly initialize and function.

3.3 Troubleshooting Problems with Related Components

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

3.3.1 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.

> **See Also:** *Oracle Process Manager and Notification Server Administrator's Guide*

3.3.2 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.

3.3.2.1 How DCM Uses Oracle Internet Directory

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

1. DCM obtains a password for the metadata repository from Oracle Internet Directory.

2. It logs in to Oracle Internet Directory using an internal, randomly generated password that is stored with the Oracle Application Server instance.

3. It obtains the metadata repository password from Oracle Internet Directory.

4. It uses this password to access the metadata repository.

3.3.2.2 How to Troubleshoot Oracle Internet Directory Problems

1. Determine the host and port the instance is using to access Oracle Internet Directory. Examine the *ORACLE_HOME*/config/ias.properties file in the Oracle home of the middle tier 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 Oracle Application Server 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 instance is using is the DBConnect field corresponding to ComponentNameDCM in the *ORACLE_HOME*/config/iasschema.xml file.

   - ***For Portal and Wireless and Business Intelligence and Forms installations:*** The metadata repository the 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 10g 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:

3.3.2.2.1 Option 1: Reset the password in Oracle Internet Directory and the metadata repository using the Oracle Enterprise Manager Application Server Control (Application Server Control):

1. Log in to the Application Server Control and navigate to the Instance Home page for the instance that is reporting the problem.

2. Click Infrastructure.

3. Select the DCM schema.

4. Click Change 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.

3.3.2.2.2 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
```

where `password_in_oid` is the DCM schema password retrieved from Oracle Internet Directory.

3.3.3 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.

3.3.3.1 How DCM Uses the Metadata Repository

DCM stores configuration information in the DCM schema in the metadata repository.

3.3.3.2 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 instance is listed in the DBConnect field in the *ORACLE_HOME*/config/iasschema.xml file.

2. If the error message states that an 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:

   METADATA_REPOSITORY_ORACLE_HOME/admin/iasdb/bdump

   METADATA_REPOSITORY_ORACLE_HOME/admin/iasdb/udump

   If there is evidence of database failure or corruption, refer to database documentation for information on restoring and recovering the database.

3.3.4 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.

3.3.4.1 How DCM Uses the File based Repository

DCM stores configuration information in *ORACLE_HOME*/dcm/repository.

3.3.4.2 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.

   For example, in UNIX, if the user is oracle, the repository file owner and permissions strings should be as shown in Example 3–2.

   Example 3–2   File based Repository Permissions

   ```
   drwxr-x---  4 oracle   g632            512 Apr  4 17:30 .
   drwxr-xr-x  7 oracle   g632            512 Apr  4 16:48 ..
   -rw-r-----  1 oracle   g632            990 Apr  4 17:45 app.bom
   -rw-r-----  1 oracle   g632              0 Apr  4 16:48 archive
   -rw-r-----  1 oracle   g632              0 Apr  4 16:48 archmember
   -rw-r-----  1 oracle   g632           1291 Apr  4 17:30 cluster
   -rw-r-----  1 oracle   g632             42 Apr  4 16:48 cluster.bom
   -rw-r-----  1 oracle   g632           1034 Apr  4 17:30 componentCluster
   -rw-r-----  1 oracle   g632            635 Apr  4 17:45 conf.bom
   -rw-r-----  1 oracle   g632            650 Apr  4 16:48 farm
   -rw-r-----  1 oracle   g632           1545 Apr  4 17:30 instance
   -rw-r-----  1 oracle   g632           3597 Apr  4 17:30 instanceAttr
   -rw-r-----  1 oracle   g632           3855 Apr  4 17:30 member
   drwxr-x---  2 oracle   g632            512 Apr  5 16:07 mutex
   -rw-r-----  1 oracle   g632            777 Apr  4 16:48 node
   -rw-r-----  1 oracle   g632           5140 Apr  4 17:30 obj
   -rw-r-----  1 oracle   g632              0 Apr  4 16:48 tag
   drwxr-x---  2 oracle   g632            512 Apr  4 17:30 tmp
   -rw-r-----  1 oracle   g632            550 Apr  4 17:39 tp.bom
   ```

2. If the error message states that an 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 instance is in a cluster:

   a. In the Oracle home of the instance you are trying to recover:

   ORACLE_HOME/dcm/bin/dcmctl resyncinstance

   If you have a backup of the Oracle home:

   a. Restore the repository files in *ORACLE_HOME*/dcm/repository.

b. In the Oracle home of the instance you are trying to recover, issue this command:

```
ORACLE_HOME/dcm/bin/dcmctl resyncinstance
```

4

Archiving Configurations

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
- Creating Archives
- Best Practices for Archiving
- Archive and Instance Compatibility
- Exporting and Importing Archives
- Using Automatic Archiving
- Saving Configurations
- Importing and Applying an Archive to an Instance

4.1 DCM Archiving Commands

The `dcmctl` utility provides commands (listed in Table 2–6, " Archive Commands") that enable you to create an archive of the configuration of an Oracle Application Server instance or cluster, and then apply the archived configuration to the same instance or cluster, or to a different instance or 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`.

> **See Also:**   "Using Distributed Configuration Management Archiving" on page 1-5

4.2 Initial Archive Creation

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

```
InstalledImageinstance name
```

4.3 Creating Archives

There are two ways to create archives:

- Using Automatic Archiving

- Using Explicit Archiving

> **See Also:** Chapter 4, "Archiving Configurations" on page 4-1

4.3.1 Using Automatic Archiving

Automatic Archiving creates an archive automatically prior to performing a DCM administrative operation. Use the dcmctl set command to verify that auto-archiving is enabled.

For example:

```
dcmctl set
Verbose:  true
Sort:  false
Warning:  true
Debug:  true
Default Timeout:  120
Auto Archive Count:  15
```

The dcmctl set command shows the Auto Archive count value. If the Auto Archive Count is nonzero, then auto-archiving is enabled and an archive will be automatically created prior to issuing a DCM administrative command.

> **See Also:** "set" on page 2-35

4.3.2 Using Explicit Archiving

Even when auto-archiving is enabled, the Administrator may want to explicitly create a named version of an archive prior to performing a DCM administrative operation. Explicitly created archives are saved until the Administrator deletes them.

For example, to create an archive prior to deploying a new J2EE application named "foo" use the command,

```
dcmctl createArchive -arch PriorToDeployingFoo -comment "prior to foo deploy V1"
```

When using createArchive, it is a good practice to use an archive name and a corresponding comment that identifies the version of configuration that the archive is associated with. Regardless of whether you use automatic or explicit archiving, it is recommended that you create an archive prior to performing any administrative operation.

> **See Also:** "createArchive" on page 2-9

4.4 Best Practices for Archiving

This section describes best practices for explicit and automatic archiving. It consists of the following topics.

- Creating Archives with createArchive
- Automatic Archiving and Archive Types
- Restoring Archives with applyArchiveTo

4.4.1 Creating Archives with createArchive

There are two archive types available to be created. One archives cluster-wide data while the other archives both cluster-wide data and one single instance's instance-specific data.

`createArchive` only archives data stored in the DCM Repository. After making manual modifications to OAS files, it is critical that `dcmctl updateConfig` be run. This command stores changes to DCM managed files in the DCM Repository and ensures that a properly created archive will then contain those changes.

> **Note:**   While it is recommended that the `-comment` switch, representing comments, be utilized to help describe the archive, this switch has been left off in the following documentation to help improve clarity of the concepts being discussed.

The contents of an archive created with `createArchive -arch` *myArchiveName* or `createArchive -arch` *myArchiveName* `-i <myInstanceName>` differ from the contents of an archive created with `createArchive -arch myArchiveName -cl <clusterName>`

The difference is as follows.

- Cluster-wide (`createArchive -cl`) archives contain only cluster-specific information and do not contain any information specific to the instance it is created on.
- Instance-specific (`createArchive -i` / or no option) archives contain cluster-specific information, plus any DCM managed information pertaining to the instance where `createArchive` was run.

If you have made instance-specific changes to an instance, and you want to ensure that they are saved, you should use the `-i` option and not the `-cl` option when running `createArchive`.

Currently, there is no simple way to differentiate instance-specific archives from cluster-specific archives. Due to this, it is highly recommended that you use comments to reflect which type of archive you are creating.

Note that `applyArchiveTo` does not apply instance-specific data, stored with the `-i` option, to a cluster. Please see the Best Practices for Restoring Archives section.

It is highly recommended that archives be exported for the highest level of protection.

> **Note:**   Please be certain to review the Automatic Archiving and
> Archive Types section to understand how Automatic Archiving
> handles archiving instance-specific data. It is also important to review
> the Best Practices for Restoring Archives section prior to needing to
> restore and archive to ensure that any restore operation goes smoothly
> and as expected.

Example 1

If you have a cluster which is comprised of three Oracle Application Server instances,
you have made instance-specific changes on one instance, and you wish to save those
changes in case you need to restore them in the future, you must run the following
command to properly save the changes.

```
createArchive -arch archive_name -i instance_name
```

Example 2

If you have a cluster which is comprised of three Oracle Application Server instances,
you have made instance-specific changes on all instances, and you wish to save those
changes in case you need to restore them in the future, you must run the following for
each instance you wish to back up.

```
% createArchive -arch archive_name -i instance_name1
% createArchive -arch archive_name -i instance_name2
% createArchive -arch archive_name -i instance_name2
```

4.4.2 Automatic Archiving and Archive Types

The Automatic Archiving feature of DCM works in the following fashion when
generating archives.

- If an instance is in a cluster, a cluster-specific (`createArchive -cl`) archive is
  generated.

- If an instance is standalone, an instance-specific (`createArchive -i`) archive is
  generated.

It is important to note that any instance-specific changes made to a clustered instance
do not get saved by the Automatic Archive mechanism. These changes will need to be
saved explicitly by the Administrator following the guidelines in the Best Practices for
Creating Archives section.

4.4.3 Restoring Archives with applyArchiveTo

The process of restoring an explicit or automatic archive is the same. There are five
different scenarios for restoring an archive.

- Restoring an archive to a cluster.

- Restoring an archive to a standalone instance.

- Restoring an archive to a non-clustered instance

- Restoring an instance-specific archive to only one instance in a cluster

- Restoring instance-specific archives to a set of instances in a cluster

The procedure for each of these scenarios is given below. Please note that by
peforming any of these procedures, the target cluster will be stopped. It is

recommended that you create and export an archive before restoring another archive in case there are any issues, including accidentally restoring an undesired archive.

4.4.3.1 Restoring an Archive to a Cluster

1. Pick an instance in the cluster as your management point.

2. Choose the archive you wish to restore. If necessary, import the archive from the file system using `importArchive`.

3. Run the following command.

   ```
   % applyarchiveTo -cl cluster_name -arch archive_name
   ```

4.4.3.2 Restoring an Archive to a Standalone Instance

1. Pick an instance in the cluster as your management point.

2. Choose the archive you wish to restore. If necessary, import the archive from the file system using `importArchive`.

3. Run the following command.

   ```
   % applyarchiveTo -cl archive_name -arch archive_name
   ```

4.4.3.3 Restoring an Archive to a Non-clustered Instance

1. Pick an instance in the cluster as your management point.

2. Choose the archive you wish to restore. If necessary, import the archive from the file system using `importArchive`.

3. Run the following command.

   ```
   % applyarchiveTo -arch archive_name
   ```

4.4.3.4 Restoring an Instance-specific Archive to Only One Instance in a Cluster

Run these steps if you wish to restore instance-specific changes to an instance which is part of a cluster. Upon rejoining the cluster, cluster-specific information will be reset to cluster-settings, but instance-specific data will not be modified.

1. Pick an instance in the cluster as your management point.

2. Choose the archive you wish to restore. If necessary, import the archive from the file system using `importArchive`.

3. Have the instance leave the cluster by running `leaveCluster` on that instance.

   ```
   % leaveCluster
   ```

4. Apply the archive.

   ```
   % applyArchiveTo -arch archive_name
   ```

5. Verify that the instance-specific changes were restored.

6. Rejoin the cluster.

   ```
   % joinCluster -cl cluster_name
   ```

4.4.3.5 Restoring Instance-specific Archives to a Set of Instances in a Cluster

You can also restore instance-specific changes to some or all instances which are part of a cluster. To do this, repeat the steps outlined in Restoring an Instance-specific Archive to Only One Instance in a Cluster for each instance you want to restore.

4.5 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 cluster) is stored in the repository. This archived image can then be applied to any compatible instance or cluster in the repository, or exported to a file to be applied to an instance or cluster in another repository. The compatibility of an archive with an instance or cluster is similar to the compatibility of instances to be clustered.

Applying archives to instances and 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 a cluster, the archive cannot contain any non-clusterable objects.

- Information specific to an instance, such as host name, can only be applied back to the instance from which it came.

- The configuration from one instance or cluster may be applied directly to another instance or cluster. That is, the apply functionality can take, as a source, an archive of an instance or a cluster.

> **See Also:** "isClusterable" on page 2-19

4.6 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 repository to another Database repository
- A File based repository to another File based repository
- A Database repository to a File based repository
- A File based repository to a Database 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 per component instance. Example 4–1, Example 4–2, and Example 4–3 provide sample archive files.

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

Example 4–1   Expanded Export file

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

Example 4–2   Export_Information File

```
Export_Information:
Version: 9.0.4.0.0
Comments: "User comments about the export"
Farm: dbmachine.1234.5678
Create Time: Mon Sep 30 12:42:10 PDT 2002
```

Example 4–3   Archive_Information File

```
Archive_Information:
Name: archive1
Source: cluster1
Version: 9.0.4.0.0
Configured Component Types: [Apache, J2EE]
Comments: "User comments about the archive"
Create Time: 2002-09-26 16:46:40.0
Clusterable:true
```

4.7 Using Automatic Archiving

Using automatic archiving, an archive is created automatically when you perform any operation listed in Table 4–1. The automatic archives coexist with archives created with the createArchive command, and are made distinct by their system-generated name. Example 4–6 shows an example of an automatic archive name for archive 2 in the example.

Table 4–1   Automatic Archive Operations

Add an instance to a cluster

Change a cluster's attributes

Change an instance's attributes

Configure Oracle Application Server Single Sign-On with configuration tool

Create a new OC4J instance

Deploy a J2EE application

Issue the updateConfig command

Join a cluster

Configure Oracle Application Server using Enterprise Manager standalone console

Remove an OC4J instance

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 4–4 and Example 4–5.

Example 4–4   Setting Automatic Archiving to Save 10 Versions

```
dcmctl set -arch 10
```

Example 4–5   Turning Automatic Archiving Off

```
dcmctl set -arch 0
```

The integer value specifies the automatic archive limit for the associated instance or cluster. For example, if you have a farm with 10 instances and each instance has an

automatic archive count of 15, the automatic archive count will continue to grow until 150 archives are created.

Automatic archives have system-generated names that resemble those shown in archive 2 in Example 4–6. 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 farm. When an instance is a member of a cluster, the Source: identifier is shown as: "`cluster: `*`cluster name`*".

Example 4–6   Automatic Archive Names (listArchives Command Output)

```
dcmctl> listarchives
1
Name:         InstalledImage_10gM21.example.com
Source:       instance: 10gM21.example.com
Version:      9.0.4.0.0
Comments:      This is an archive of the initial installed configuration for
 this instance.
Created:      2003-11-12 14:08:34.162
Clusterable:  true

2
Name:         dcm.autoarchive_222.222.3.12.999f1921.f8d70c4e79.-7fff
Source:       instance: 10gM21.example.com
Version:      9.0.4.0.0
Comments:     Automatic archival prior to hand-editing of configuration files
 OC4J OHS opmn jazn
Created:      2003-11-13 13:45:49.259
Clusterable:  true
```

4.8 Saving Configurations

You should save the DCM configuration regularly. You may find it useful to create "before" and "after" snapshots of a configuration when extensive configuration changes are performed.

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`*

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, archives saved there are lost. Exporting the archive to a file system provides an extra measure of safety.

4.9 Importing and Applying an Archive to an Instance

If, by accident, an instance is destroyed or a repository contains errors, you can restore the configuration from an exported archive with the import command:

```
 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
```

Index