

Oracle® Clusterware

Clusterware Administration and Deployment Guide



20c
F16801-01
February 2020

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2007, 2020, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Richard Strohm, David Jimenez Alvarez, Donald Graves, Lars Mortimer, Kamalesh Ramaswamy, Manasvi Nishith Vohra

Contributors: Troy Anthony, Ram Avudaiappan, Mark Bauer, Devang Bagaria, Eric Belden, Suman Bezawada, Gajanan Bhat, Burt Clouse, Ian Cookson, Jonathan Creighton, Mark Fuller, Apostolos Giannakidis, Angad Gokakkar, John Grout, Vikash Gunreddy, Andrey Gusev, Winston Huang, Shankar Iyer, Sameer Joshi, Ashwinee Khaladkar, Roland Knapp, Erich Kreisler, Karen Li, Barb Lundhild, Manuel Garcia Maciel, Saar Maoz, John McHugh, Markus Michalewicz, Anil Nair, Siva Nandan, Philip Newlan, Balaji Pagadala, Srinivas Poovala, Sampath Ravindhran, Kevin Reardon, Dipak Saggi, Duane Smith, Janet Stern, Su Tang, James Warnes, Douglas Williams, Soo Huey Wong

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xxx
Documentation Accessibility	xxx
Related Documents	xxx
Conventions	xxxi

1 Introduction to Oracle Clusterware

Changes in Oracle Clusterware 20c	1-1
Deprecated Features in Oracle Clusterware 20c	1-2
Desupported Features in Oracle Clusterware 20c	1-2
Overview of Oracle Clusterware	1-3
Understanding System Requirements for Oracle Clusterware	1-5
Oracle Clusterware Hardware Concepts and Requirements	1-5
Oracle Clusterware Operating System Concepts and Requirements	1-6
Oracle Clusterware Software Concepts and Requirements	1-7
Oracle Clusterware Network Configuration Concepts	1-8
Single Client Access Name (SCAN)	1-9
Manual Addresses Configuration	1-9
Overview of Oracle Clusterware Platform-Specific Software Components	1-10
The Oracle Clusterware Technology Stack	1-10
The Cluster Ready Services Technology Stack	1-10
The Oracle High Availability Services Technology Stack	1-11
Oracle Clusterware Processes on Windows Systems	1-15
Overview of Installing Oracle Clusterware	1-16
Oracle Clusterware Version Compatibility	1-16
Overview of Upgrading and Patching Oracle Clusterware	1-17
Overview of Grid Infrastructure Management Repository	1-18
Overview of Domain Services Clusters	1-19
Overview of Managing Oracle Clusterware Environments	1-21
Overview of Command Evaluation	1-23
Overview of Cloning and Extending Oracle Clusterware in Grid Environments	1-24
Overview of the Oracle Clusterware High Availability Framework and APIs	1-24

Overview of Cluster Time Management	1-25
Activating and Deactivating Cluster Time Management	1-26

2 Oracle Clusterware Configuration and Administration

Role-Separated Management	2-1
Managing Cluster Administrators	2-2
Configuring Role Separation	2-2
Configuring Oracle Grid Infrastructure Using Grid Setup Wizard	2-4
Configuring a Single Node	2-5
Configuring Multiple Nodes	2-5
Upgrading Oracle Grid Infrastructure	2-6
Running the Configuration Wizard in Silent Mode	2-6
Moving and Patching an Oracle Grid Infrastructure Home	2-7
Server Weight-Based Node Eviction	2-7
Overview of Oracle Database Quality of Service Management	2-8
Overview of Grid Naming Service	2-9
Network Administration Tasks for GNS and GNS Virtual IP Address	2-9
Understanding Grid Naming Service Configuration Options	2-10
Highly-Available Grid Naming Service	2-11
Automatic Configuration Option for Addresses	2-12
Static Configuration Option for Addresses	2-12
Shared GNS Option for Addresses	2-12
Administering Grid Naming Service	2-13
Configuring Highly-Available GNS	2-13
Removing Primary and Secondary GNS Instances	2-14
Starting and Stopping GNS with SRVCTL	2-15
Converting Clusters to GNS Server or GNS Client Clusters	2-15
Converting a Non-GNS Cluster to a GNS Server Cluster	2-16
Converting a Non-GNS Cluster to a Client Cluster	2-16
Converting a Single Cluster Running GNS to a Server Cluster	2-17
Converting a Single Cluster Running GNS to be a GNS Client Cluster	2-17
Moving GNS to Another Cluster	2-18
Changing the GNS Subdomain when Moving from IPv4 to IPv6 Network	2-19
Rolling Conversion from DNS to GNS Cluster Name Resolution	2-20
Node Failure Isolation	2-21
Server Hardware Configuration for IPMI	2-22
Post-installation Configuration of IPMI-based Failure Isolation Using CRSCCTL	2-22
IPMI Post-installation Configuration with Oracle Clusterware	2-23
Modifying IPMI Configuration Using CRSCCTL	2-24
Removing IPMI Configuration Using CRSCCTL	2-25

Understanding Network Addresses on Manually Configured Networks	2-26
Understanding Network Address Configuration Requirements	2-26
About IPv6 Address Formats	2-27
Name Resolution and the Network Resource Address Type	2-27
Understanding SCAN Addresses and Client Service Connections	2-28
SCAN Listeners and Service Registration Restriction With Valid Node Checking	2-29
Configuring Shared Single Client Access Names	2-30
About Configuring Shared Single Client Access Names	2-30
Configuring the Use of Shared SCAN	2-31
Changing Network Addresses on Manually Configured Systems	2-32
Changing the Virtual IP Addresses Using SRVCTL	2-32
Changing Oracle Clusterware Private Network Configuration	2-34
About Private Networks and Network Interfaces	2-35
Redundant Interconnect Usage	2-35
Consequences of Changing Interface Names Using OIFCFG	2-36
Changing a Network Interface	2-37
Creating a Network Using SRVCTL	2-39
Network Address Configuration in a Cluster	2-40
Changing Static IPv4 Addresses To Static IPv6 Addresses Using SRVCTL	2-41
Changing Dynamic IPv4 Addresses To Dynamic IPv6 Addresses Using SRVCTL	2-43
Changing an IPv4 Network to an IPv4 and IPv6 Network	2-44
Transitioning from IPv4 to IPv6 Networks for VIP Addresses Using SRVCTL	2-44
Cross-Cluster Dependency Proxies	2-44

3 Policy-Based Cluster and Capacity Management

Overview of Server Pools and Policy-Based Management	3-1
Server Pools and Server Categorization	3-2
Server Pools and Policy-Based Management	3-2
How Server Pools Work	3-3
Default Server Pools	3-3
The Free Server Pool	3-3
The Generic Server Pool	3-3
Server Pool Attributes	3-4
How Oracle Clusterware Assigns New Servers Using Server Pools	3-7
Servers Moving from Server Pool to Server Pool	3-9
Managing Server Pools Using Default Attributes	3-9
Overview of Server Categorization	3-10
Overview of Cluster Configuration Policies and the Policy Set	3-10
Load-Aware Resource Placement	3-11
Server Configuration and Server State Attributes	3-12

Memory Pressure Management for Database Servers	3-15
Server Category Attributes	3-16
An Example Policy Set Configuration	3-17

4 Oracle Flex Clusters

Overview of Oracle Flex Clusters	4-1
Managing Oracle Flex Clusters	4-3
Changing the Cluster Mode	4-3
Changing an Oracle Clusterware Standard Cluster to an Oracle Flex Cluster	4-3
Oracle Extended Clusters	4-4
Configuring Oracle Extended Clusters	4-5

5 Oracle Fleet Patching and Provisioning

Fleet Patching and Provisioning Architecture	5-7
Fleet Patching and Provisioning Server	5-9
Fleet Patching and Provisioning Targets	5-10
Fleet Patching and Provisioning Clients	5-10
Authentication Options for Fleet Patching and Provisioning Operations	5-11
Fleet Patching and Provisioning Roles	5-12
Basic Built-In Roles	5-13
Composite Built-In Roles	5-14
Fleet Patching and Provisioning Images	5-14
Gold Image Distribution Among Fleet Patching and Provisioning Servers	5-15
Fleet Patching and Provisioning Server Auditing	5-17
Fleet Patching and Provisioning Notifications	5-17
Fleet Patching and Provisioning Implementation	5-17
Creating a Fleet Patching and Provisioning Server	5-18
Adding Gold Images to the Fleet Patching and Provisioning Server	5-19
Image State	5-20
Image Series	5-20
Image Type	5-20
Provisioning Copies of Gold Images	5-21
User Group Management in Fleet Patching and Provisioning	5-22
Storage Options for Provisioned Software	5-24
Provisioning for a Different User	5-25
Propagating Images Between Fleet Patching and Provisioning Servers	5-26
Oracle Grid Infrastructure Management	5-27
About Deploying Oracle Grid Infrastructure Using Oracle Fleet Patching and Provisioning	5-27

Provisioning Oracle Grid Infrastructure Software	5-29
Patching Oracle Grid Infrastructure Software	5-31
Patching Oracle Grid Infrastructure Using the Rolling Method	5-31
Patching Oracle Grid Infrastructure Using the Non-Rolling Method	5-32
Patching Oracle Grid Infrastructure Using Batches	5-32
Combined Oracle Grid Infrastructure and Oracle Database Patching	5-36
Zero-Downtime Oracle Grid Infrastructure Patching	5-38
Patching Oracle Grid Infrastructure Using Local-Mode Configuration	5-38
Error Prevention and Automated Recovery Options	5-39
Upgrading Oracle Grid Infrastructure Software	5-40
Oracle Database Software Management	5-41
Provisioning a Copy of a Gold Image of a Database Home	5-41
Creating an Oracle Database on a Copy of a Gold Image	5-42
Patching Oracle Database Software	5-43
Patching Oracle Database with the Independent Automaton	5-45
Patching Oracle Exadata Software	5-46
Upgrading Oracle Database Software	5-48
Zero-Downtime Upgrade	5-49
Running a Zero-Downtime Upgrade Using Oracle GoldenGate for Replication	5-50
Running a Zero-Downtime Upgrade Using Oracle Data Guard for Replication	5-52
Customizing Zero-Downtime Upgrades	5-54
Persistent Home Path During Patching	5-54
Managing Fleet Patching and Provisioning Clients	5-55
Creating a Fleet Patching and Provisioning Client	5-55
Enabling and Disabling Fleet Patching and Provisioning Clients	5-57
Deleting a Fleet Patching and Provisioning Client	5-58
Creating Users and Assigning Roles for Fleet Patching and Provisioning Client Cluster Users	5-59
Managing the Fleet Patching and Provisioning Client Password	5-59
User-Defined Actions	5-60
Job Scheduler for Operations	5-66
Oracle Restart Patching and Upgrading	5-67
Fleet Patching and Provisioning Use Cases	5-68
Creating an Oracle Grid Infrastructure 12c Release 2 Deployment	5-68
Provisioning an Oracle Database Home and Creating a Database	5-69
Provisioning a Pluggable Database	5-70
Upgrading to Oracle Grid Infrastructure 12c Release 2	5-70
Patching Oracle Grid Infrastructure Without Changing the Grid Home Path	5-71
Patching Oracle Grid Infrastructure and Oracle Databases Simultaneously	5-72
Patching Oracle Database 12c Release 1 Without Downtime	5-73
Upgrading to Oracle Database 12c Release 2	5-74

Adding a Node to a Cluster and Scaling an Oracle RAC Database to the Node	5-76
Adding Gold Images for Fleet Patching and Provisioning	5-76
User Actions for Common Fleet Patching and Provisioning Tasks	5-77

6 Managing Oracle Cluster Registry and Voting Files

Managing Oracle Cluster Registry and Oracle Local Registry	6-2
Migrating Oracle Cluster Registry to Oracle Automatic Storage Management	6-2
Migrating Oracle Cluster Registry from Oracle ASM to Other Types of Storage	6-4
Adding, Replacing, Repairing, and Removing Oracle Cluster Registry Locations	6-5
Adding an Oracle Cluster Registry Location	6-7
Removing an Oracle Cluster Registry Location	6-7
Replacing an Oracle Cluster Registry Location	6-8
Repairing an Oracle Cluster Registry Configuration on a Local Node	6-9
Overriding the Oracle Cluster Registry Data Loss Protection Mechanism	6-10
Backing Up Oracle Cluster Registry	6-11
Listing Backup Files	6-11
Changing Backup Location	6-12
Restoring Oracle Cluster Registry	6-12
Restoring the Oracle Cluster Registry on Linux or UNIX Systems	6-13
Restoring the Oracle Cluster Registry on Windows Systems	6-16
Restoring the Oracle Cluster Registry in an Oracle Restart Environment	6-18
Diagnosing Oracle Cluster Registry Problems	6-19
Administering Oracle Cluster Registry with Export and Import Commands	6-19
Importing Oracle Cluster Registry Content on Linux or UNIX Systems	6-20
Importing Oracle Cluster Registry Content on Windows Systems	6-22
Oracle Local Registry	6-24
Upgrading and Downgrading the Oracle Cluster Registry Configuration	6-25
Managing Voting Files	6-25
Storing Voting Files on Oracle ASM	6-26
Backing Up Voting Files	6-28
Restoring Voting Files	6-28
Adding, Deleting, or Migrating Voting Files	6-30
Modifying Voting Files that are Stored in Oracle ASM	6-30
Modifying Voting Files that are Not Stored on Oracle ASM	6-30
Migrating Voting Files to Oracle ASM	6-31
Verifying the Voting File Location	6-31

7 Adding and Deleting Cluster Nodes

Prerequisite Steps for Adding Cluster Nodes	7-1
---	-----

Adding and Deleting Cluster Nodes on Linux and UNIX Systems	7-4
Adding a Cluster Node on Linux and UNIX Systems	7-4
Deleting a Cluster Node on Linux and UNIX Systems	7-8
Adding and Deleting Cluster Nodes on Windows Systems	7-10
Adding a Node to a Cluster on Windows Systems	7-10
Deleting a Cluster Node on Windows Systems	7-13

8 Cloning Oracle Clusterware

Introduction to Cloning Oracle Clusterware	8-1
Preparing the Oracle Grid Infrastructure Home for Cloning	8-3
Step 1: Install Oracle Clusterware	8-3
Step 2: Shut Down Running Software	8-4
Step 3: Create a Copy of the Oracle Grid Infrastructure Home	8-4
Method 1: Create a Copy of the Oracle Grid Infrastructure Home Using the Setup Wizard	8-4
Method 2a: Create a Copy of the Oracle Grid Infrastructure Home by Removing Unnecessary Files from the Copy	8-5
Method 2b: Create a Copy of the Oracle Grid Infrastructure Home Using the -X Option	8-6
Creating a Cluster by Cloning Oracle Clusterware	8-7
Step 1: Prepare the New Cluster Nodes	8-7
Step 2: Deploy the Oracle Grid Infrastructure Home	8-8
Step 3: Run the gridSetup.sh Utility	8-9
Using Cloning to Add Nodes to a Cluster	8-10
Locating and Viewing Log Files Generated During Cloning	8-11

9 Making Applications Highly Available Using Oracle Clusterware

Oracle Clusterware Resources and Agents	9-1
Oracle Clusterware Resources	9-2
Virtual Machine Resources	9-2
Resource Groups	9-5
Oracle Clusterware Resource Types	9-13
Agents in Oracle Clusterware	9-16
Oracle Clusterware Built-in Agents	9-19
Action Scripts	9-20
Building an Agent	9-20
Building and Deploying C and C++ Agents	9-21
Registering a Resource in Oracle Clusterware	9-22
Overview of Using Oracle Clusterware to Enable High Availability	9-23
Resource Attributes	9-24

Resource States	9-25
Resource Dependencies	9-26
Start Dependencies	9-26
Stop Dependencies	9-31
Resource Placement	9-32
Registering an Application as a Resource	9-33
Creating an Application VIP Managed by Oracle Clusterware	9-33
Adding an Application VIP with Oracle Enterprise Manager	9-35
Adding User-Defined Resources	9-36
Deciding on a Deployment Scheme	9-36
Adding a Resource to a Specified Server Pool	9-37
Adding a Resource Using a Server-Specific Deployment	9-38
Creating Resources that Use the generic_application Resource Type	9-39
Adding Resources Using Oracle Enterprise Manager	9-41
Changing Resource Permissions	9-42
Application Placement Policies	9-42
Unregistering Applications and Application Resources	9-43
Managing Resources	9-43
Registering Application Resources	9-44
Starting Application Resources	9-44
Relocating Applications and Application Resources	9-45
Stopping Applications and Application Resources	9-45
Displaying Clusterware Application and Application Resource Status Information	9-46
Managing Automatic Restart of Oracle Clusterware Resources	9-46
Preventing Automatic Restarts of Oracle Clusterware Resources	9-47
Automatically Manage Restart Attempts Counter for Oracle Clusterware Resources	9-47

A Cluster Verification Utility Reference

About Cluster Verification Utility	A-1
Overview of CVU	A-2
CVU Operational Notes	A-3
CVU Installation Requirements	A-3
CVU Usage Information	A-4
CVU Configuration File	A-5
Privileges and Security	A-6
Using CVU Help	A-7
Deprecated and Desupported CLUVFY Commands	A-7
Special CVU Topics	A-7
Generating Fixup Scripts	A-8

Using CVU to Determine if Installation Prerequisites are Complete	A-8
Using CVU with Oracle Database 10g Release 1 or 2	A-9
Entry and Exit Criteria	A-9
Verbose Mode and UNKNOWN Output	A-9
CVU Node List Shortcuts	A-10
Cluster Verification Utility Command Reference	A-11
cluvfy comp acfs	A-11
cluvfy comp admprv	A-11
cluvfy comp asm	A-13
cluvfy comp baseline	A-14
cluvfy comp clocksync	A-16
cluvfy comp clumgr	A-16
cluvfy comp crs	A-17
cluvfy comp dhcp	A-17
cluvfy comp dns	A-19
cluvfy comp freespace	A-20
cluvfy comp gns	A-20
cluvfy comp gpnv	A-21
cluvfy comp ha	A-22
cluvfy comp healthcheck	A-22
cluvfy comp nodeapp	A-23
cluvfy comp nodecon	A-24
cluvfy comp nodereach	A-25
cluvfy comp ocr	A-26
cluvfy comp ohasd	A-27
cluvfy comp olr	A-28
cluvfy comp peer	A-29
cluvfy comp scan	A-30
cluvfy comp software	A-30
cluvfy comp space	A-31
cluvfy comp ssa	A-32
cluvfy comp sys	A-34
cluvfy comp vdisk	A-36
cluvfy stage [-pre -post] acfscfg	A-36
cluvfy stage -post appcluster	A-37
cluvfy stage [-pre -post] cfs	A-37
cluvfy stage [-pre -post] crsinst	A-38
cluvfy stage -pre dbcfg	A-40
cluvfy stage -pre dbinst	A-41
cluvfy stage [-pre -post] hacfg	A-43
cluvfy stage -post hwos	A-44

cluvfy stage [-pre -post] nodeadd	A-45
cluvfy stage -post nodedel	A-46
Troubleshooting and Diagnostic Output for CVU	A-46
Enabling Tracing	A-47
Known Issues for the Cluster Verification Utility	A-47
Database Versions Supported by Cluster Verification Utility	A-47
Linux Shared Storage Accessibility (ssa) Check Reports Limitations	A-47
Shared Disk Discovery on Red Hat Linux	A-47

B Oracle Clusterware Resource Reference

Resource Attributes	B-1
Configurable Resource Attributes	B-2
ACL	B-3
ACTION_SCRIPT	B-4
ACTION_TIMEOUT	B-4
ACTIONS	B-4
ACTIVE_PLACEMENT	B-5
AGENT_FILENAME	B-5
ALERT_TEMPLATE	B-5
AUTO_START	B-5
CARDINALITY	B-6
CARDINALITY_ID	B-6
CHECK_INTERVAL	B-6
CHECK_TIMEOUT	B-6
CLEAN_TIMEOUT	B-7
CRITICAL_RESOURCES	B-7
DELETE_TIMEOUT	B-7
DESCRIPTION	B-7
ENABLED	B-7
FAILURE_INTERVAL	B-8
FAILURE_THRESHOLD	B-8
HOSTING_MEMBERS	B-8
INSTANCE_FAILOVER	B-9
INTERMEDIATE_TIMEOUT	B-9
LOAD	B-9
MODIFY_TIMEOUT	B-9
NAME	B-10
OFFLINE_CHECK_INTERVAL	B-10
ONLINE_RELOCATION_TIMEOUT	B-10
PLACEMENT	B-10

RELOCATE_KIND	B-11
RELOCATE_BY_DEPENDENCY	B-11
RESTART_ATTEMPTS	B-11
SCRIPT_TIMEOUT	B-11
SERVER_CATEGORY	B-12
SERVER_POOLS	B-12
START_CONCURRENCY	B-13
START_DEPENDENCIES	B-13
START_TIMEOUT	B-16
STOP_CONCURRENCY	B-16
STOP_DEPENDENCIES	B-16
STOP_TIMEOUT	B-17
UPTIME_THRESHOLD	B-18
USER_WORKLOAD	B-18
USE_STICKINESS	B-19
Read-Only Resource Attributes	B-19
ACTION_FAILURE_EVENT_TEMPLATE	B-19
INSTANCE_COUNT	B-19
INTERNAL_STATE	B-19
LAST_SERVER	B-20
LAST_STATE_CHANGE	B-20
PROFILE_CHANGE_EVENT_TEMPLATE	B-20
RESOURCE_LIST	B-20
RESTART_COUNT	B-20
STATE	B-20
STATE_CHANGE_EVENT_TEMPLATE	B-21
STATE_DETAILS	B-21
TARGET	B-21
TARGET_SERVER	B-21
TYPE	B-21
Deprecated Resource Attributes	B-21
DEGREE	B-22
Examples of Action Scripts for Third-party Applications	B-22

C OLSNODES Command Reference

Using OLSNODES	C-1
Overview	C-1
Operational Notes	C-1
Summary of the OLSNODES Command	C-1
Syntax	C-2

D Oracle Interface Configuration Tool (OIFCFG) Command Reference

Starting the OIFCFG Command-Line Interface	D-1
Summary of the OIFCFG Usage	D-1
OIFCFG Command Format	D-2
OIFCFG Commands	D-2
OIFCFG Command Parameters	D-2
OIFCFG Usage Notes	D-3
OIFCFG Examples	D-5

E Oracle Clusterware Control (CRSCTL) Utility Reference

CRSCTL Overview	E-1
Clusterized (Cluster Aware) Commands	E-2
CRSCTL Operational Notes	E-2
Deprecated Subprograms or Commands	E-3
Dual Environment CRSCTL Commands	E-6
crsctl check css	E-6
crsctl check evm	E-6
crsctl get hostname	E-7
crsctl add resource	E-7
crsctl delete resource	E-11
crsctl eval add resource	E-12
crsctl eval fail resource	E-14
crsctl eval relocate resource	E-15
crsctl eval modify resource	E-16
crsctl eval start resource	E-18
crsctl eval stop resource	E-18
crsctl getperm resource	E-19
crsctl modify resource	E-20
crsctl relocate resource	E-22
crsctl restart resource	E-24
crsctl setperm resource	E-25
crsctl start resource	E-27
crsctl status resource	E-28
crsctl stop resource	E-31
crsctl add resourcegroup	E-33
crsctl check resourcegroup	E-33
crsctl delete resourcegroup	E-34

crsctl eval add resourcegroup	E-35
crsctl eval fail resourcegroup	E-36
crsctl eval relocate resourcegroup	E-36
crsctl eval start resourcegroup	E-37
crsctl eval stop resourcegroup	E-38
crsctl export resourcegroup	E-38
crsctl modify resourcegroup	E-39
crsctl relocate resourcegroup	E-41
crsctl restart resourcegroup	E-42
crsctl start resourcegroup	E-43
crsctl status resourcegroup	E-44
crsctl stop resourcegroup	E-46
crsctl add resourcegrouptype	E-47
crsctl delete resourcegrouptype	E-48
crsctl modify resourcegrouptype	E-49
crsctl get tracefileopts	E-49
crsctl set tracefileopts	E-49
crsctl add type	E-50
crsctl delete type	E-53
crsctl getperm type	E-53
crsctl modify type	E-54
crsctl setperm type	E-55
crsctl status type	E-56
crsctl add wallet	E-57
crsctl delete wallet	E-58
crsctl modify wallet	E-59
crsctl query wallet	E-60
Oracle RAC Environment CRSCTL Commands	E-61
crsctl request action	E-61
crsctl add category	E-62
crsctl delete category	E-63
crsctl modify category	E-64
crsctl status category	E-65
crsctl check cluster	E-65
crsctl start cluster	E-66
crsctl stop cluster	E-67
crsctl get cluster class	E-68
crsctl get cluster configuration	E-68
crsctl set cluster disabledtlsciphersuite	E-68
crsctl get cluster extended	E-68
crsctl get cluster hubsize	E-69

crsctl set cluster hubsize	E-69
crsctl get cluster mode	E-69
crsctl set cluster mode	E-70
crsctl get cluster name	E-70
crsctl add cluster site	E-70
crsctl delete cluster site	E-71
crsctl modify cluster site	E-71
crsctl query cluster site	E-72
crsctl get cluster tlsciphersuite	E-73
crsctl get cluster type	E-73
crsctl set cluster type	E-73
crsctl get cpu equivalency	E-73
crsctl set cpu equivalency	E-74
crsctl get restricted placement	E-74
crsctl set restricted placement	E-75
crsctl check crs	E-76
crsctl config crs	E-76
crsctl disable crs	E-76
crsctl enable crs	E-77
crsctl start crs	E-77
crsctl stop crs	E-78
crsctl query crs activeversion	E-79
crsctl add crs administrator	E-79
crsctl delete crs administrator	E-80
crsctl query crs administrator	E-81
crsctl query crs autostart	E-82
crsctl set crs autostart	E-82
crsctl query crs releasepatch	E-83
crsctl query crs releaseversion	E-83
crsctl query crs site	E-83
crsctl query crs softwarepatch	E-84
crsctl query crs softwareversion	E-84
crsctl get css	E-85
crsctl pin css	E-85
crsctl set css	E-86
crsctl unpin css	E-86
crsctl unset css	E-87
crsctl get ipmi binaryloc	E-87
crsctl set ipmi binaryloc	E-88
crsctl get css ipmiaddr	E-88
crsctl set css ipmiaddr	E-89

crsctl set css ipmiadmin	E-89
crsctl query css ipmiconfig	E-90
crsctl unset css ipmiconfig	E-90
crsctl query css ipmidevice	E-91
crsctl get css noautorestart	E-92
crsctl set css noautorestart	E-92
crsctl delete css votedisk	E-92
crsctl query css votedisk	E-93
crsctl check ctss	E-94
crsctl discover dhcp	E-94
crsctl get clientid dhcp	E-95
crsctl release dhcp	E-95
crsctl request dhcp	E-96
crsctl replace discoverystring	E-96
crsctl query dns	E-97
crsctl start ip	E-98
crsctl status ip	E-98
crsctl stop ip	E-99
crsctl lsmodules	E-99
crsctl create member_cluster_configuration	E-100
crsctl delete member_cluster_configuration	E-101
crsctl query member_cluster_configuration	E-101
crsctl delete node	E-102
crsctl get nodename	E-102
crsctl get node role	E-103
crsctl add policy	E-103
crsctl delete policy	E-104
crsctl eval activate policy	E-105
crsctl modify policy	E-106
crsctl status policy	E-107
crsctl create policysset	E-108
crsctl modify policysset	E-108
crsctl status policysset	E-109
crsctl check resource	E-111
crsctl relocate resource	E-112
crsctl get resource use	E-114
crsctl set resource use	E-114
crsctl start rollingpatch	E-115
crsctl stop rollingpatch	E-115
crsctl start rollingupgrade	E-116
crsctl eval add server	E-116

crsctl eval delete server	E-118
crsctl eval relocate server	E-119
crsctl modify server	E-120
crsctl relocate server	E-121
crsctl status server	E-122
crsctl get server css_critical	E-124
crsctl set server css_critical	E-124
crsctl get server label	E-124
crsctl set server label	E-124
crsctl add serverpool	E-125
crsctl delete serverpool	E-127
crsctl eval add serverpool	E-127
crsctl eval delete serverpool	E-130
crsctl eval modify serverpool	E-132
crsctl getperm serverpool	E-135
crsctl modify serverpool	E-136
crsctl setperm serverpool	E-138
crsctl status serverpool	E-139
crsctl query socket udp	E-141
crsctl start testdns	E-142
crsctl status testdns	E-143
crsctl stop testdns	E-144
crsctl replace votedisk	E-144
Oracle Restart Environment CRSCTL Commands	E-145
crsctl check has	E-146
crsctl config has	E-146
crsctl disable has	E-146
crsctl enable has	E-147
crsctl query has releaseversion	E-147
crsctl query has softwareversion	E-147
crsctl start has	E-148
crsctl stop has	E-148
Troubleshooting and Diagnostic Output	E-148
Dynamic Debugging Using crsctl set log	E-149
Component Level Debugging	E-149
Enabling Debugging for Oracle Clusterware Modules	E-150
Enabling Debugging for Oracle Clusterware Resources	E-153

F Fleet Patching and Provisioning Control (RHPCTL) Command Reference

RHPCTL Overview	F-1
Using RHPCTL Help	F-1
RHPCTL Command Reference	F-2
rhpcctl delete audit	F-2
rhpcctl modify audit	F-2
rhpcctl query audit	F-2
rhpcctl add client	F-4
rhpcctl delete client	F-5
rhpcctl discover client	F-6
rhpcctl export client	F-7
rhpcctl modify client	F-7
rhpcctl query client	F-8
rhpcctl update client	F-9
rhpcctl verify client	F-10
rhpcctl add credentials	F-11
rhpcctl delete credentials	F-12
rhpcctl add database	F-12
rhpcctl addnode database	F-14
rhpcctl addpdb database	F-16
rhpcctl deletepdb database	F-17
rhpcctl delete database	F-18
rhpcctl deletenode database	F-19
rhpcctl move database	F-21
rhpcctl movepdb database	F-24
rhpcctl upgrade database	F-26
rhpcctl zdtupgrade database	F-28
rhpcctl addnode gihome	F-29
rhpcctl deletenode gihome	F-31
rhpcctl move gihome	F-32
rhpcctl upgrade gihome	F-34
rhpcctl add image	F-35
rhpcctl allow image	F-36
rhpcctl delete image	F-37
rhpcctl deploy image	F-37
rhpcctl disallow image	F-38
rhpcctl import image	F-39
rhpcctl instantiate image	F-40
rhpcctl modify image	F-41

rhpctl query image	F-42
rhpctl promote image	F-43
rhpctl unstantiate image	F-43
rhpctl add imagetype	F-44
rhpctl allow imagetype	F-45
rhpctl delete imagetype	F-45
rhpctl disallow imagetype	F-45
rhpctl modify imagetype	F-46
rhpctl query imagetype	F-46
rhpctl delete job	F-47
rhpctl query job	F-48
rhpctl collect osconfig	F-50
rhpctl compare osconfig	F-50
rhpctl disable osconfig	F-51
rhpctl enable osconfig	F-51
rhpctl query osconfig	F-52
rhpctl query peerserver	F-52
rhpctl add role	F-53
rhpctl delete role	F-54
rhpctl grant role	F-55
rhpctl query role	F-56
rhpctl revoke role	F-56
rhpctl add series	F-57
rhpctl delete series	F-57
rhpctl deleteimage series	F-58
rhpctl insertimage series	F-58
rhpctl query series	F-59
rhpctl subscribe series	F-60
rhpctl unsubscribe series	F-61
rhpctl export server	F-61
rhpctl query server	F-61
rhpctl register server	F-62
rhpctl unregister server	F-62
rhpctl delete user	F-63
rhpctl modify user	F-63
rhpctl register user	F-64
rhpctl unregister user	F-64
rhpctl add useraction	F-65
rhpctl delete useraction	F-66
rhpctl modify useraction	F-66
rhpctl query useraction	F-68

rhctl add workingcopy	F-68
rhctl addnode workingcopy	F-74
rhctl delete workingcopy	F-75
rhctl query workingcopy	F-76
rhctl update workingcopy	F-77

G Server Control (SRVCTL) Command Reference

SRVCTL Usage Information	G-1
Specifying Command Parameters as Keywords Instead of Single Letters	G-2
Character Set and Case Sensitivity of SRVCTL Object Values	G-3
Using SRVCTL Help	G-4
SRVCTL Privileges and Security	G-5
Additional SRVCTL Topics	G-5
Deprecated SRVCTL Subprograms or Commands	G-6
Single Character Parameters for all SRVCTL Commands	G-6
Miscellaneous SRVCTL Commands and Parameters	G-13
SRVCTL Command Reference	G-14
svctl add asm	G-16
svctl config asm	G-17
svctl disable asm	G-18
svctl enable asm	G-19
svctl getenv asm	G-19
svctl modify asm	G-20
svctl predict asm	G-21
svctl relocate asm	G-21
svctl remove asm	G-22
svctl setenv asm	G-23
svctl start asm	G-24
svctl status asm	G-25
svctl stop asm	G-26
svctl unsetenv asm	G-26
svctl add asmnetwork	G-27
svctl config asmnetwork	G-27
svctl modify asmnetwork	G-28
svctl remove asmnetwork	G-28
svctl add cdp	G-28
svctl config cdp	G-29
svctl disable cdp	G-29
svctl enable cdp	G-30
svctl modify cdp	G-30

svctl relocate cdp	G-30
svctl remove cdp	G-31
svctl start cdp	G-31
svctl status cdp	G-32
svctl stop cdp	G-32
svctl add cdproxy	G-32
svctl config cdproxy	G-33
svctl disable cdproxy	G-33
svctl enable cdproxy	G-34
svctl modify cdproxy	G-34
svctl relocate cdproxy	G-35
svctl remove cdproxy	G-35
svctl start cdproxy	G-36
svctl status cdproxy	G-37
svctl stop cdproxy	G-37
svctl add cvu	G-38
svctl config cvu	G-38
svctl disable cvu	G-38
svctl enable cvu	G-39
svctl modify cvu	G-39
svctl relocate cvu	G-40
svctl remove cvu	G-40
svctl start cvu	G-41
svctl status cvu	G-41
svctl stop cvu	G-42
svctl add exportfs	G-42
svctl config exportfs	G-43
svctl disable exportfs	G-44
svctl enable exportfs	G-44
svctl modify exportfs	G-45
svctl remove exportfs	G-46
svctl start exportfs	G-47
svctl status exportfs	G-47
svctl stop exportfs	G-48
svctl add filesystem	G-49
svctl config filesystem	G-51
svctl disable filesystem	G-52
svctl enable filesystem	G-52
svctl modify filesystem	G-53
svctl predict filesystem	G-54
svctl remove filesystem	G-55

svctl start filesystem	G-55
svctl status filesystem	G-56
svctl stop filesystem	G-57
svctl add gns	G-58
svctl config gns	G-60
svctl disable gns	G-61
svctl enable gns	G-61
svctl export gns	G-62
svctl import gns	G-63
svctl modify gns	G-63
svctl relocate gns	G-64
svctl remove gns	G-65
svctl start gns	G-65
svctl status gns	G-66
svctl stop gns	G-66
svctl update gns	G-67
svctl add havip	G-69
svctl config havip	G-70
svctl disable havip	G-70
svctl enable havip	G-71
svctl modify havip	G-71
svctl relocate havip	G-72
svctl remove havip	G-73
svctl start havip	G-73
svctl status havip	G-74
svctl stop havip	G-74
svctl add ioserver	G-75
svctl config ioserver	G-76
svctl disable ioserver	G-76
svctl enable ioserver	G-77
svctl getenv ioserver	G-77
svctl modify ioserver	G-77
svctl relocate ioserver	G-78
svctl remove ioserver	G-78
svctl setenv ioserver	G-79
svctl start ioserver	G-79
svctl status ioserver	G-79
svctl stop ioserver	G-80
svctl unsetenv ioserver	G-81
svctl add mgmtldb	G-81
svctl config mgmtldb	G-81

svctl disable mgmtldb	G-82
svctl enable mgmtldb	G-82
svctl getenv mgmtldb	G-83
svctl modify mgmtldb	G-83
svctl relocate mgmtldb	G-84
svctl remove mgmtldb	G-85
svctl setenv mgmtldb	G-86
svctl start mgmtldb	G-86
svctl status mgmtldb	G-87
svctl stop mgmtldb	G-88
svctl unsetenv mgmtldb	G-88
svctl add mgmtlsnr	G-89
svctl config mgmtlsnr	G-90
svctl disable mgmtlsnr	G-90
svctl enable mgmtlsnr	G-91
svctl getenv mgmtlsnr	G-91
svctl modify mgmtlsnr	G-92
svctl remove mgmtlsnr	G-92
svctl setenv mgmtlsnr	G-93
svctl start mgmtlsnr	G-94
svctl status mgmtlsnr	G-94
svctl stop mgmtlsnr	G-95
svctl unsetenv mgmtlsnr	G-95
svctl add mountfs	G-96
svctl config mountfs	G-96
svctl disable mountfs	G-97
svctl enable mountfs	G-97
svctl modify mountfs	G-98
svctl remove mountfs	G-99
svctl start mountfs	G-99
svctl status mountfs	G-100
svctl stop mountfs	G-100
svctl add netstorageservice	G-100
svctl config netstorageservice	G-101
svctl disable netstorageservice	G-101
svctl enable netstorageservice	G-102
svctl remove netstorageservice	G-102
svctl start netstorageservice	G-103
svctl status netstorageservice	G-103
svctl stop netstorageservice	G-104
svctl add ovmm	G-104

svctl config ovmm	G-105
svctl modify ovmm	G-106
svctl remove ovmm	G-106
svctl add qosmserver	G-107
svctl config qosmserver	G-107
svctl disable qosmserver	G-108
svctl enable qosmserver	G-108
svctl modify qosmserver	G-109
svctl predict qosmserver	G-109
svctl relocate qosmserver	G-109
svctl remove qosmserver	G-110
svctl start qosmserver	G-110
svctl status qosmserver	G-111
svctl stop qosmserver	G-111
svctl add rhpclient	G-112
svctl config rhpclient	G-113
svctl disable rhpclient	G-113
svctl enable rhpclient	G-114
svctl modify rhpclient	G-114
svctl relocate rhpclient	G-115
svctl remove rhpclient	G-116
svctl start rhpclient	G-116
svctl status rhpclient	G-117
svctl stop rhpclient	G-117
svctl add rhpserver	G-118
svctl config rhpserver	G-119
svctl disable rhpserver	G-119
svctl enable rhpserver	G-120
svctl modify rhpserver	G-120
svctl relocate rhpserver	G-121
svctl remove rhpserver	G-122
svctl start rhpserver	G-122
svctl status rhpserver	G-123
svctl stop rhpserver	G-123
svctl add vm	G-124
svctl check vm	G-125
svctl config vm	G-125
svctl disable vm	G-126
svctl enable vm	G-127
svctl modify vm	G-127
svctl relocate vm	G-128

svctl remove vm	G-129
svctl start vm	G-129
svctl status vm	G-130
svctl stop vm	G-131

H Oracle Clusterware Agent Framework C Application Program Interfaces

Agent Framework Data Types	H-1
Agent Framework Context Initialization and Persistence	H-2
Prototype for C and C++ Entry Point Functions	H-2
C and C++ Entry Point Types and Codes	H-2
C and C++ Entry Point Function Return Values	H-3
Multithreading Considerations	H-4
Deprecated APIs	H-4
API Reference	H-4
clsagfw_add_type()	H-5
clsagfw_check_resource()	H-5
clsagfw_create_attr_iterator()	H-6
clsagfw_delete_cookie()	H-6
clsagfw_exit2()	H-6
clsagfw_get_attr_from_iterator()	H-7
clsagfw_get_attrvalue()	H-7
clsagfw_get_check_type()	H-8
clsagfw_get_cmdid()	H-9
clsagfw_get_cookie()	H-9
clsagfw_get_request_action_name()	H-9
clsagfw_get_resource_id()	H-10
clsagfw_get_resource_name()	H-10
clsagfw_get_retry_count()	H-10
clsagfw_get_type_name()	H-11
clsagfw_init()	H-11
clsagfw_is_cmd_timedout()	H-12
clsagfw_log()	H-12
clsagfw_modify_attribute()	H-13
clsagfw_reset_attr_iterator()	H-13
clsagfw_send_status2()	H-13
clsagfw_set_cookie()	H-14
clsagfw_set_entrypoint()	H-14
clsagfw_set_exitcb()	H-15
clsagfw_set_resource_state_label()	H-15

clsagfw_startup()	H-16
Agent Example	H-16

Oracle Clusterware C Application Program Interfaces

About the Programming Interface (C API) to Oracle Clusterware	I-1
Overview	I-1
Operational Notes	I-2
Deprecated CLSCRS APIs	I-8
Changes to Existing CLSCRS APIs	I-10
Interactive CLSCRS APIs	I-10
Non-Interactive CLSCRS APIs	I-12
Command Evaluation APIs	I-15
clscrs_whatif_set_activepolicy	I-17
clscrs_whatif_fail_resource	I-18
clscrs_whatif_register_resource	I-18
clscrs_whatif_relocate_resource	I-19
clscrs_whatif_start_resource	I-20
clscrs_whatif_stop_resource	I-20
clscrs_whatif_register_serverpool	I-21
clscrs_whatif_unregister_serverpool	I-22
clscrs_whatif_add_server	I-23
clscrs_whatif_delete_server	I-24
clscrs_whatif_relocate_server	I-24
Server Categorization APIs	I-25
clscrs_servercategory_create	I-25
clscrs_servercategory_destroy	I-26
clscrs_register_servercategory	I-26
clscrs_unregister_servercategory	I-27
clscrs_get_server_by_category	I-28
clscrs_register_server	I-28
STAT3 API	I-30
clscrs_stat3	I-30
Miscellaneous APIs	I-31
clscrs_get_error_details	I-31
clscrs_request_action	I-32
clscrs_restart_resource	I-32
clscrs_start_resource_in_pools	I-33
clscrs_stop_resource_in_pools	I-34

J Oracle Cluster Registry Utility Reference

About OCRCONFIG	J-1
OCRCONFIG Command Reference	J-1
ocrconfig -add	J-2
ocrconfig -backuploc	J-3
ocrconfig -copy	J-4
ocrconfig -delete	J-5
ocrconfig -downgrade	J-5
ocrconfig -export	J-5
ocrconfig -import	J-6
ocrconfig -manualbackup	J-6
ocrconfig -overwrite	J-7
ocrconfig -repair	J-7
ocrconfig -replace	J-8
ocrconfig -restore	J-9
ocrconfig -showbackup	J-10
ocrconfig -showbackuploc	J-10
ocrconfig -upgrade	J-11
Troubleshooting Oracle Cluster Registry and Diagnostic Output	J-11
Troubleshooting Oracle Cluster Registry	J-11
Using the OCRCHECK Utility	J-12
Syntax	J-12
Examples	J-13
Using the OCRDUMP Utility to View Oracle Cluster Registry Content	J-15
OCRDUMP Utility Syntax and Options	J-16
OCRDUMP Utility Examples	J-16
Sample OCRDUMP Utility Output	J-17

K REST APIs for Oracle Clusterware

About REST APIs for Oracle Clusterware	K-1
--	-----

L Troubleshooting Oracle Clusterware

Troubleshooting an Incompatible Fleet Patching and Provisioning Client Resource	L-1
Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures	L-2
crsctl query calog	L-3
crsctl get calog maxsize	L-11
crsctl get calog retentiontime	L-11
crsctl set calog maxsize	L-11
crsctl set calog retentiontime	L-12

Oracle Clusterware Diagnostic and Alert Log Data	L-12
Diagnostics Collection Script	L-15
Storage Split in Oracle Extended Clusters	L-17
Rolling Upgrade and Driver Installation Issues	L-17
Testing Zone Delegation	L-18
Oracle Clusterware Alerts	L-19
Alert Messages Using Diagnostic Record Unique IDs	L-20

Glossary

Index

Preface

The *Oracle Clusterware Administration and Deployment Guide* describes the Oracle Clusterware architecture and provides an overview of this product. This book also describes administrative and deployment topics for Oracle Clusterware.

Information in this manual applies to Oracle Clusterware as it runs on all platforms unless otherwise noted. In addition, the content of this manual supplements administrative and deployment topics for Oracle single-instance databases that appear in other Oracle documentation. Where necessary, this manual refers to platform-specific documentation. This Preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

The *Oracle Clusterware Administration and Deployment Guide* is intended for database administrators, network administrators, and system administrators who perform the following tasks:

- Install and configure Oracle Real Application Clusters (Oracle RAC) databases
- Administer and manage Oracle RAC databases
- Manage and troubleshoot clusters and networks that use Oracle RAC

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the Oracle resources listed in this section.

- Platform-specific Oracle Clusterware and Oracle RAC installation guides
Each platform-specific Oracle Database 11g installation media contains a copy of an Oracle Clusterware and Oracle RAC platform-specific installation and configuration guide in HTML and PDF formats. These installation books contain the preinstallation, installation, and postinstallation information for the various UNIX, Linux, and Windows platforms on which Oracle Clusterware and Oracle RAC operate.
- *Oracle Real Application Clusters Administration and Deployment Guide*
This is an essential companion book that describes topics including instance management, tuning, backup and recovery, and managing services.
- *Oracle Database 2 Day DBA*
- *Oracle Database Administrator's Guide*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems*
- *Oracle Database Error Messages*

See Also:

- *Oracle Database Licensing Information User Manual* to determine whether a feature is available on your edition of Oracle Database
- *Oracle Database New Features Guide* for a complete description of the new features in this release
- *Oracle Database Upgrade Guide* for a complete description of the deprecated and desupported features in this release

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to Oracle Clusterware

Oracle Clusterware concepts and components.

Oracle Clusterware enables servers to communicate with each other, so that they appear to function as a collective unit. This combination of servers is commonly known as a cluster. Although the servers are standalone servers, each server has additional processes that communicate with other servers. In this way the separate servers appear as if they are one system to applications and end users.

This chapter includes the following topics:

- [Changes in Oracle Clusterware 20c](#)
- [Overview of Oracle Clusterware](#)
- [Understanding System Requirements for Oracle Clusterware](#)
- [Overview of Oracle Clusterware Platform-Specific Software Components](#)
- [Overview of Installing Oracle Clusterware](#)
- [Overview of Upgrading and Patching Oracle Clusterware](#)
- [Overview of Grid Infrastructure Management Repository](#)
- [Overview of Domain Services Clusters](#)
- [Overview of Managing Oracle Clusterware Environments](#)
- [Overview of Command Evaluation](#)
- [Overview of Cloning and Extending Oracle Clusterware in Grid Environments](#)
- [Overview of the Oracle Clusterware High Availability Framework and APIs](#)
- [Overview of Cluster Time Management](#)

Changes in Oracle Clusterware 20c

The following are changes in Oracle Clusterware 20c.

See Also:

- *Oracle Database Licensing Information User Manual* to determine whether a feature is available on your edition of Oracle Database
- *Oracle Database New Features Guide* for a complete description of the new features in this release
- *Oracle Database Upgrade Guide* for a complete description of the deprecated and desupported features in this release

New Features

- Common Data Model across Fleet Patching and Provisioning Servers
The common data model across Fleet Patching and Provisioning (FPP) servers provides a unified view of fleet targets regardless of the FPP server deployment.

See Also:

- [Oracle Fleet Patching and Provisioning](#)
- [RHPCTL Command Reference](#)
- *Oracle Database Rapid Home Provisioning User's Guide*

Deprecated Features in Oracle Clusterware 20c

The following features are deprecated in Oracle Clusterware 20c, and may be desupported in a future release:

- Deprecation of Policy-Managed Databases
Starting with Oracle Grid Infrastructure 20c, creation of new server pools is eliminated, and policy-managed databases are deprecated and can be desupported in a future release. Server pools will be migrated to the new Oracle Services feature that provides similar functionality.
- Deprecation of Cluster Domain - Domain Services Cluster
Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

Desupported Features in Oracle Clusterware 20c

These are the desupported features for Oracle Clusterware 20c:

- Desupport of Vendor Clusterware Integration with Oracle Clusterware
Starting with Oracle Oracle Clusterware 20c , the integration of vendor or third party clusterware with Oracle Clusterware is desupported.
- Desupport of Cluster Domain - Member Clusters
Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Members Clusters in releases previous to Oracle Grid Infrastructure 20c.
Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

Overview of Oracle Clusterware

Oracle Clusterware is portable cluster software that provides comprehensive multi-tiered high availability and resource management for consolidated environments. It supports clustering of independent servers so that they cooperate as a single system.

Oracle Clusterware is the integrated foundation for Oracle Real Application Clusters (Oracle RAC), and the high-availability and resource management framework for all applications on any major platform. Oracle Clusterware was first released with Oracle Database 10g Release 1 (10.1) as the required cluster technology for the Oracle multiinstance database, Oracle RAC. The intent is to leverage Oracle Clusterware in the cloud to provide enterprise-class resiliency where required, and dynamic, online allocation of compute resources where and when they are needed.

You can configure Oracle Clusterware to manage the availability of user applications and Oracle databases. In an Oracle RAC environment, Oracle Clusterware manages all of the resources automatically. All of the applications and processes that Oracle Clusterware manages are either **cluster resources**, or **local resources**.

Oracle Clusterware is required for using Oracle RAC; it is the only clusterware that you need for platforms on which Oracle RAC operates. Note that the servers on which you want to install and run Oracle Clusterware must use the same operating system.

Using Oracle Clusterware eliminates the need for proprietary vendor clusterware and provides the benefit of using only Oracle software. Oracle provides an entire software solution, including everything from disk management with Oracle Automatic Storage Management (Oracle ASM) to data management with Oracle Database and Oracle RAC. In addition, Oracle Database features, such as Oracle Services, provide advanced functionality when used with the underlying Oracle Clusterware high-availability framework.

Oracle Clusterware has two stored components, besides the binaries: The voting files, which record node membership information, and the Oracle Cluster Registry (OCR), which records cluster configuration information. Voting files and OCRs must reside on shared storage available to all cluster member nodes.

Benefits of Oracle Clusters

The benefits of using a cluster include:

- Scalability of applications (including Oracle RAC and Oracle RAC One databases)
- Reduce total cost of ownership for the infrastructure by providing a scalable system with low-cost commodity hardware
- Ability to fail over
- Increase throughput on demand for cluster-aware applications, by adding servers to a cluster to increase cluster resources
- Increase throughput for cluster-aware applications by enabling the applications to run on all of the nodes in a cluster
- Ability to program the startup of applications in a planned order that ensures dependent processes are started in the correct sequence
- Ability to monitor processes and restart them if they stop
- Eliminate unplanned downtime due to hardware or software malfunctions

- Reduce or eliminate planned downtime for software maintenance

Oracle Flex Clusters

Starting at Oracle Clusterware 12c release 2 (12.2), all clusters are configured as Oracle Flex Clusters. Clusters configured under older versions of Oracle Clusterware are converted in place as part of the upgrade process, including the activation of Oracle Flex ASM (which is a requirement for Oracle Flex Clusters). For information about Oracle Flex Clusters, refer to [Overview of Oracle Flex Clusters](#).

Clusterware Architectures

Note:

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Members Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

Note:

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

Oracle Clusterware provides several deployment architecture choices for new clusters during the installation process. You can either choose a *standalone cluster*, a *domain services cluster*, or a *member cluster* which is used to host applications and databases.

A **standalone cluster** hosts all Oracle Grid Infrastructure services and Oracle ASM locally and requires direct access to shared storage.

A **domain services cluster** is an Oracle Flex Cluster that has one or more Hub Nodes (for database instances) and zero or more other nodes. Shared storage is locally mounted on each of the Hub Nodes and an Oracle ASM instance is available to all Hub Nodes. In addition, a management database is stored and accessed, locally, within the cluster. This deployment is also used for an upgraded, pre-existing cluster.

A **member cluster** groups multiple cluster configurations for management purposes and makes use of shared services available within that cluster domain. The cluster configurations within that cluster domain are:

- **Domain services cluster:** A cluster that provides centralized services to other clusters within the Cluster Domain. Services can include a centralized Grid Infrastructure Management Repository (on which the management database for each of the clusters within the Cluster Domain resides), the trace file analyzer

service, an optional Fleet Patching and Provisioning service, and, very likely, a consolidated Oracle ASM storage management service.

- **Database member cluster:** A cluster that is intended to support Oracle RAC or Oracle RAC One database instances, the management database for which is off-loaded to the domain services cluster, and that can be configured with local Oracle ASM storage management or make use of the consolidated Oracle ASM storage management service offered by the domain services cluster.
- **Application member cluster:** A cluster that is configured to support applications without the resources necessary to support Oracle RAC or Oracle RAC One database instances. This cluster type has no configured local shared storage but it is intended to provide a highly available, scalable platform for running application processes.

Understanding System Requirements for Oracle Clusterware

Oracle Clusterware hardware and software concepts and requirements.

To use Oracle Clusterware, you must understand the hardware and software concepts and requirements.

Oracle Clusterware Hardware Concepts and Requirements

Understanding the hardware concepts and requirements helps ensure a successful Oracle Clusterware deployment.

A cluster consists of one or more servers. Access to an external network is the same for a server in a cluster (also known as a *cluster member* or *node*) as for a standalone server.

Note:

Many hardware providers have validated cluster configurations that provide a single part number for a cluster. If you are new to clustering, then use the information in this section to simplify your hardware procurement efforts when you purchase hardware to create a cluster.

A node that is part of a cluster requires a second network. This second network is referred to as the interconnect. For this reason, cluster member nodes require at least two network interface cards: one for a public network and one for a private network. The interconnect network is a private network using a switch (or multiple switches) that only the nodes in the cluster can access.¹

¹ Oracle Clusterware supports up to 100 nodes in a cluster on configurations running Oracle Database 10g release 2 (10.2) and later releases.

**Note:**

Oracle does not support using crossover cables as Oracle Clusterware interconnects.

Cluster size is determined by the requirements of the workload running on the cluster and the number of nodes that you have configured in the cluster. If you are implementing a cluster for high availability, then configure redundancy for all of the components of the infrastructure as follows:

- At least two network interfaces for the public network, bonded to provide one address
- At least two network interfaces for the private interconnect network

The cluster requires shared connection to storage for each server in the cluster. Oracle Clusterware supports Network File Systems (NFSs), iSCSI, Direct Attached Storage (DAS), Storage Area Network (SAN) storage, and Network Attached Storage (NAS).

To provide redundancy for storage, generally provide at least two connections from each server to the cluster-aware storage. There may be more connections depending on your I/O requirements. It is important to consider the I/O requirements of the entire cluster when choosing your storage subsystem.

Most servers have at least one local disk that is internal to the server. Often, this disk is used for the operating system binaries; you can also use this disk for the Oracle software binaries. The benefit of each server having its own copy of the Oracle binaries is that it increases high availability, so that corruption of one binary does not affect all of the nodes in the cluster simultaneously. It also allows rolling upgrades, which reduce downtime.

Oracle Clusterware Operating System Concepts and Requirements

You must first install and verify the operating system before you can install Oracle Clusterware.

Each server must have an operating system that is certified with the Oracle Clusterware version you are installing. Refer to the certification matrices available in the *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform or on My Oracle Support (formerly OracleMetaLink) for details.

When the operating system is installed and working, you can then install Oracle Clusterware to create the cluster. Oracle Clusterware is installed independently of Oracle Database. After you install Oracle Clusterware, you can then install Oracle Database or Oracle RAC on any of the nodes in the cluster.

Related Topics

- [Oracle RAC Technologies Certification Matrix for UNIX Platforms](#)
- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Oracle Clusterware Software Concepts and Requirements

Oracle Clusterware uses voting files to provide fencing and cluster node membership determination. Oracle Cluster Registry (OCR) provides cluster configuration information. Collectively, voting files and OCR are referred to as Oracle Clusterware files.

Oracle Clusterware files must be stored on Oracle ASM. If the underlying storage for the Oracle ASM disks is not *hardware protected*, such as RAID, then Oracle recommends that you configure multiple locations for OCR and voting files. The voting files and OCR are described as follows:

- **Voting Files**

Oracle Clusterware uses voting files to determine which nodes are members of a cluster. You can configure voting files on Oracle ASM, or you can configure voting files on shared storage.

If you configure voting files on Oracle ASM, then you do not need to manually configure the voting files. Depending on the redundancy of your disk group, an appropriate number of voting files are created.

If you do not configure voting files on Oracle ASM, then for high availability, Oracle recommends that you have a minimum of three voting files on physically separate storage. This avoids having a single point of failure. If you configure a single voting file, then you must use external mirroring to provide redundancy.

Oracle recommends that you do not use more than five voting files, even though Oracle supports a maximum number of 15 voting files.

- **Oracle Cluster Registry**

Oracle Clusterware uses the Oracle Cluster Registry (OCR) to store and manage information about the components that Oracle Clusterware controls, such as Oracle RAC databases, listeners, virtual IP addresses (VIPs), and services and any applications. OCR stores configuration information in a series of key-value pairs in a tree structure. To ensure cluster high availability, Oracle recommends that you define multiple OCR locations. In addition:

- You can have up to five OCR locations
- Each OCR location must reside on shared storage that is accessible by all of the nodes in the cluster
- You can replace a failed OCR location online if it is not the only OCR location
- You must update OCR through supported utilities such as Oracle Enterprise Manager, the Oracle Clusterware Control Utility (CRSCTL), the Server Control Utility (SRVCTL), the OCR configuration utility (OCRCONFIG), or the Database Configuration Assistant (DBCA)

Related Topics

- [Oracle Clusterware Configuration and Administration](#)
Configuring and administering Oracle Clusterware and its various components involves managing applications and databases, and networking within a cluster.

Oracle Clusterware Network Configuration Concepts

Oracle Clusterware enables a dynamic Oracle Grid Infrastructure through the self-management of the network requirements for the cluster.

Oracle Clusterware supports the use of [Dynamic Host Configuration Protocol \(DHCP\)](#) or stateless address auto-configuration for the VIP addresses and the [Single Client Access Name \(SCAN\)](#) address, but not the public address. DHCP provides dynamic assignment of IPv4 VIP addresses, while Stateless Address Autoconfiguration provides dynamic assignment of IPv6 VIP addresses.

The use of node VIPs is optional in a cluster deployment. By default node VIPs are included when you deploy the cluster environment.

When you are using Oracle RAC, all of the clients must be able to reach the database, which means that the clients must resolve VIP and SCAN names to all of the VIP and SCAN addresses, respectively. This problem is solved by the addition of [Grid Naming Service \(GNS\)](#) to the cluster. GNS is linked to the corporate Domain Name Service (DNS) so that clients can resolve host names to these dynamic addresses and transparently connect to the cluster and the databases. Oracle supports using GNS without DHCP or zone delegation in Oracle Clusterware 12c (as with Oracle Flex ASM server clusters, which you can configure without zone delegation or dynamic networks).



Note:

Oracle *does not* support using GNS without DHCP or zone delegation on Windows.

Oracle Clusterware 12c, a GNS instance was enhanced to enable the servicing multiple clusters rather than just one, thus only a single domain must be delegated to GNS in DNS. GNS still provides the same services as in previous versions of Oracle Clusterware.

The cluster in which the GNS server runs is referred to as the [server cluster](#). A [client cluster](#) advertises its names with the server cluster. Only one GNS daemon process can run on the server cluster. Oracle Clusterware puts the GNS daemon process on one of the nodes in the cluster to maintain availability.

In previous, single-cluster versions of GNS, the single cluster could easily locate the GNS service provider within itself. In the multicluster environment, however, the client clusters must know the GNS address of the server cluster. Given that address, client clusters can find the GNS server running on the server cluster.

In order for GNS to function on the server cluster, you must have the following:

- The DNS administrator must delegate a zone for use by GNS
- A GNS instance must be running somewhere on the network and it must not be blocked by a firewall
- All of the node names in a set of clusters served by GNS must be unique

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- [Overview of Grid Naming Service](#)
Oracle Clusterware uses Grid Naming Service (GNS) for address resolution in a single-cluster or multi-cluster environment. You can configure your clusters with a single, primary GNS instance, and you can also configure one or more secondary GNS instances with different roles to provide high availability address lookup and other services to clients.

Single Client Access Name (SCAN)

Oracle Clusterware can use the Single Client Access Name (SCAN) for dynamic VIP address configuration, removing the need to perform manual server configuration.

The SCAN is a domain name registered to at least one and up to three IP addresses, either in DNS or GNS. When using GNS and DHCP, Oracle Clusterware configures the VIP addresses for the SCAN name that is provided during cluster configuration.

The node VIP and the three SCAN VIPs are obtained from the DHCP server when using GNS. If a new server joins the cluster, then Oracle Clusterware dynamically obtains the required VIP address from the DHCP server, updates the cluster resource, and makes the server accessible through GNS.

Related Topics

- [Understanding SCAN Addresses and Client Service Connections](#)
Public network addresses are used to provide services to clients.

Manual Addresses Configuration

You have the option to manually configure addresses, instead of using GNS and DHCP for dynamic configuration.

In manual address configuration, you configure the following:

- One public address and host name for each node.
- One VIP address for each node.

You must assign a VIP address to each node in the cluster. Each VIP address must be on the same subnet as the public IP address for the node and should be an address that is assigned a name in the DNS. Each VIP address must also be unused and unpingable from within the network before you install Oracle Clusterware.

- Up to three SCAN addresses for the entire cluster.

Note:

The SCAN must resolve to at least one address on the public network. For high availability and scalability, Oracle recommends that you configure the SCAN to resolve to three addresses on the public network.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Overview of Oracle Clusterware Platform-Specific Software Components

In an operational Oracle Clusterware, various platform-specific processes or services run on each cluster node.

This section describes these processes and services.

The Oracle Clusterware Technology Stack

Oracle Clusterware consists of two separate technology stacks: an upper technology stack anchored by the Cluster Ready Services (CRS) daemon (CRSD) and a lower technology stack anchored by the Oracle High Availability Services daemon (OHASD).

These two technology stacks have several processes that facilitate cluster operations. The following sections describe these technology stacks in more detail.

The Cluster Ready Services Technology Stack

The Cluster Ready Services (CRS) technology stack leverages several processes to manage various services.

The following list describes these processes:

- **Cluster Ready Services (CRS):** The primary program for managing high availability operations in a cluster.

The CRSD manages cluster resources based on the configuration information that is stored in OCR for each resource. This includes start, stop, monitor, and failover operations. The CRSD process generates events when the status of a resource changes. When you have Oracle RAC installed, the CRSD process monitors the Oracle database instance, listener, and so on, and automatically restarts these components when a failure occurs.
- **Cluster Synchronization Services (CSS):** Manages the cluster configuration by controlling which nodes are members of the cluster and by notifying members when a node joins or leaves the cluster. If you are using certified third-party clusterware, then CSS processes interface with your clusterware to manage node membership information.

The `cssdagent` process monitors the cluster and provides I/O fencing. This service formerly was provided by Oracle Process Monitor Daemon (`oproc`), also known as `OraFenceService` on Windows. A Cluster Synchronization Services Daemon (CSSD) failure may result in Oracle Clusterware restarting the node.
- **Oracle ASM:** Provides disk management for Oracle Clusterware and Oracle Database.
- **Cluster Time Synchronization Service (CTSS):** Provides time management in a cluster for Oracle Clusterware.
- **Event Management (EVM):** A background process that publishes events that Oracle Clusterware creates.
- **Grid Naming Service (GNS):** Handles requests sent by external DNS servers, performing name resolution for names defined by the cluster.

- **Oracle Agent (oraagent):** Extends clusterware to support Oracle-specific requirements and complex resources. This process runs server callout scripts when FAN events occur. This process was known as RACG in Oracle Clusterware 11g release 1 (11.1).
- **Oracle Notification Service (ONS):** A publish and subscribe service for communicating Fast Application Notification (FAN) events.
- **Oracle Root Agent (orarootagent):** A specialized `oraagent` process that helps the CRSD manage resources owned by `root`, such as the network, and the Grid virtual IP address.

The Cluster Synchronization Service (CSS), Event Management (EVM), and Oracle Notification Services (ONS) components communicate with other cluster component layers on other nodes in the same cluster database environment. These components are also the main communication links between Oracle Database, applications, and the Oracle Clusterware high availability components. In addition, these background processes monitor and manage database operations.

The Oracle High Availability Services Technology Stack

The Oracle High Availability Services technology stack uses several processes to provide Oracle Clusterware high availability.

The following list describes the processes in the Oracle High Availability Services technology stack:

- **appagent:** Protects any resources of the `application` resource type used in previous versions of Oracle Clusterware.
- **Cluster Logger Service (ologgerd):** Receives information from all the nodes in the cluster and persists in an Oracle Grid Infrastructure Management Repository-based database. This service runs on only two nodes in a cluster.
- **Grid Interprocess Communication (GIPC):** A support daemon that enables Redundant Interconnect Usage.
- **Grid Plug and Play (GPMPD):** Provides access to the Grid Plug and Play profile, and coordinates updates to the profile among the nodes of the cluster to ensure that all of the nodes have the most recent profile.
- **Multicast Domain Name Service (mDNS):** Used by Grid Plug and Play to locate profiles in the cluster, and by GNS to perform name resolution. The mDNS process is a background process on Linux and UNIX and on Windows.
- **Oracle Agent (oraagent):** Extends clusterware to support Oracle-specific requirements and complex resources. This process manages daemons that run as the Oracle Clusterware owner, like the GIPC, GPMPD, and GIPC daemons.

Note:

This process is distinctly different from the process of the same name that runs in the Cluster Ready Services technology stack.

- **Oracle Root Agent (orarootagent):** A specialized `oraagent` process that helps the CRSD manage resources owned by `root`, such as the Cluster Health Monitor (CHM).

 **Note:**

This process is distinctly different from the process of the same name that runs in the Cluster Ready Services technology stack.

- **scriptagent**: Protects resources of resource types other than application when using shell or batch scripts to protect an application.
- **System Monitor Service (osysmond)**: The monitoring and operating system metric collection service that sends the data to the cluster logger service. This service runs on every node in a cluster.

Table 1-1 lists the processes and services associated with Oracle Clusterware components. In Table 1-1, if a UNIX or a Linux system process has an (r) beside it, then the process runs as the root user.

 **Note:**

Oracle ASM is not just one process, but an instance. Given Oracle Flex ASM, Oracle ASM does not necessarily run on every cluster node but only some of them.

Table 1-1 List of Processes and Services Associated with Oracle Clusterware Components

Oracle Clusterware Component	Linux/UNIX Process	Windows Processes
CRS	crsd.bin (r)	crsd.exe
CSS	ocssd.bin, cssdmonitor, cssdagent	cssdagent.exe, cssdmonitor.exe ocssd.exe
CTSS	octssd.bin (r)	octssd.exe
EVM	evmd.bin, evmlogger.bin	evmd.exe
GIPC	gipcd.bin	
GNS	gnsd (r)	gnsd.exe
Grid Plug and Play	gpnpd.bin	gpnpd.exe
LOGGER	ologgerd.bin (r)	ologgerd.exe
Master Diskmon	diskmon.bin	
mDNS	mdnsd.bin	mDNSResponder.exe
Oracle agent	oraagent.bin (Oracle Clusterware 12c release 1 (12.1) and 11g release 2 (11.2)), or racgmain and racgimon (Oracle Clusterware 11g release 1 (11.1))	oraagent.exe
Oracle High Availability Services	ohasd.bin (r)	ohasd.exe

Table 1-1 (Cont.) List of Processes and Services Associated with Oracle Clusterware Components

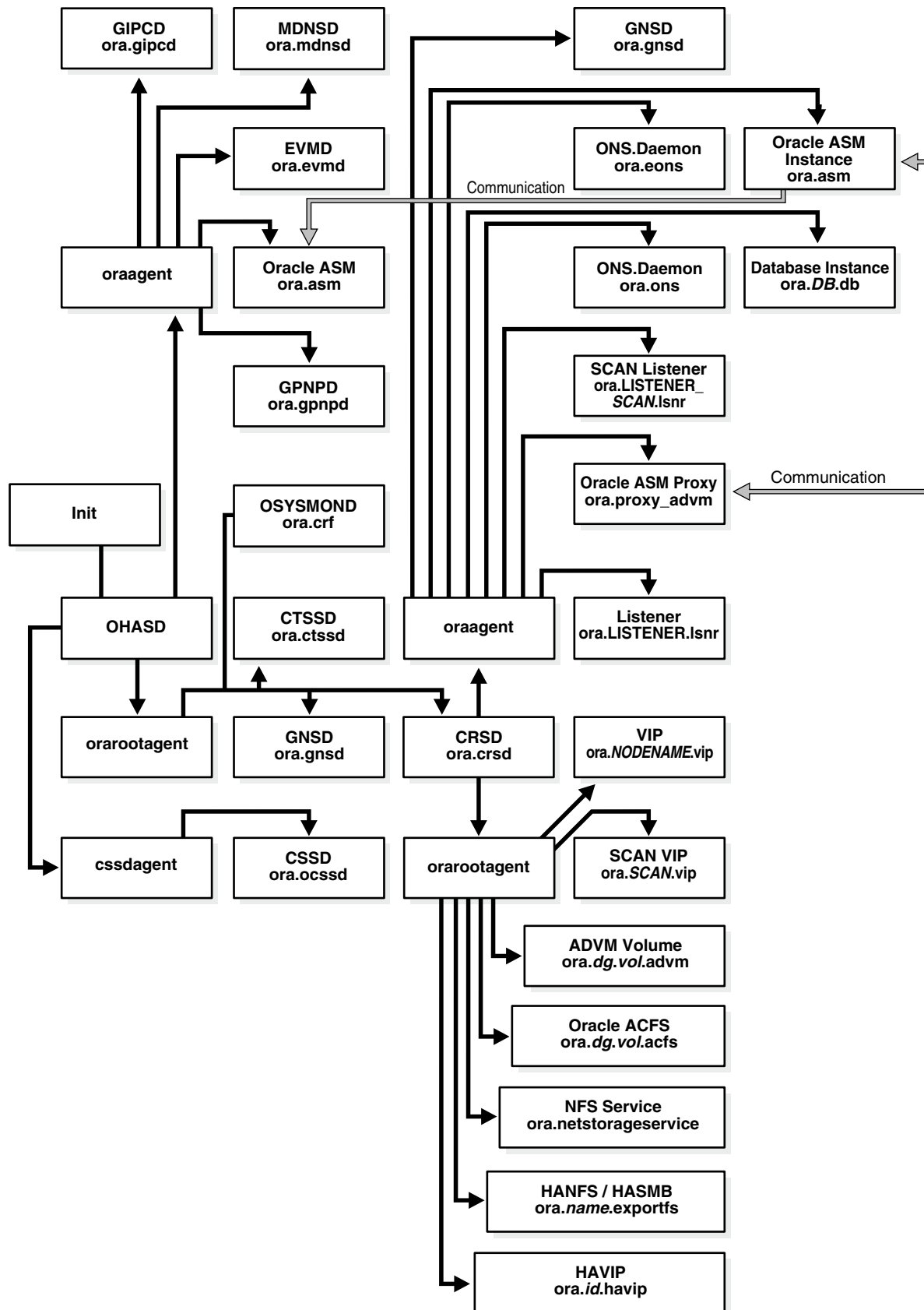
Oracle Clusterware Component	Linux/UNIX Process	Windows Processes
ONS	ons	ons.exe
Oracle root agent	orarootagent (r)	orarootagent.exe
SYSMON	osysmond.bin (r)	osysmond.exe

 **Note:**

Oracle Clusterware on Linux platforms can have multiple threads that appear as separate processes with unique process identifiers.

Figure 1-1 illustrates cluster startup.

Figure 1-1 Cluster Startup



Related Topics

- [Oracle Clusterware Resources](#)
Oracle Clusterware manages applications and processes as resources that you register with Oracle Clusterware.
- [Oracle Clusterware Diagnostic and Alert Log Data](#)
Review this content to understand clusterware-specific aspects of how Oracle Clusterware uses ADR.

Transport Layer Security Cipher Suite Management

Oracle Clusterware 19c provides CRSCTL commands to disable a given cipher suite, and stores disabled cipher suite details in Oracle Local Registry and Oracle Cluster Registry, ensuring that cipher suites included on the disabled list are not used to negotiate transport layer security.

The Oracle Clusterware technology stack uses the GIPC library for both inter-node and intra-node communication. To secure an inter-node communication channel, the GIPC library uses transport layer security. For any Oracle Clusterware release, the GIPC library supports a set of precompiled cipher suites. Over time, a cipher suite may get compromised. Prior to Oracle Clusterware 19c, there was no way to disable a given cipher suite included in the set, in order to prevent it from being used in any new connections in the future.

Querying the Cipher List

To obtain a list of available cipher suites:

```
crsctl get cluster tlsciphersuite
```

Adding a Cipher Suite to the Disabled List

To add a cipher suite to the disabled list:

```
crsctl set cluster disabledtlsciphersuite add cipher_suite_name
```

Removing a Cipher Suite from the Disabled List

To remove a cipher suite from the disabled list:

```
crsctl set cluster disabledtlsciphersuite delete cipher_suite_name
```

Related Topics

- [crsctl get cluster tlsciphersuite](#)
- [crsctl set cluster disabledtlsciphersuite](#)

Oracle Clusterware Processes on Windows Systems

Oracle Clusterware uses various Microsoft Windows processes for operations on Microsoft Windows systems.

These include the following processes:

- `mDNSResponder.exe`: Manages name resolution and service discovery within attached subnets
- `OracleOHService`: Starts all of the Oracle Clusterware daemons

Overview of Installing Oracle Clusterware

A successful deployment of Oracle Clusterware is more likely if you understand the installation and deployment concepts.



Note:

Install Oracle Clusterware with the Oracle Universal Installer.

Oracle Clusterware Version Compatibility

You can install different releases of Oracle Clusterware, Oracle ASM, and Oracle Database on your cluster. However you should be aware of compatibility considerations.

Follow these guidelines when installing different releases of software on your cluster:

- You can only have one installation of Oracle Clusterware running in a cluster, and it must be installed into its own software home (*Grid_home*). The release of Oracle Clusterware that you use must be equal to or higher than the Oracle ASM and Oracle RAC versions that run in the cluster. You cannot install a version of Oracle RAC that was released after the version of Oracle Clusterware that you run on the cluster. For example:
 - Oracle Clusterware 19c only supports Oracle ASM 19c, because Oracle ASM is in the Oracle Grid Infrastructure home, which also includes Oracle Clusterware
 - Oracle Clusterware 19c supports Oracle Database 19c and Oracle Database 12.1 Release 1 (12.1 and later)
 - Oracle ASM 19c requires Oracle Clusterware 19c, and supports Oracle Database 19c and Oracle Database 12c Release 1 (12.1 and later)
 - Oracle Database 19c requires Oracle Clusterware 19c

For example:

- * If you have Oracle Clusterware 19c installed as your clusterware, then you can have an Oracle Database 12c Release 1 (12.1) single-instance database running on one node, and separate Oracle Real Application Clusters 12c Release 2 (12.2) and Oracle Real Application Clusters 18c databases also running on the cluster.
- * When using different Oracle ASM and Oracle Database releases, the functionality of each depends on the functionality of the earlier software release. Thus, if you install Oracle Clusterware 19c, and you later configure Oracle ASM, and you use Oracle Clusterware to support an existing Oracle Database 12c Release 2 (12.2) installation, then the Oracle ASM functionality is equivalent only to that available in the 12c

Release 2 (12.2) release version. Set the compatible attributes of a disk group to the appropriate release of software in use.

- There can be multiple Oracle homes for the Oracle Database software (both single instance and Oracle RAC) in the cluster. The Oracle homes for all nodes of an Oracle RAC database must be the same.
- You can use different users for the Oracle Clusterware and Oracle Database homes if they belong to the same primary group.
- Starting with Oracle Clusterware 12c, there can only be one installation of Oracle ASM running in a cluster. Oracle ASM is always the same version as Oracle Clusterware, which must be the same (or higher) release than that of the Oracle database.
- To install Oracle RAC 10g, you must also install Oracle Clusterware.
- Oracle recommends that you do not run different cluster software on the same servers unless they are certified to work together. However, if you are adding Oracle RAC to servers that are part of a cluster, then either migrate to Oracle Clusterware or ensure that:
 - The clusterware you run is supported to run with Oracle RAC 18c.
 - You have installed the correct options for Oracle Clusterware and the other vendor clusterware to work together.

 **Note:**

Starting with Oracle Database 19c, the integration of vendor clusterware with Oracle Clusterware is deprecated, and can be desupported in a future release. For this reason, Oracle recommends that you align your next software or hardware upgrade to transition off of vendor cluster solutions.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Overview of Upgrading and Patching Oracle Clusterware

In-place patching replaces the Oracle Clusterware software with the newer version in the same Grid home. Out-of-place upgrade has both versions of the same software present on the nodes at the same time, in different Grid homes, but only one version is active.

For Oracle Clusterware 12c, Oracle supports in-place or out-of-place patching. Oracle supports only out-of-place upgrades, because Oracle Clusterware 12c must have its own, new Grid home.

Oracle supports patch bundles and one-off patches for in-place patching but only supports patch sets and major point releases as out-of-place upgrades.

Rolling upgrades avoid downtime and ensure continuous availability of Oracle Clusterware while the software is upgraded to the new version. When you upgrade to Oracle Clusterware 12c, Oracle Clusterware and Oracle ASM binaries are installed as a single binary called the Oracle Grid Infrastructure. You can upgrade Oracle

Clusterware and Oracle ASM in a rolling manner from Oracle Clusterware 11g release 2 (11.2).

Oracle supports force upgrades in cases where some nodes of the cluster are down.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Overview of Grid Infrastructure Management Repository

The Grid Infrastructure Management Repository stores information about the cluster, such as real-time performance data and metadata that various cluster clients collect and require.

Note:

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Members Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

Note:

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

In previous versions of Oracle Grid Infrastructure, the Grid Infrastructure Management Repository was a standalone database in the Grid Infrastructure home. This required disk space in the first disk group created, sharing that space with the Oracle Cluster Registry (OCR) and the voting disk. For standalone clusters, you can install the Grid Infrastructure Management Repository into its own MGMT disk group, along with the OCR backup files during Oracle Grid Infrastructure installation.

Oracle Member Clusters use the Grid Infrastructure Management Repository service located in its Domain Services Cluster, where its Grid Infrastructure Management Repository is a pluggable database (PDB) within the Grid Infrastructure Management Repository container database (CDB) of the Domain Services Cluster. This removes the requirement to run a local Grid Infrastructure Management Repository and locate its data files in the member cluster's disk group.

Additionally, you can install both the local Grid Infrastructure Management Repository and Domain Services Cluster Grid Infrastructure Management Repository into the MGMT disk group during installation of Oracle Grid Infrastructure. It is mandatory for the domain services cluster to be located in a disk group separate from the OCR and voting disk.

The Grid Infrastructure Management Repository:

- Is an Oracle database that stores real-time operating system metrics collected by Cluster Health Monitor. You configure the Grid Infrastructure Management Repository during an installation of or upgrade to Oracle Clusterware 12c on a cluster.

 **Note:**

If you are upgrading Oracle Clusterware to Oracle Clusterware 12c and OCR and the voting file are stored on raw or block devices, then you must move them to Oracle ASM before you upgrade your software.

- Must run on a Hub Node, and must support failover to another node in case of node or storage failure.
- Communicates with any cluster clients (such as Cluster Health Advisor, cluster resource activity log, Fleet Patching and Provisioning Server, OLOGGERD, and OCLUMON) through the private network. The Grid Infrastructure Management Repository communicates with external clients, Domain Services Clusters, and member clusters over the public network, only.
- Oracle recommends that you store data files in their own Oracle ASM disk group, `MGMT`, which you can create during Oracle Grid Infrastructure installation. If you did not create this disk group during installation, then the data files will be co-located with the OCR and voting files.

Oracle increased the Oracle Clusterware shared storage requirement to accommodate the Grid Infrastructure Management Repository, which can be a network file system (NFS), cluster file system, or an Oracle ASM disk group.

- Is managed with SRVCTL commands with the `mgmtdb` and `mgmtlsnr` nouns.
- Is managed by each of its clients' command one utility for retention.

 **Note:**

Starting with Oracle Grid Infrastructure 19c, the Grid Infrastructure Management Repository (GIMR) is optional for new installations of Oracle Standalone Cluster. Oracle Domain Services Clusters still require the installation of a GIMR as a service component.

Related Topics

- [Oracle Autonomous Health Framework User's Guide](#)
- [Oracle Fleet Patching and Provisioning](#)
Oracle Fleet Patching and Provisioning is a software lifecycle management method for provisioning and maintaining Oracle homes.

Overview of Domain Services Clusters

Domain Services Cluster is the base for a service-oriented infrastructure for deploying clustered systems within a single Management Domain.

 **Note:**

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Member Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

 **Note:**

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

A Domain Services Cluster is an Oracle cluster deployed to provide services to Member Clusters within that Management Domain. The Member Clusters use the configured services as they are needed. The services include a centralized Domain Management Repository, Storage Management services, and Fleet Patching and Provisioning.

Configuring a Domain Services Cluster has many benefits, including:

- Ease of deployment and provisioning
 - Configure shared services once, reuse many times over
 - No need to provision and configure shared storage for each member cluster
 - Deploy Database Member Clusters with the option of using the shared storage management services or locally configured storage
 - Deploy Application Member Clusters without support for a database, consuming fewer resources and using Oracle ACFS Remote for shared storage
- Centralized shared services
 - Benefits of consolidation of storage management to the Domain Services Clusters:
 - * Efficiencies of scale (save storage and effort)
 - * Manage all storage simultaneously
 - Database management goes from the local cluster to the Domain Management Repository
 - Member clusters have access to the Oracle ACFS file systems on the Domain Services Cluster using Oracle ACFS Remote

The Cluster Domain consists of the Domain Services Cluster and the Member Clusters that make use of the services provided. All clusters use the same standard Oracle Clusterware and are deployed using the Oracle Universal Installer. The Domain

Services Cluster must be deployed first, then used to provide services to the Member Clusters as they are deployed and configured for the services they require.

Application Member Clusters are configured, as follows:

- To not support Oracle databases and database resources, thus consuming far fewer memory resources
- Access to shared storage provisioned from the Domain Services Cluster using Oracle ACFS Remote
- Management repositories and the storage of their Oracle Cluster Registry and voting files provided by centralized storage

The Database Member Clusters can be configured in three different ways, depending upon how storage is to be accessed. In all cases, the management repositories are stored in the Domain Management Repository. The three configurations for Database Member Cluster storage access are:

- Locally configured shared storage, with local Oracle ASM instances on the cluster
- Direct access to storage that is managed through remote Oracle ASM on the Domain Services Cluster
- Indirect access to Oracle ASM storage on the Domain Services Cluster through Oracle ACFS Remote and through an Oracle ASM IO service

Related Topics

- [Cross-Cluster Dependency Proxies](#)
Cross-cluster dependency proxies are lightweight, fault-tolerant proxy resources on Member Clusters for resources running on a Domain Services Cluster.
- [Oracle Fleet Patching and Provisioning](#)
Oracle Fleet Patching and Provisioning is a software lifecycle management method for provisioning and maintaining Oracle homes.
- *Oracle Automatic Storage Management Administrator's Guide*

Overview of Managing Oracle Clusterware Environments

Oracle Clusterware provides you with several different utilities with which to manage the environment.

The following list describes the utilities for managing your Oracle Clusterware environment:

- **Cluster Health Monitor (CHM):** [Cluster Health Monitor](#) detects and analyzes operating system and cluster resource-related degradation and failures to provide more details to users for many Oracle Clusterware and Oracle RAC issues, such as node eviction. The tool continuously tracks the operating system resource consumption at the node, process, and device levels. It collects and analyzes the clusterwide data. In real-time mode, when thresholds are met, the tool shows an alert to the user. For root-cause analysis, historical data can be replayed to understand what was happening at the time of failure.
- **Cluster Verification Utility (CVU):** CVU is a command-line utility that you use to verify a range of cluster and Oracle RAC specific components. Use CVU to verify shared storage devices, networking configurations, system requirements, and Oracle Clusterware, and operating system groups and users.

Install and use CVU for both preinstallation and postinstallation checks of your cluster environment. CVU is especially useful during preinstallation and during installation of Oracle Clusterware and Oracle RAC components to ensure that your configuration meets the minimum installation requirements. Also use CVU to verify your configuration after completing administrative tasks, such as node additions and node deletions.

- **Oracle Cluster Registry Configuration Tool (OCRCONFIG):** OCRCONFIG is a command-line tool for OCR administration. You can also use the OCRCHECK and OCRDUMP utilities to troubleshoot configuration problems that affect OCR.
- **Oracle Clusterware Control (CRSCTL):** CRSCTL is a command-line tool that you can use to manage Oracle Clusterware. Use CRSCTL for general clusterware management, management of individual resources, configuration policies, and server pools for non-database applications.

Oracle Clusterware 12c introduces cluster-aware commands with which you can perform operations from any node in the cluster on another node in the cluster, or on all nodes in the cluster, depending on the operation.

You can use `crsctl` commands to monitor cluster resources (`crsctl status resource`) and to monitor and manage servers and server pools other than server pools that have names prefixed with `ora.*`, such as `crsctl status server`, `crsctl status serverpool`, `crsctl modify serverpool`, and `crsctl relocate server`. You can also manage Oracle High Availability Services on the entire cluster (`crsctl start | stop | enable | disable | config crs`), using the optional node-specific arguments `-n` or `-all`. You also can use CRSCTL to manage Oracle Clusterware on individual nodes (`crsctl start | stop | enable | disable | config crs`).

- **Oracle Enterprise Manager:** Oracle Enterprise Manager has both the Cloud Control and Grid Control GUI interfaces for managing both single instance and Oracle RAC database environments. It also has GUI interfaces to manage Oracle Clusterware and all components configured in the Oracle Grid Infrastructure installation. Oracle recommends that you use Oracle Enterprise Manager to perform administrative tasks.
- **Oracle Interface Configuration Tool (OIFCFG):** OIFCFG is a command-line tool for both single-instance Oracle databases and Oracle RAC environments. Use OIFCFG to allocate and deallocate network interfaces to components. You can also use OIFCFG to direct components to use specific network interfaces and to retrieve component configuration information.
- **Server Control (SRVCTL):** SRVCTL is a command-line interface that you can use to manage Oracle resources, such as databases, services, or listeners in the cluster.

 **Note:**

You can only use SRVCTL to manage server pools that have names prefixed with `ora.*`.

Related Topics

- [Oracle Autonomous Health Framework User's Guide](#)
- [Cluster Verification Utility Reference](#)

- [Oracle Clusterware Configuration and Administration](#)
Configuring and administering Oracle Clusterware and its various components involves managing applications and databases, and networking within a cluster.
- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)
- [Oracle Interface Configuration Tool \(OIFCFG\) Command Reference](#)
- [Server Control \(SRVCTL\) Command Reference](#)
Use the Server Control (SRVCTL) utility to manage various components and applications in your cluster.

Overview of Command Evaluation

You can use the Oracle Clusterware Control (CRSCTL) utility to evaluate what (command evaluation) will happen and why (reasoned command evaluation) when you use CRSCTL commands to manage servers, server pools, and policies within your Oracle Clusterware environment without making any actual changes.

In addition to showing you consequences of a planned or unplanned event, command evaluation is helpful in a policy-managed environment by validating any assumptions you may have about Oracle Clusterware policy decisions. Reasoned command evaluation expands on this by showing you why Oracle Clusterware performs a particular action through verbose output of various CRSCTL commands.

Following is an example of what would happen if you removed a server from a server pool:

```
$ crsctl eval delete server mjk-node2-3 -explain
```

```
Stage Group 1:
```

```
-----
--
Stage Required      Action
-----
--
1 E Server 'mjk-node2-3' is removed from server pool 'sp1'.
   E Server pool 'sp1' is below the MIN_SIZE value of 2 with 1
     servers.
   E Looking at other server pools to see whether MIN_SIZE value 2 of
     server pool 'sp1' can be met.
   E Scanning server pools with MIN_SIZE or fewer servers in
     ascending order of IMPORTANCE.
   E Considering server pools (IMPORTANCE): sp2(2) for suitable
     servers.
   E Considering server pool 'sp2' because its MIN_SIZE is 2 and it
     has 0 servers above MIN_SIZE.
   E Relocating server 'mjk-node2-0' to server pool 'sp1'.
   Y Server 'mjk-node2-3' will be removed from pools 'sp1'.
   Y Server 'mjk-node2-0' will be moved from pools 'sp2' to
     pools 'sp1'
=====
====
```

The information contained in the preceding example is what a command evaluation returns. Each action plan explains the attributes and criteria Oracle Clusterware used to arrive at the final decision.



Note:

CRSCTL can only evaluate third-party resources. Resources with the *ora* prefix, such as `ora.orcl.db`, must be evaluated using SRVCTL commands.

Related Topics

- [Command Evaluation APIs](#)
You can use the command evaluation APIs to predict Oracle Clusterware's response to a hypothetical planned or unplanned event.
- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)

Overview of Cloning and Extending Oracle Clusterware in Grid Environments

Cloning nodes is one method of creating new clusters. Use cloning to quickly create several clusters of the same configuration.

The cloning process copies Oracle Clusterware software images to other nodes that have similar hardware and software. Before using cloning, you must install an Oracle Clusterware home successfully on at least one node using the instructions in your platform-specific Oracle Clusterware installation guide.

For new installations, or if you must install on only one cluster, Oracle recommends that you use the automated and interactive installation methods, such as Oracle Universal Installer or the Provisioning Pack feature of Oracle Enterprise Manager. These methods perform installation checks to ensure a successful installation. To add or delete Oracle Clusterware to or from nodes in the cluster, use the `gridsetup.sh` script.

Related Topics

- [Cloning Oracle Clusterware](#)
- [Adding and Deleting Cluster Nodes](#)
Describes how to add nodes to an existing cluster, and how to delete nodes from clusters.

Overview of the Oracle Clusterware High Availability Framework and APIs

Oracle Clusterware provides many high availability application programming interfaces called CLSCRS APIs that you use to enable Oracle Clusterware to manage applications or processes that run in a cluster. The CLSCRS APIs enable you to provide high availability for all of your applications.

You can define a VIP address for an application to enable users to access the application independently of the node in the cluster on which the application is running. This is referred to as the application VIP. You can define multiple application VIPs, with generally one application VIP defined for each application running. The application VIP is related to the application by making it dependent on the application resource defined by Oracle Clusterware.

To maintain high availability, Oracle Clusterware components can respond to status changes to restart applications and processes according to defined high availability rules. You can use the Oracle Clusterware high availability framework by registering your applications with Oracle Clusterware and configuring the clusterware to start, stop, or relocate your application processes. That is, you can make custom applications highly available by using Oracle Clusterware to create profiles that monitor, relocate, and restart your applications.

Related Topics

- [Oracle Clusterware C Application Program Interfaces](#)

Overview of Cluster Time Management

The Cluster Time Synchronization Service (CTSS) can detect time synchronization problems between nodes in the cluster.

CTSS is installed as part of Oracle Clusterware. It runs in *observer* mode if it detects a time synchronization service (such as NTP or Chrony) or a time synchronization service configuration, valid or broken, on the system. For example, if the `etc/ntp.conf` file exists on any node in the cluster, then CTSS runs in observer mode even if no time synchronization software is running.

If CTSS detects that there is no time synchronization service or time synchronization service configuration on *any* node in the cluster, then CTSS goes into *active* mode and takes over time management for the cluster.

If CTSS is running in active mode while another, non-NTP, time synchronization software is running, then you can change CTSS to run in observer mode by creating a file called `etc/ntp.conf`. CTSS puts an entry in the alert log about the change to observer mode.

When nodes join the cluster, if CTSS is in active mode, then it compares the time on those nodes to a reference clock located on one node in the cluster. If there is a discrepancy between the two times and the discrepancy is within a certain stepping limit, then CTSS performs *step time synchronization*, which is to step the time, forward or backward, of the nodes joining the cluster to synchronize them with the reference.

Clocks on nodes in the cluster become desynchronized with the reference clock (a time CTSS uses as a basis and is on the first node started in the cluster) periodically for various reasons. When this happens, CTSS performs *slew time synchronization*, which is to speed up or slow down the system time on the nodes until they are synchronized with the reference system time. In this time synchronization method, CTSS does not adjust time backward, which guarantees monotonic increase of the system time.

When Oracle Clusterware starts, if CTSS is running in active mode and the time discrepancy is outside the stepping limit (the limit is 24 hours), then CTSS generates an alert in the alert log, exits, and Oracle Clusterware startup fails. You must manually

adjust the time of the nodes joining the cluster to synchronize with the cluster, after which Oracle Clusterware can start and CTSS can manage the time for the nodes.

When performing slew time synchronization, CTSS never runs time backward to synchronize with the reference clock. CTSS periodically writes alerts to the alert log containing information about how often it adjusts time on nodes to keep them synchronized with the reference clock.

CTSS writes entries to the Oracle Clusterware alert log and `syslog` when it:

- Detects a time change
- Detects significant time difference from the reference node
- The mode switches from observer to active or vice versa

Having CTSS running to synchronize time in a cluster facilitates troubleshooting Oracle Clusterware problems, because you will not have to factor in a time offset for a sequence of events on different nodes.

Activating and Deactivating Cluster Time Management

You can activate CTSS to assume time management services for your cluster. You can also deactivate it if you want to use a different cluster time synchronization service.

To activate CTSS in your cluster, you must stop and deconfigure the vendor time synchronization service on *all* nodes in the cluster. CTSS detects when this happens and assumes time management for the cluster.

For example, to deconfigure NTP, you must remove or rename the `etc/ntp.conf` file.

Similarly, to deactivate CTSS in your cluster:

1. Configure the vendor time synchronization service on *all* nodes in the cluster. CTSS detects this change and reverts back to observer mode.
2. Use the `crsctl check ctss` command to ensure that CTSS is operating in observer mode.
3. Start the vendor time synchronization service on *all* nodes in the cluster.
4. Use the `cluvfy comp clocksync -n all` command to verify that the vendor time synchronization service is operating.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- [crsctl check ctss](#)
- [cluvfy comp clocksync](#)

2

Oracle Clusterware Configuration and Administration

Configuring and administering Oracle Clusterware and its various components involves managing applications and databases, and networking within a cluster.

You can choose from one of two methods of configuring and administering clusters. You can use the traditional, administrator-managed approach, where you administer cluster resources and workloads manually, or you can invoke varying degrees of automated administration using a policy-managed approach.

Administrator-managed clusters requires that you manually configure how the cluster resources are deployed and where the workload is managed. Typically, this means that must configure which database instances run on what cluster nodes, by preference, and where those instances will restart in case of failures. By configuring where the database instances reside, You configure the workloads across the cluster.

With policy-managed clusters, you configure the workloads to run in server pools, for which you configure policy sets to direct how those workloads are managed in each server pool. You manage the server pools and policy sets, leaving the details of the database instance location and workload placement to the policies they have instituted. In using this approach, you have the additional option of further automating the management of the cluster by using Oracle Quality of Service Management (Oracle QoS Management).

Role-Separated Management

Role-separated management is an approach to managing cluster resources and workloads in a coordinated fashion in order to reduce the risks of resource conflicts and shortages.

Role-separated management uses operating system security and role definitions, and Oracle Clusterware access permissions to separate resource and workload management according to the user's role. This is particularly important for those working in consolidated environments, where there is likely to be competition for computing resources, and a degree of isolation is required for resource consumers and management of those resources. By default, this feature is not implemented during installation.

Configuring role-separated management consists of establishing the operating system users and groups that will administer the cluster resources (such as databases), according to the roles intended, adding the permissions on the cluster resources and server pools through access control lists (ACLs), as necessary. In addition, Oracle Automatic Storage Management (Oracle ASM) provides the capability to extend these role-separation constructs to the storage management functions.

Role-separated management principles apply equally to administrator-managed and policy-managed systems. In the case of administrator-managed, you configure the cluster resources and roles to manage them at the node level, while for policy-

managed systems you configure the cluster resources and roles to manage them in the server pools.

Role-separated management in Oracle Clusterware no longer depends on a [cluster administrator](#) (although Oracle maintains backward compatibility). By default, the user that installed Oracle Clusterware in the Oracle Grid Infrastructure home (Grid home) and `root` are *permanent* cluster administrators. Primary group privileges (`oinstall`, by default) enable database administrators to create databases in newly created server pools using the Database Configuration Assistant (DBCA), but do not enable role separation.

 **Note:**

Oracle recommends that you enable role separation before you create the first server pool in the cluster. Create and manage server pools using configuration policies and a respective policy set. Access permissions are stored for each server pool in the `ACL` attribute, described in [Table 3-1](#).

Related Topics

- [Overview of Cluster Configuration Policies and the Policy Set](#)
A cluster configuration policy is a document that contains exactly one definition for each server pool managed by the cluster configuration policy set. A cluster configuration policy set is a document that defines the names of all server pools configured for the cluster and definitions for all policies.

Managing Cluster Administrators

You use an access control list (ACL) to define administrators for the cluster.

The ability to create server pools in a cluster is limited to the cluster administrators. In prior releases, by default, every registered operating system user was considered a cluster administrator and, if necessary, the default could be changed using `crsctl add | delete crs administrator` commands. The use of these commands, however, is deprecated in this release and, instead, you should use the ACL of the policy set to control the ability to create server pools.

As a rule, to have permission to create a server pool or cluster resource, the operating system user or an operating system group of which the user is a member must have the read, write, and execute permissions set in the `ACL` attribute.

Configuring Role Separation

Role separation is the determination of the roles that are needed, the resources and server pools that they will administer, and what their access privileges should be. After you determine these, you then create or modify the operating system user accounts for group privileges (such as `oinstall` or `grid`), using the ACLs and the `CRSCTL` utility.

The most basic case is to create two operating system users as part of the `oinstall` group, then create the cluster, and two server pools. For each server pool, assign one of the operating system users to administer that server pool and exclude anyone else from all but read access to that server pool.

This requires careful planning, and disciplined, detail-oriented execution, but you can modify the configuration after implementation, to correct mistakes or make adjustments over time.

 **Note:**

You cannot apply role separation techniques to ora.* resources (Oracle RAC database resources). You can only apply these techniques to server pools and user-defined cluster resources and types.

You create the server pools or resources under the `root` or `grid` accounts. For the designated operating system users to administer these server pools or resources, they must then be given the correct permissions, enabling them to fulfill their roles.

Use the `crsctl setperm` command to configure horizontal role separation using ACLs that are assigned to server pools, resources, or both. The CRCTL utility is located in the path `Grid_home/bin`, where `Grid_home` is the Oracle Grid Infrastructure for a cluster home.

The command uses the following syntax, where the access control (ACL) string is indicated by italics:

```
crsctl setperm {resource | type | serverpool} name {-u acl_string |
-x acl_string | -o user_name | -g group_name}
```

The flag options are:

- `-u`: Update the entity ACL
- `-x`: Delete the entity ACL
- `-o`: Change the entity owner
- `-g`: Change the entity primary group

The ACL strings are:

```
{user:user_name[:readPermwritePermexecPerm] |
  group:group_name[:readPermwritePermexecPerm] |
  other[::readPermwritePermexecPerm] }
```

In the preceding syntax example:

- `user`: Designates the user ACL (access permissions granted to the designated user)
- `group`: Designates the group ACL (permissions granted to the designated group members)
- `other`: Designates the other ACL (access granted to users or groups not granted particular access permissions)
- `readperm`: Location of the read permission (`r` grants permission and `-` forbids permission)
- `writeperm`: Location of the write permission (`w` grants permission and `-` forbids permission)

- *execperm*: Location of the execute permission (x grants permission, and "-" forbids permission)

For example, to set permissions on a server pool called `psft` for the group `personnel`, where the administrative user has read/write/execute privileges, the members of the `personnel` group have read/write privileges, and users outside of the group are granted no access, enter the following command as the `root` user:

```
# crsctl setperm serverpool psft -u
user:personadmin:rw,group:personnel:rw-,
other::---
```

For cluster resources, to set permissions on an application (resource) called `MyProgram` (administered by `Maynard`) for the group `crsadmin`, where the administrative user has read, write, and execute privileges, the members of the `crsadmin` group have read and execute privileges, and users outside of the group are granted only read access (for status and configuration checks), enter the following command as whichever user originally created the resource (`root` or `grid` owner):

```
# crsctl setperm resource MyProgram -u user:Maynard:r-
x,group:crsadmin:rw-,other:---:r--
```

Related Topics

- [Oracle Clusterware Resource Reference](#)

Configuring Oracle Grid Infrastructure Using Grid Setup Wizard

Using the Configuration Wizard, you can configure a new Oracle Grid Infrastructure on one or more nodes, or configure an upgraded Oracle Grid Infrastructure. You can also run the Grid Setup Wizard in silent mode.

After performing a software-only installation of the Oracle Grid Infrastructure, you can configure the software using Grid Setup Wizard. This Wizard performs various validations of the Grid home and inputs before and after you run through the wizard.

 **Note:**

- Before running the Grid Setup Wizard, ensure that the Oracle Grid Infrastructure home is current, with all necessary patches applied.
- To launch the Grid Setup Wizard in the subsequent procedures:

On Linux and UNIX, run the following command:

```
Oracle_home/gridSetup.sh
```

On Windows, run the following command:

```
Oracle_home\gridSetup.bat
```

Configuring a Single Node

You can configure a single node by using the Configuration Wizard.

To configure a single node:

1. Start the Configuration Wizard, as follows:

```
$ Oracle_home/gridSetup.sh
```

2. On the Select Installation Option page, select **Configure Oracle Grid Infrastructure for a Cluster**.
3. On the Cluster Node Information page, select only the local node and corresponding VIP name.
4. Continue adding your information on the remaining wizard pages.
5. Review your inputs on the Summary page and click **Finish**.
6. Run the `root.sh` script as instructed by the Configuration Wizard.

Configuring Multiple Nodes

You can use the Configuration Wizard to configure multiple nodes in a cluster.

It is not necessary that Oracle Grid Infrastructure software be installed on nodes you want to configure using the Configuration Wizard.

 **Note:**

Before you launch the Configuration Wizard, ensure the following:

While software is not required to be installed on all nodes, if it is installed, then the software must be installed in the same `Grid_home` path and be at the identical level on all the nodes.

To use the Configuration Wizard to configure multiple nodes:

1. Start the Configuration Wizard, as follows:

```
$ Oracle_home/gridSetup.sh
```
2. On the Select Installation Option page, select **Configure Oracle Grid Infrastructure for a Cluster**.
3. On the Cluster Node Information page, select the nodes you want to configure and their corresponding VIP names. The Configuration Wizard validates the nodes you select to ensure that they are ready.
4. Continue adding your information on the remaining wizard pages.
5. Review your inputs on the Summary page and click **Finish**.
6. Run the `root.sh` script as instructed by the Configuration Wizard.

Upgrading Oracle Grid Infrastructure

You use the Grid Setup Wizard to upgrade a cluster's Oracle Grid Infrastructure.

To use upgrade Oracle Grid Infrastructure for a cluster:

1. Start the Grid Setup Wizard:

```
$ Oracle_home/gridSetup.sh
```
2. On the Select Installation Option page, select **Upgrade Oracle Grid Infrastructure**.
3. On the Oracle Grid Infrastructure Node Selection page, review the nodes you want to upgrade. Additionally, you can choose not to upgrade nodes that are down.
4. Continue adding your information on the remaining wizard pages.
5. Review your inputs on the Summary page and click **Finish**.
6. Run the `rootupgrade.sh` script as instructed by the Configuration Wizard.

Related Topics

- [Oracle Database Installation Guide](#)



See Also:

[Oracle Database Installation Guide](#) for your platform for Oracle Restart procedures

Running the Configuration Wizard in Silent Mode

You can run the Configuration Wizard in silent mode by specifying the `—silent` parameter.

To use the Configuration Wizard in silent mode to configure or upgrade nodes:

1. Start the Configuration Wizard from the command line, as follows:

```
$ $ORACLE_HOME/gridSetup.sh -silent -responseFile file_name
```

The Configuration Wizard validates the response file and proceeds with the configuration. If any of the inputs in the response file are found to be invalid, then the Configuration Wizard displays an error and exits.

2. Run the `root` and `Grid_home/gridSetup -executeConfigTools` scripts as prompted.

Moving and Patching an Oracle Grid Infrastructure Home

You use the Grid Setup Wizard to move and patch an Oracle Grid Infrastructure home.

The Oracle Installer script `gridSetup.sh` supports a new switch `-switchGridHome` for this purpose. This feature enables you to move and patch an Oracle Grid Infrastructure home to a newer or same patch level.

See Also:

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux* for information about switching the Oracle Grid Infrastructure home after patching

Server Weight-Based Node Eviction

You can configure the Oracle Clusterware failure recovery mechanism to choose which cluster nodes to terminate or evict in the event of a private network (cluster interconnect) failure.

In a split-brain situation, where a cluster experiences a network split, partitioning the cluster into disjoint cohorts, Oracle Clusterware applies certain rules to select the surviving cohort, potentially evicting a node that is running a critical, singleton resource.

You can affect the outcome of these decisions by adding value to a database instance or node so that, when Oracle Clusterware must decide whether to evict or terminate, it will consider these factors and attempt to ensure that all critical components remain available. You can configure weighting functions to add weight to critical components in your cluster, giving Oracle Clusterware added input when deciding which nodes to evict when resolving a split-brain situation.

You may want to ensure that specific nodes survive the tie-breaking process, perhaps because of certain hardware characteristics, or that certain resources survive, perhaps because of particular databases or services. You can assign weight to particular nodes, resources, or services, based on the following criteria:

- You can assign weight only to administrator-managed nodes.
- You can assign weight to servers or applications that are registered Oracle Clusterware resources.

Weight contributes to importance of the component and influences the choice that Oracle Clusterware makes when managing a split-brain situation. With other critical factors being equal between the various cohorts, Oracle Clusterware chooses the heaviest cohort to survive.

You can assign weight to various components, as follows:

- To assign weight to database instances or services, you use the `-css_critical yes` parameter with the `srvctl add database` or `srvctl add service` commands when adding a database instance or service. You can also use the parameter with the `srvctl modify database` and `srvctl modify service` commands.
- To assign weight to non `ora.*` resources, use the `-attr "CSS_CRITICAL=yes"` parameter with the `crsctl add resource` and `crsctl modify resource` commands when you are adding or modifying resources.
- To assign weight to a server, use the `-css_critical yes` parameter with the `crsctl set server` command.

 **Note:**

- You must restart the Oracle Clusterware stack on the node for the values to take effect. This does not apply to resources where the changes take effect without having to restart the resource.
- If you change the environment from administrator managed to policy managed, or a mixture of the two, any weight that you have assigned is stored, but is not considered, meaning that it will no longer apply or be considered unless and until you reconfigure the cluster back to being administrator managed.

Overview of Oracle Database Quality of Service Management

Oracle Database Quality of Service Management (Oracle Database QoS Management) is an automated, policy-based product that monitors the workload requests for an entire system.

Oracle Database QoS Management manages the resources that are shared across applications, and adjusts the system configuration to keep the applications running at the performance levels needed by your business. Oracle Database QoS Management responds gracefully to changes in system configuration and demand, thus avoiding additional oscillations in the performance levels of your applications.

Oracle Database QoS Management monitors and manages Oracle RAC database workload performance objectives by identifying bottlenecked resources impacting these objectives, and both recommending and taking actions to restore performance. Administrator-managed deployments bind database instances to nodes but policy-managed deployments do not, so the Oracle Database QoS Management server pool size resource control is only available for the latter. All other resource management controls are available for both deployments.

 **Note:**

Starting with Oracle Grid Infrastructure 20c, creation of new server pools is eliminated, and policy-managed databases are deprecated.

You can continue to use existing server pools. Resources using existing server pools can continue to use them transparently. However, use of cluster-ready services (CRS) configuration policies, and the CRS policy set are restricted only to manage preexisting CRS policies. Creation of new policies is disabled.

The use of CRS configuration policies and the CRS policy set can be desupported in a future release. In place of server pools and policy-managed databases, Oracle recommends that you use the new "Merged" management style.

Oracle Database QoS Management supports administrator-managed Oracle RAC and Oracle RAC One Node databases with its Measure-Only, Monitor, and Management modes. This enables schema consolidation support within an administrator-managed Oracle RAC database by adjusting the CPU shares of performance classes running in the database. Additionally, database consolidation is supported by adjusting CPU counts for databases hosted on the same physical servers.

Because administrator-managed databases do not run in server pools, the ability to expand or shrink the number of instances by changing the server pool size that is supported in policy-managed database deployments is not available for administrator-managed databases. This new deployment support is integrated into the Oracle QoS Management pages in Oracle Enterprise Manager Cloud Control.

Overview of Grid Naming Service

Oracle Clusterware uses Grid Naming Service (GNS) for address resolution in a single-cluster or multi-cluster environment. You can configure your clusters with a single, primary GNS instance, and you can also configure one or more secondary GNS instances with different roles to provide high availability address lookup and other services to clients.

- [Network Administration Tasks for GNS and GNS Virtual IP Address](#)
- [Understanding Grid Naming Service Configuration Options](#)

Network Administration Tasks for GNS and GNS Virtual IP Address

To implement GNS, your network administrator must configure the DNS to set up a domain for the cluster, and delegate resolution of that domain to the GNS VIP. You can use a separate domain, or you can create a subdomain of an existing domain for the cluster.

GNS distinguishes between nodes by using cluster names and individual node identifiers as part of the host name for that cluster node, so that cluster node 123 in cluster A is distinguishable from cluster node 123 in cluster B.

However, if you configure host names manually, then the subdomain you delegate to GNS should have no subdomains. For example, if you delegate the subdomain

mydomain.example.com to GNS for resolution, then there should be no other.mydomain.example.com domains. Oracle recommends that you delegate a subdomain to GNS that is used by GNS exclusively.

 **Note:**

You can use GNS without DNS delegation in configurations where static addressing is being done, such as in Oracle Flex ASM or Oracle Flex Clusters. However, GNS requires a domain be delegated to it if addresses are assigned using DHCP.

Example 2-1 shows DNS entries required to delegate a domain called myclustergns.example.com to a GNS VIP address 10.9.8.7.

The GNS daemon and the GNS VIP run on one node in the server cluster. The GNS daemon listens on the GNS VIP using port 53 for DNS requests. Oracle Clusterware manages the GNS daemon and the GNS VIP to ensure that they are always available. If the server on which the GNS daemon is running fails, then Oracle Clusterware fails over the GNS daemon and the GNS VIP to a surviving cluster member node. If the cluster is an Oracle Flex Cluster configuration, then Oracle Clusterware fails over the GNS daemon and the GNS VIP to a Hub Node.

 **Note:**

Oracle Clusterware does not fail over GNS addresses to different clusters. Failovers occur only to members of the same cluster.

Example 2-1 DNS Entries

```
# Delegate to gns on mycluster
mycluster.example.com NS myclustergns.example.com
#Let the world know to go to the GNS vip
myclustergns.example.com. 10.9.8.7
```

Related Topics

- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)
- [Oracle Flex Clusters](#)
An Oracle Flex Cluster scales Oracle Clusterware to large numbers of nodes.

Understanding Grid Naming Service Configuration Options

GNS can run in either *automatic* or *standard* cluster address configuration mode. Automatic configuration uses either the Dynamic Host Configuration Protocol (DHCP) for IPv4 addresses or the Stateless Address Autoconfiguration Protocol (autoconfig) (RFC 2462 and RFC 4862) for IPv6 addresses.

This section includes the following topics:

- [Highly-Available Grid Naming Service](#)
- [Automatic Configuration Option for Addresses](#)

- [Static Configuration Option for Addresses](#)
- [Shared GNS Option for Addresses](#)

Highly-Available Grid Naming Service

Highly-available GNS consists of one primary GNS instance and zero or more secondary GNS instances.

The primary GNS instance services all updates from the clients, while both the primary and the secondary GNS instances process the lookup queries. Additionally, the secondary GNS instances act as backup for the primary GNS instance. Secondary GNS instances can be promoted to the primary role whenever an existing primary GNS instance fails or is removed by the cluster administrator.

Further, highly-available GNS provides fault tolerance by taking data backup on secondary GNS instance using zone transfer. Secondary GNS instances get a copy of data from the primary GNS instance during installation. Thereafter, any update on the primary GNS instance gets replicated to the secondary GNS instances.

The primary GNS instance manages [zone data](#) and holds all records on the delegated domain. It stores the zone data and its change history in the Oracle Cluster Registry (OCR). Updating the zone data on a secondary GNS instance involves a zone transfer, which can be one of two methods:

- **Full zone transfer:** The primary GNS instance replicates all zone data to the secondary GNS instances.
- **Incremental zone transfer:** The primary GNS instance only replicates the changed data to secondary GNS instances. GNS uses this transfer mechanism for the following scenarios:
 - When there is an update to the zone data in the primary GNS instance, the instance notifies the secondary instances to initiate a data transfer. The secondary GNS instances will ask for a data transfer only if the serial number of the data in OCR of the primary GNS instance is greater than that of the data of the secondary GNS instances.
 - When the refresh time of a secondary GNS instance expires, the instance sends a query containing its data serial number to the primary GNS instance. If the serial number of the secondary GNS instance is less than that of the primary GNS instance, then GNS initiates a zone transfer.

 **Note:**

Refresh time must be long enough to reduce the load on the primary GNS instance so that answering a secondary GNS instance does not prevent the primary instance from being able to function. Default refresh time is one hour but the cluster administrator can change this value based on cluster size.

You must configure a primary GNS instance before you configure any secondary. Once you successfully configure a primary GNS instance, you export client data for clients and secondary GNS instances. You provide exported client data when you configure secondary GNS instances. All secondary GNS instances register themselves with the primary GNS instance and get a copy of zone data. Secondary GNS instances contact the primary GNS instance for data updates using the zone

transfer mechanism, when either the refresh time of the secondary GNS instance expires or in response to a notification.

Related Topics

- [Configuring Highly-Available GNS](#)
Configuring highly-available GNS involves configuring primary and secondary GNS instances. You can configure GNS during installation of Oracle Clusterware using Oracle Universal Installer but you can only configure highly-available GNS after you install Oracle Clusterware because you can only configure a secondary GNS instance after you install Oracle Clusterware.
- [Removing Primary and Secondary GNS Instances](#)
You can remove primary and secondary GNS instances from a cluster.

Automatic Configuration Option for Addresses

With automatic configurations, a DNS administrator delegates a domain on the DNS to be resolved through the GNS subdomain. During installation, Oracle Universal Installer assigns names for each cluster member node interface designated for Oracle Grid Infrastructure use during installation or configuration. SCANs and all other cluster names and addresses are resolved within the cluster, rather than on the DNS.

Automatic configuration occurs in one of the following ways:

- For IPv4 addresses, Oracle Clusterware assigns unique identifiers for each cluster member node interface allocated for Oracle Grid Infrastructure, and generates names using these identifiers within the subdomain delegated to GNS. A DHCP server assigns addresses to these interfaces, and GNS maintains address and name associations with the IPv4 addresses leased from the IPv4 DHCP pool.
- For IPv6 addresses, Oracle Clusterware automatically generates addresses with autoconfig.

Static Configuration Option for Addresses

With static configurations, no subdomain is delegated. A DNS administrator configures the GNS VIP to resolve to a name and address configured on the DNS, and a DNS administrator configures a SCAN name to resolve to three static addresses for the cluster.

A DNS administrator also configures a static public IP name and address, and virtual IP name and address for each cluster member node. A DNS administrator must also configure new public and virtual IP names and addresses for each node added to the cluster. All names and addresses are resolved by DNS.

GNS without subdomain delegation using static VIP addresses and SCANs enables Oracle Flex Cluster and CloudFS features that require name resolution information within the cluster. However, any node additions or changes must be carried out as manual administration tasks.

Shared GNS Option for Addresses

With dynamic configurations, you can configure GNS to provide name resolution for one cluster, or to advertise resolution for multiple clusters, so that a single GNS instance can perform name resolution for multiple registered clusters. This option is called shared GNS.

Shared GNS provides the same services as standard GNS, and appears the same to clients receiving name resolution. The difference is that the GNS daemon running on one cluster is configured to provide name resolution for all clusters in domains that are delegated to GNS for resolution, and GNS can be centrally managed using SRVCTL commands. You can use shared GNS configuration to minimize network administration tasks across the enterprise for Oracle Grid Infrastructure clusters.

You cannot use the static address configuration option for a cluster providing shared GNS to resolve addresses in a multi-cluster environment. Shared GNS requires automatic address configuration, either through addresses assigned by DHCP, or by IPv6 stateless address autoconfiguration.

 **Note:**

All of the node names in a set of clusters served by GNS must be unique.

Oracle Universal Installer enables you to configure static addresses with GNS for shared GNS clients or servers, with GNS used for discovery.

Administering Grid Naming Service

Use SRVCTL to administer Grid Naming Service (GNS) in both single-cluster and multi-cluster environments.

 **Note:**

The GNS server and client must run on computers using the same operating system and processor architecture. Oracle does not support running GNS on computers with different operating systems, processor architectures, or both.

This section includes the following topics:

- [Configuring Highly-Available GNS](#)
- [Removing Primary and Secondary GNS Instances](#)
- [Starting and Stopping GNS with SRVCTL](#)
- [Converting Clusters to GNS Server or GNS Client Clusters](#)
- [Moving GNS to Another Cluster](#)
- [Changing the GNS Subdomain when Moving from IPv4 to IPv6 Network](#)
- [Rolling Conversion from DNS to GNS Cluster Name Resolution](#)

Configuring Highly-Available GNS

Configuring highly-available GNS involves configuring primary and secondary GNS instances. You can configure GNS during installation of Oracle Clusterware using Oracle Universal Installer but you can only configure highly-available GNS after you

install Oracle Clusterware because you can only configure a secondary GNS instance after you install Oracle Clusterware.

Highly-available GNS provides the ability to run multiple GNS instances in multi-cluster environment with different roles. There will be one primary GNS instance and zero or more secondary instances. The primary instance manages update and resolution operations. However, the secondary instance only manages resolution operations. If you choose to use secondary GNS instances, then you must first configure a primary GNS instance or change the existing GNS in a shared GNS environment to a primary GNS instance.

1. As the cluster administrator, configure the primary GNS instance on any node in an existing cluster, as follows:

```
# srvctl add gns -vip gns_vip -domain gns_subdomain
```

2. Start the primary GNS instance, as follows:

```
# srvctl start gns
```

There can be only one primary GNS instance in a multi-cluster environment. If GNS detects more than one primary GNS instance, then it will abort the start command. You can then configure this instance as a secondary GNS instance, as described in step 5.

3. Create client data for the secondary GNS instances, as follows

```
# srvctl export gns -clientdata file_name -role secondary
```

GNS stores this zone data information in OCR and updates secondary GNS instances once they are configured.

4. Copy the client data file you created in the preceding step to the secondary GNS instance.
5. Configure the secondary GNS instance, as follows:

```
# srvctl add gns -vip gns_vip -clientdata file_name
```

6. Start the secondary GNS instance, as follows:

```
# srvctl start gns
```

Once you successfully configure and start a secondary GNS instance, it queries the primary GNS instance for the zone data.

Removing Primary and Secondary GNS Instances

You can remove primary and secondary GNS instances from a cluster.

You cannot remove a primary GNS instance until you configure another primary GNS instance. If you remove a primary GNS instance before you have another one to replace it, then GNS will not function.

As the cluster administrator, select a secondary GNS instance and promote it to primary, as follows:

```
# srvctl modify gns -role primary
```

1. Depending on whether you are removing a primary or secondary GNS instance, stop the instance, as follows:

```
# srvctl stop gns
```

2. Remove the instance, as follows:

```
# srvctl remove gns
```

 **Note:**

A DNS administrator can delete the instance's name server and A/AAAA record entries from the DNS zone data, so that the client will not attempt to query an instance which is no longer available.

Starting and Stopping GNS with SRVCTL

You use the `srvctl` command to start and stop GNS.

Start and stop GNS on the server cluster by running the following commands as `root`, respectively:

```
# srvctl start gns  
# srvctl stop gns
```

 **Note:**

You cannot start or stop GNS on a client cluster.

Converting Clusters to GNS Server or GNS Client Clusters

You can convert clusters that are not running GNS into GNS server or client clusters, and you can change GNS cluster type configurations for server and client clusters.

This section includes the following cluster conversion scenarios:

- [Converting a Non-GNS Cluster to a GNS Server Cluster](#)
- [Converting a Non-GNS Cluster to a Client Cluster](#)
- [Converting a Single Cluster Running GNS to a Server Cluster](#)
- [Converting a Single Cluster Running GNS to be a GNS Client Cluster](#)

Converting a Non-GNS Cluster to a GNS Server Cluster

You can use the `srvctl` command to convert a cluster that is not running GNS to a GNS server cluster.

1. Add GNS to the cluster by running the following command as `root`, providing a valid IP address and a domain:

```
# srvctl add gns -vip IP_address -domain domain
```

2. Start the GNS instance:

```
# srvctl start gns -node node_name
```

Note:

- Specifying a domain is *not* required when adding a GNS VIP.
- The IP address you specify cannot currently be used by another GNS instance.
- The configured cluster must have DNS delegation for it to be a GNS server cluster.

Converting a Non-GNS Cluster to a Client Cluster

To convert a cluster that is not running GNS to a client cluster, you must import the credentials file from the server cluster.

Convert a cluster that is not running GNS to a GNS client cluster, as follows:

1. Run the following command in the server cluster to export the GNS instance client data configuration to a file:

```
$ srvctl export gns -clientdata path_to_file -role client
```

You must specify the fully-qualified path to the file.

Note:

You can use the GNS configuration Client Data file you generate with Oracle Universal Installer as an input file for creating shared GNS clients.

2. Import the file you created in the preceding step on a node in the cluster to make that cluster a client cluster by running the following command, as `root`:

```
# srvctl add gns -clientdata path_to_file
```

 **Note:**

You must copy the file containing the GNS data from the server cluster to a node in the cluster where you run this command.

3. Change the SCAN name, as follows:

```
$ srvctl modify scan -scanname  
scan_name.client_cluster_name.server_GNS_subdomain
```

Converting a Single Cluster Running GNS to a Server Cluster

You do not need to do anything to convert a single cluster running GNS to be a GNS server cluster. It is automatically considered to be a server cluster when a client cluster is added.

Converting a Single Cluster Running GNS to be a GNS Client Cluster

You can use the `srvctl` command to convert a single cluster running GNS to GNS client cluster. Because it is necessary to stay connected to the current GNS during this conversion process, the procedure is more involved than that of converting a single cluster to a server cluster.

To convert a single cluster running GNS to a GNS client cluster:

1. Run the following command in the server cluster to export the GNS client information to a file:

```
$ srvctl export gns -clientdata path_to_client_data_file
```

You must specify the fully-qualified path to the file.

2. Stop GNS on the cluster you want to convert to a client cluster.

```
# srvctl stop gns
```

 **Note:**

While the conversion is in progress, name resolution using GNS will be unavailable.

3. Run the following command in the server cluster to export the GNS instance:

```
$ srvctl export gns -instance path_to_file
```

You must specify the fully-qualified path to the file.

4. Run the following command as `root` in the server cluster to import the GNS instance file:

```
# srvctl import gns -instance path_to_file
```

You must specify the fully-qualified path to the file.

5. Run the following command as `root` on the node where you imported the GNS instance file to start the GNS instance:

```
# srvctl start gns
```

By not specifying the name of the node on which you want to start the GNS instance, the instance will start on a random node.

6. Remove GNS from the GNS client cluster using the following command:

```
# srvctl remove gns
```

7. Make the former cluster a client cluster, as follows:

```
# srvctl add gns -clientdata path_to_client_data_file
```

 **Note:**

You must copy the file containing the GNS data from the server cluster to a node in the cluster where you run this command.

8. Modify the SCAN in the GNS client cluster to use the GNS subdomain qualified with the client cluster name, as follows:

```
$ srvctl modify scan -scanname scan_name.gns_domain
```

In the preceding command, *gns_domain* is in the form
client_cluster_name.server GNS subdomain

Moving GNS to Another Cluster

If it becomes necessary to make another cluster the GNS server cluster, either because a cluster failure, or because of an administration plan, then you can move GNS to another cluster with the `srvctl` command.

 **Note:**

This procedure requires server cluster and client cluster downtime. Additionally, you must import GNS client data from the new server cluster to any Oracle Flex ASM and Fleet Patching and Provisioning Servers and Clients.

To move GNS to another cluster:

1. Stop the GNS instance on the current server cluster.

```
# srvctl stop gns
```

2. Export the GNS instance configuration to a file.

```
# srvctl export gns -instance path_to_file
```

Specify the fully-qualified path to the file.

3. Remove the GNS configuration from the former server cluster.

```
# srvctl remove gns
```

4. Add GNS to the new cluster.

```
# srvctl add gns -domain domain_name -vip vip_name
```

Alternatively, you can specify an IP address for the VIP.

5. Configure the GNS instance in the new server cluster using the instance information stored in the file you created in step 2, by importing the file, as follows:

```
# srvctl import gns -instance path_to_file
```

**Note:**

The file containing the GNS data from the former server cluster must reside on the node in the cluster where you run the `srvctl import gns` command.

6. Start the GNS instance in the new server cluster.

```
# srvctl start gns
```

Changing the GNS Subdomain when Moving from IPv4 to IPv6 Network

When you move from an IPv4 network to an IPv6 network, you must change the GNS subdomain.

To change the GNS subdomain, you must add an IPv6 network, update the GNS domain, and update the SCAN, as follows:

1. Add an IPv6 subnet using the `srvctl modify network` command, as follows:

```
$ srvctl modify network -subnet ipv6_subnet/ipv6_prefix_length[/interface] -nettype autoconfig
```

2. Update the GNS domain, as follows:

```
$ srvctl stop gns -force  
$ srvctl stop scan -force  
$ srvctl remove gns -force
```

```
$ srvctl add gns -vip gns_vip -domain gns_subdomain
$ srvctl start gns
```

3. Update the SCAN name with a new domain, as follows:

```
$ srvctl remove scan -force
$ srvctl add scan -scanname new_domain
$ srvctl start scan
```

4. Convert the network IP type from IPv4 to both IPv4 DHCP and IPv6 autoconfig, as follows:

```
$ srvctl modify network -iptype both
```

5. Transition the network from using both protocols to using only IPv6 autoconfig, as follows:

```
$ srvctl modify network -iptype ipv6
```

Rolling Conversion from DNS to GNS Cluster Name Resolution

You can convert Oracle Grid Infrastructure cluster networks using DNS for name resolution to cluster networks using Grid Naming Service (GNS) obtaining name resolution through GNS.

Use the following procedure to convert from a standard DNS name resolution network to a GNS name resolution network, with no downtime:

1. Log in as the Grid user (*grid*), and use the following Cluster Verification Utility to check the status for moving the cluster to GNS, where *nodelist* is a comma-delimited list of cluster member nodes:

```
$ cluvfy stage -pre crsinst -n nodelist
```

2. As the Grid user, check the integrity of the GNS configuration using the following commands, where *domain* is the domain delegated to GNS for resolution, and *gns_vip* is the GNS VIP:

```
$ cluvfy comp gns -precrsinst -domain domain -vip gns_vip
```

3. Log in as *root*, and use the following SRVCTL command to configure the GNS resource, where *domain_name* is the domain that your network administrator has configured your DNS to delegate for resolution to GNS, and *ip_address* is the IP address on which GNS listens for DNS requests:

```
# srvctl add gns -domain domain_name -vip ip_address
```

4. Use the following command to start GNS:

```
# srvctl start gns
```

GNS starts and registers VIP and SCAN names.

5. As `root`, use the following command to change the network CRS resource to support a mixed mode of static and DHCP network addresses:

```
# srvctl modify network -nettype MIXED
```

The necessary VIP addresses are obtained from the DHCP server, and brought up.

6. As the Grid user, enter the following command to ensure that Oracle Clusterware is using the new GNS, dynamic addresses, and listener end points:

```
cluvfy stage -post crsinst -n all
```

7. After the verification succeeds, change the remote endpoints that previously used the SCAN or VIPs resolved through the DNS to use the SCAN and VIPs resolved through GNS.

For each client using a SCAN, change the SCAN that the client uses so that the client uses the SCAN in the domain delegated to GNS.

For each client using VIP names, change the VIP name on each client so that they use the same server VIP name, but with the domain name in the domain delegated to GNS.

8. Enter the following command as `root` to update the system with the SCAN name in the GNS subdomain:

```
# srvctl modify scan -scanname scan_name.gns_domain
```

In the preceding command syntax, *gns_domain* is the domain name you entered in step 3 of this procedure.

9. Disable the static addresses once all clients are using the dynamic addresses, as follows:

```
$ srvctl modify network -nettype DHCP
```

Related Topics

- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)

Node Failure Isolation

Failure isolation is a process by which a failed node is isolated from the rest of the cluster to prevent the failed node from corrupting data.

When a node fails, isolating it involves an external mechanism capable of restarting a problem node without cooperation either from Oracle Clusterware or from the operating system running on that node. To provide this capability, Oracle Clusterware supports the Intelligent Platform Management Interface specification (IPMI) (also known as Baseboard Management Controller (BMC)), an industry-standard management protocol.

Typically, you configure failure isolation using IPMI during Oracle Grid Infrastructure installation, when you are provided with the option of configuring IPMI from the Failure Isolation Support screen. If you do not configure IPMI during installation, then you can

configure it after installation using the Oracle Clusterware Control utility (CRSCTL), as described in a subsequent section.

To use IPMI for failure isolation, each cluster member node must be equipped with an IPMI device running firmware compatible with IPMI version 2.0, which supports IPMI over a local area network (LAN). In addition to BMC, `ipmiutil` must be installed on each node. The `ipmiutil` utility is not distributed with the Oracle Grid Infrastructure installation and must be downloaded. You can download `ipmiutil` from `ipmiutil.sourceforge.net` or other repositories. The minimum version of the `ipmiutil` utility required is 3.08. Running `ipmiutil` displays the version number, in addition to a version number display from the help prompt.

During database operation, failure isolation is accomplished by communication from the evicting Cluster Synchronization Services daemon to the failed node's IPMI device over the LAN. The IPMI-over-LAN protocol is carried over an authenticated session protected by a user name and password, which are obtained from the administrator during installation.

To support dynamic IP address assignment for IPMI using DHCP, the Cluster Synchronization Services daemon requires direct communication with the local IPMI device during Cluster Synchronization Services startup to obtain the IP address of the IPMI device. (This is not true for HP-UX and Solaris platforms, however, which require that the IPMI device be assigned a static IP address.) This is accomplished using an IPMI probe command (OSD), which communicates with the IPMI device through an IPMI driver, which you must install on each cluster system.

If you assign a static IP address to the IPMI device, then the IPMI driver is not strictly required by the Cluster Synchronization Services daemon. The driver *is* required, however, to use `ipmitool` or `ipmiutil` to configure the IPMI device but you can also do this with management consoles on some platforms.

Related Topics

- [Post-installation Configuration of IPMI-based Failure Isolation Using CRSCTL](#)
You use the `crsctl` command to configure IPMI-based failure isolation, after installing Oracle Clusterware. You can also use this command to modify or remove the IPMI configuration.

Server Hardware Configuration for IPMI

You must first install the `ipmiutil` binary, install and enable the IPMI driver, and configure the IPMI device, as described in the *Oracle Grid Infrastructure Installation and Upgrade Guide* for your platform.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Post-installation Configuration of IPMI-based Failure Isolation Using CRSCTL

You use the `crsctl` command to configure IPMI-based failure isolation, after installing Oracle Clusterware. You can also use this command to modify or remove the IPMI configuration.

This is described in the following topics:

- [IPMI Postinstallation Configuration with Oracle Clusterware](#)
- [Modifying IPMI Configuration Using CRSCTL](#)
- [Removing IPMI Configuration Using CRSCTL](#)

IPMI Post-installation Configuration with Oracle Clusterware

After you install the `ipmiutil` binary, install and enable the IPMI driver, configure the IPMI device, and complete the server configuration, you can use the CRSCTL command to complete IPMI configuration.

Before you started the installation, you installed the `ipmiutil` binary, installed and enabled the IPMI driver in the server operating system, and configured the IPMI hardware on each node (IP address mode, admin credentials, and so on), as described in *Oracle Grid Infrastructure Installation Guide*. When you install Oracle Clusterware, the installer collects the IPMI administrator user ID and password, and stores them in an Oracle Wallet in node-local storage, in OLR. In addition, the installer also collects information on the `ipmiutil` binary location.

After you complete the server configuration, and the configuration of the `ipmiutil` binary location, complete the following procedure on each cluster node to register IPMI administrators and passwords on the nodes.

Note:

If IPMI is configured to obtain its IP address using DHCP, it may be necessary to reset IPMI or restart the node to cause it to obtain an address.

1. Start Oracle Clusterware, which allows it to obtain the current IP address from IPMI. This confirms the ability of the clusterware to communicate with IPMI, which is necessary at startup.

If Oracle Clusterware was running before IPMI was configured, you can shut Oracle Clusterware down and restart it. Alternatively, you can use the IPMI management utility to obtain the IPMI IP address and then use CRSCTL to store the IP address in OLR by running a command similar to the following:

```
crsctl set css ipmiaddr 192.168.10.45
```

2. Use CRSCTL to set the the `ipmiutil` binary location on each node.

For example:

```
crsctl set ipmi binaryloc /usr/bin/ipmiutil
```

3. Use CRSCTL to store the previously established user ID and password for the resident IPMI in OLR by running the `crsctl set css ipmiadmin` command, and supplying password at the prompt. For example:

```
crsctl set css ipmiadmin administrator_name  
IPMI BMC password: password
```

This command validates the supplied credentials and fails if another cluster node cannot access the local IPMI using them.

After you complete hardware and operating system configuration, and register the IPMI administrator on Oracle Clusterware, IPMI-based failure isolation should be fully functional.

Before you start the installation, you must download and install `ipmiutil` on each node in the cluster, install and enable the IPMI driver in the server operating system, and configure the IPMI hardware on each node (IP address mode, admin credentials, and so on), as described in *Oracle Grid Infrastructure Installation Guide*. When you install Oracle Clusterware, the installer collects the IPMI administrator user ID and password, and stores them in an Oracle Wallet in node-local storage, in OLR.

Modifying IPMI Configuration Using CRSCTL

You may need to modify an existing IPMI-based failure isolation configuration to change IPMI passwords, or to configure IPMI for failure isolation in an existing installation. You may also need to modify the path to the `ipmiutil` binary file.

You use CRSCTL with the IPMI configuration tool appropriate to your platform to accomplish these modifications.

For example, to change the administrator password for IPMI, you must first modify the IPMI configuration as described in *Oracle Grid Infrastructure Installation and Upgrade Guide*, and then use CRSCTL to change the password in OLR.

The configuration data needed by Oracle Clusterware for IPMI is kept in an Oracle Wallet in OCR. Because the configuration information is kept in a secure store, it must be written by the Oracle Clusterware installation owner account (the Grid user), so you must log in as that installation user.

Use the following procedure to modify an existing IPMI configuration:

1. Set the location of the `ipmiutil` binary file if the binary file location has been changed. For example:

```
$ crsctl set ipmi binaryloc /usr/bin/ipmiutil
```

Before running any other CRSCTL commands that modify the IPMI configuration, the `crsctl set ipmi binaryloc` command must be run on all nodes where the binary location has changed.

2. Enter the `crsctl set css ipmiadmin administrator_name` command. For example, with the user `IPMIadm`:

```
$ crsctl set css ipmiadmin IPMIadm
```

Provide the administrator password. Oracle Clusterware stores the administrator name and password for the local IPMI in OLR.

After storing the new credentials, Oracle Clusterware can retrieve the new credentials and distribute them as required.

3. Enter the `crsctl set css ipmiaddr bmc_ip_address` command. For example:

```
$ crsctl set css ipmiaddr 192.0.2.244
```

This command stores the new IPMI IP address of the local IPMI in OLR. After storing the IP address, Oracle Clusterware can retrieve the new configuration and distribute it as required.

4. Enter the `crsctl get css ipmiaddr` command. For example:

```
$ crsctl get css ipmiaddr
```

This command retrieves the IP address for the local IPMI from OLR and displays it on the console.

5. Remove the IPMI configuration information for the local IPMI from OLR and delete the registry entry, as follows:

```
$ crsctl unset css ipmiconfig
```

Related Topics

- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)
- [Oracle RAC Environment CRSCTL Commands](#)
You can use the following commands only in an Oracle RAC environment.

Removing IPMI Configuration Using CRSCTL

You can remove an IPMI configuration from a cluster using CRSCTL if you want to stop using IPMI completely or if IPMI was initially configured by someone other than the user that installed Oracle Clusterware.

If the latter is true, then Oracle Clusterware cannot access the IPMI configuration data and IPMI is not usable by the Oracle Clusterware software, and you must reconfigure IPMI as the user that installed Oracle Clusterware.

To completely remove IPMI, perform the following steps. To reconfigure IPMI as the user that installed Oracle Clusterware, perform steps 3 and 4, then repeat steps 3 and 4 in the previous section.

1. Disable the IPMI driver and eliminate the boot-time installation, as follows:

```
/sbin/modprobe -r
```

2. Disable IPMI-over-LAN for the local IPMI using either `ipmitool` or `ipmiutil`, to prevent access over the LAN or change the IPMI administrator user ID and password.
3. Ensure that Oracle Clusterware is running and then use CRSCTL to remove the IPMI configuration data from OLR by running the following command:

```
$ crsctl unset css ipmiconfig
```

4. Restart Oracle Clusterware so that it runs without the IPMI configuration by running the following commands as `root`:

```
# crsctl stop crs  
# crsctl start crs
```

Related Topics

- [Modifying IPMI Configuration Using CRSCCTL](#)
You may need to modify an existing IPMI-based failure isolation configuration to change IPMI passwords, or to configure IPMI for failure isolation in an existing installation. You may also need to modify the path to the `ipmiutil` binary file.
- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Understanding Network Addresses on Manually Configured Networks

It is helpful to understand the concepts and requirements for network addresses on manually configured networks.

This section contains the following topics:

- [Understanding Network Address Configuration Requirements](#)
- [Understanding SCAN Addresses and Client Service Connections](#)
- [SCAN Listeners and Service Registration Restriction With Valid Node Checking](#)

Understanding Network Address Configuration Requirements

An Oracle Clusterware configuration requires at least one public network interface and one private network interface.

- A public network interface connects users and application servers to access data on the database server.
- A private network interface is for internode communication and used exclusively by Oracle Clusterware.

You can configure a public network interface for either [IPv4](#), [IPv6](#), or both types of addresses on a given network. If you use redundant network interfaces (bonded or teamed interfaces), then be aware that Oracle does not support configuring one interface to support IPv4 addresses and the other to support IPv6 addresses. You must configure network interfaces of a redundant interface pair with the same IP protocol.

You can configure one or more private network interfaces, using either IPv4 or IPv6 addresses for all the network adapters. You cannot mix IPv4 and IPv6 addresses for any private network interfaces.



Note:

You can only use IPv6 for private networks in clusters using Oracle Clusterware 12c release 2 (12.2), or later.

All the nodes in the cluster must use the same IP protocol configuration. Either all the nodes use only IPv4, or all the nodes use only IPv6, or all the nodes use both IPv4 and IPv6. You cannot have some nodes in the cluster configured to support only IPv6 addresses, and other nodes in the cluster configured to support only IPv4 addresses.

The VIP agent supports the generation of IPv6 addresses using the Stateless Address Autoconfiguration Protocol (RFC 2462), and advertises these addresses with GNS. Run the `srvctl config network` command to determine if DHCP or stateless address autoconfiguration is being used.

This section includes the following topics:

- [About IPv6 Address Formats](#)
- [Name Resolution and the Network Resource Address Type](#)

About IPv6 Address Formats

Each node in an Oracle Grid Infrastructure cluster can support both IPv4 and IPv6 addresses on the same network. The preferred IPv6 address format is as follows, where each x represents a hexadecimal character:

```
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
```

The IPv6 address format is defined by RFC 2460 and Oracle Grid Infrastructure supports IPv6 addresses, as following:

- Global and site-local IPv6 addresses as defined by RFC 4193.

Note:

Link-local and site-local IPv6 addresses as defined in RFC 1884 *are not* supported.

- The leading zeros compressed in each field of the IP address.
- Empty fields collapsed and represented by a '::' separator. For example, you could write the IPv6 address 2001:0db8:0000:0000:0000:8a2e:0370:7334 as 2001:db8::8a2e:370:7334.
- The four lower order fields containing 8-bit pieces (standard IPv4 address format). For example 2001:db8:122:344::192.0.2.33.

Name Resolution and the Network Resource Address Type

You can review the network configuration and control the network address type using the `srvctl config network` (to review the configuration) and `srvctl modify network -iptype` commands, respectively.

You can configure how addresses are acquired using the `srvctl modify network -nettype` command. Set the value of the `-nettype` parameter to `dhcp` or `static` to control how IPv4 network addresses are acquired. Alternatively, set the value of the `-nettype` parameter to `autoconfig` or `static` to control how IPv6 addresses are generated.

The `-nettype` and `-iptype` parameters are not directly related but you can use `-nettype dhcp` with `-iptype ipv4` and `-nettype autoconfig` with `-iptype ipv6`.

 **Note:**

If a network is configured with both IPv4 and IPv6 subnets, then Oracle does not support both subnets having `-nettype` set to `mixed`.

Oracle does not support making transitions from IPv4 to IPv6 while `-nettype` is set to `mixed`. You must first finish the transition from `static` to `dhcp` before you add IPv6 into the subnet.

Similarly, Oracle does not support starting a transition to IPv4 from IPv6 while `-nettype` is set to `mixed`. You must first finish the transition from `autoconfig` to `static` before you add IPv4 into the subnet.

Related Topics

- [Network Address Configuration in a Cluster](#)
You can configure a network interface for either IPv4, IPv6, or both types of addresses on a given network.

Understanding SCAN Addresses and Client Service Connections

Public network addresses are used to provide services to clients.

If your clients are connecting to the Single Client Access Name (SCAN) addresses, then you may need to change public and virtual IP addresses as you add or remove nodes from the cluster, but you do not need to update clients with new cluster addresses.

 **Note:**

You can edit the `listener.ora` file to make modifications to the Oracle Net listener parameters for SCAN and the node listener. For example, you can set `TRACE_LEVEL_listener_name`. However, you cannot set protocol address parameters to define listening endpoints, because the listener agent dynamically manages them.

SCANS function like a cluster alias. However, SCANS are resolved on any node in the cluster, so unlike a VIP address for a node, clients connecting to the SCAN no longer require updated VIP addresses as nodes are added to or removed from the cluster. Because the SCAN addresses resolve to the cluster, rather than to a node address in the cluster, nodes can be added to or removed from the cluster without affecting the SCAN address configuration.

The SCAN is a fully qualified name (host name and domain) that is configured to resolve to all the addresses allocated for the SCAN. The SCAN resolves to all three addresses configured for the SCAN name on the DNS server, or resolves within the cluster in a GNS configuration. SCAN listeners can run on any node in the cluster. SCANS provide location independence for the databases, so that client configuration does not have to depend on which nodes run a particular database.

Oracle Database 11g release 2 (11.2), and later, instances only register with SCAN listeners as remote listeners. Upgraded databases register with SCAN listeners as remote listeners, and also continue to register with all node listeners.

 **Note:**

Because of the Oracle Clusterware installation requirement that you provide a SCAN name during installation, if you resolved at least one IP address using the `server/etc/hosts` file to bypass the installation requirement but you do not have the infrastructure required for SCAN, then, after the installation, you can ignore the SCAN and connect to the databases in the cluster using VIPs.

Oracle *does not* support removing the SCAN address.

Related Topics

- [Oracle Database Net Services Reference](#)

SCAN Listeners and Service Registration Restriction With Valid Node Checking

You can use valid node checking to specify the nodes and subnets from which the SCAN listener accepts registrations. You can specify the nodes and subnet information using SRVCTL. SRVCTL stores the node and subnet information in the SCAN listener resource profile. The SCAN listener agent reads that information from the resource profile and writes it to the `listener.ora` file.

For non-cluster (single-instance) databases, the local listener accepts service registrations only from database instances on the local node. Oracle RAC releases before Oracle RAC 11g release 2 (11.2) do not use SCAN listeners, and attempt to register their services with the local listener and the listeners defined by the `REMOTE_LISTENERS` initialization parameter. To support service registration for these database instances, the default value of `valid_node_check_for_registration_alias` for the local listener in Oracle RAC 12c is set to the value `SUBNET`, rather than to the local node. To change the valid node checking settings for the node listeners, edit the `listener.ora` file.

The SCAN listener is aware of the HTTP protocol so that it can redirect HTTP clients to the appropriate handler, which can reside on different nodes in the cluster than the node on which the SCAN listener resides.

SCAN listeners must accept service registration from instances on remote nodes. For SCAN listeners, the value of `valid_node_check_for_registration_alias` is set to `SUBNET` in the `listener.ora` file so that the corresponding listener can accept service registrations that originate from the same subnet.

You can configure the listeners to accept service registrations from a different subnet. For example, you might want to configure this environment when SCAN listeners share with instances on different clusters, and nodes in those clusters are on a different subnet. Run the `srvctl modify scan_listener -invitednodes -invitedsubnets` command to include the nodes in this environment.

You must also run the `srvctl modify nodeapps -remoteservers host:port,...` command to connect the Oracle Notification Service networks of this cluster and the cluster with the invited instances.

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database Net Services Administrator's Guide*

Configuring Shared Single Client Access Names

A shared single client access name (SCAN) enables you to share one set of SCAN virtual IPs (VIPs) and listeners on a dedicated cluster with other clusters.

About Configuring Shared Single Client Access Names

You must configure the shared single client access name (SCAN) on both the database server and the database client.

Note:

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Member Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

Note:

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

The use of a shared SCAN enables multiple clusters to use a single common set of SCAN virtual IP (VIP) addresses to manage user connections, instead of deploying a set of SCAN VIPs per cluster. For example, instead of 10 clusters deploying 3 SCAN VIPs per cluster using a total of 30 IP addresses, with shared SCAN deployments, you only deploy 3 SCAN VIPs for those same 10 clusters, requiring only 3 IP addresses.

Be aware the SCAN VIPs (shared or otherwise) are required for Oracle Real Application Cluster (Oracle RAC) database clusters, but not for application member clusters or the domain services cluster.

The general procedure for configuring shared SCANS is to use the `srvctl` utility to configure first on the server (that is, the cluster that hosts the shared SCAN), then on the client (the Oracle RAC cluster that will use this shared SCAN). On the server, in addition to the configuration using `srvctl`, you must to set environment variables,

create a credential file, and ensure that the Oracle Notification Service (ONS) process that is specific to a SCAN cluster can access its own configuration directory to create and manage the ONS configuration.

Configuring the Use of Shared SCAN

Use SRVCTL to configure shared SCANS on the server that hosts the dedicated cluster, in addition to performing other necessary configuration tasks.

1. Log in to the server cluster on which you want to configure the shared SCAN.
2. Create a SCAN listener that is exclusive to this shared SCAN cluster, as follows:

```
$ srvctl add scan_listener -clientcluster cluster_name
```

3. Create a new Oracle Notification Service (ONS) resource that is specific to the server cluster.

```
$ srvctl add ons -clientcluster cluster_name
```

The `srvctl add ons` command assigns an ID to the SCAN.

4. Export the SCAN listener to the client cluster, as follows:

```
$ srvctl export scan_listener -clientcluster cluster_name -clientdata file_name
```

5. Export the ONS resource to the client cluster, as follows:

```
$ srvctl export ons -clientcluster cluster_name -clientdata file_name
```

Note:

You can use the same credential file name for both the SCAN listener and ONS. SRVCTL creates a credential file that you will use when adding these objects to the client cluster.

6. Configure shared SCAN on each cluster that will use this service.
 - a. Log in to the client cluster on which you want to configure the shared SCAN.
 - b. Add the SCAN to the client cluster, as follows:

```
$ srvctl add scan -clientdata file_name
```

- c. Create a SCAN listener that is exclusive to this client cluster, as follows:

```
$ srvctl add scan_listener -clientdata file_name
```

- d. Create an ONS resource for this cluster, as follows:

```
$ srvctl add ons -clientdata file_name
```


 **Note:**

For each of the preceding commands, specify the name of the credential file you created in the previous steps.

Changing Network Addresses on Manually Configured Systems

You can perform network address maintenance on manually configured systems.

This is described in the following topics:

- [Changing the Virtual IP Addresses Using SRVCTL](#)
- [Changing Oracle Clusterware Private Network Configuration](#)
- [Creating a Network Using SRVCTL](#)
- [Network Address Configuration in a Cluster](#)
- [Changing Static IPv4 Addresses To Static IPv6 Addresses Using SRVCTL](#)
- [Changing Dynamic IPv4 Addresses To Dynamic IPv6 Addresses Using SRVCTL](#)
- [Changing an IPv4 Network to an IPv4 and IPv6 Network](#)
- [Transitioning from IPv4 to IPv6 Networks for VIP Addresses Using SRVCTL](#)

Changing the Virtual IP Addresses Using SRVCTL

You can use SRVCTL to change a virtual IP address.

Clients configured to use public VIP addresses for Oracle Database releases before Oracle Database 11g release 2 (11.2) can continue to use their existing connection addresses. Oracle recommends that you configure clients to use SCANs, but you are not required to use SCANs. When an earlier version of Oracle Database is upgraded, it is registered with the SCAN, and clients can start using the SCAN to connect to that database, or continue to use VIP addresses for connections.

If you continue to use VIP addresses for client connections, you can modify the VIP address while Oracle Database and Oracle ASM continue to run. However, you must stop services while you modify the address. When you restart the VIP address, services are also restarted on the node.

You cannot use this procedure to change a static public subnet to use DHCP. Only the `srvctl add network -subnet` command creates a DHCP network.

 **Note:**

The following instructions describe how to change only a VIP address, and assume that the host name associated with the VIP address does not change. Note that you do not need to update VIP addresses manually if you are using GNS, and VIPs are assigned using DHCP.

If you are changing only the VIP address, then update the DNS and the client hosts files. Also, update the server hosts files, if those are used for VIP addresses.

Perform the following steps to change a VIP address:

1. Stop all services running on the node whose VIP address you want to change using the following command syntax, where *database_name* is the name of the database, *service_name_list* is a list of the services you want to stop, and *my_node* is the name of the node whose VIP address you want to change:

```
srvctl stop service -db database_name -service "service_name_list" -  
node node_name
```

The following example specifies the database name (*grid*) using the `-db` option and specifies the services (*sales,oltp*) on the appropriate node (*mynode*).

```
$ srvctl stop service -db grid -service "sales,oltp" -node mynode
```

2. Confirm the current IP address for the VIP address by running the `srvctl config vip` command. This command displays the current VIP address bound to one of the network interfaces. The following example displays the configured VIP address for a VIP named *node03-vip*:

```
$ srvctl config vip -vipname node03-vip  
VIP exists: /node03-vip/192.168.2.20/255.255.255.0/eth0
```

3. Stop the VIP resource using the `srvctl stop vip` command:

```
$ srvctl stop vip -node node_name
```

4. Verify that the VIP resource is no longer running by running the `ifconfig -a` command on Linux and UNIX systems (or issue the `ipconfig /all` command on Windows systems), and confirm that the interface (in the example it was *eth0:1*) is no longer listed in the output.
5. Make any changes necessary to the `/etc/hosts` files on all nodes on Linux and UNIX systems, or the `%windir%\system32\drivers\etc\hosts` file on Windows systems, and make any necessary DNS changes to associate the new IP address with the old host name.
6. To use a different subnet or network interface card for the default network before you change any VIP resource, you must use the `srvctl modify network -subnet subnet/netmask/interface` command as `root` to change the network resource, where *subnet* is the new subnet address, *netmask* is the new netmask, and

interface is the new interface. After you change the subnet, then you must change each node's VIP to an IP address on the new subnet, as described in the next step.

7. Modify the node applications and provide the new VIP address using the following `srvctl modify nodeapps` syntax:

```
$ srvctl modify nodeapps -node node_name -address new_vip_address
```

The command includes the following flags and values:

- `-n node_name` is the node name
- `-A new_vip_address` is the node-level VIP address: `name|ip/netmask/[if1[|if2|...]]`

For example, run the following command as the `root` user:

```
# srvctl modify nodeapps -node mynode -address  
192.168.2.125/255.255.255.0/eth0
```

Attempting to run this command as the installation owner account may result in an error. For example, if the installation owner is `oracle`, then you may see the error `PRCN-2018: Current user oracle is not a privileged user.` To avoid the error, run the command as the `root` or system administrator account.

8. Start the node VIP by running the `srvctl start vip` command:

```
$ srvctl start vip -node node_name
```

The following command example starts the VIP on the node named `mynode`:

```
$ srvctl start vip -node mynode
```

9. Repeat the steps for each node in the cluster.

Because the SRVCTL utility is a clusterwide management tool, you can accomplish these tasks for any specific node from any node in the cluster, without logging in to each of the cluster nodes.

10. Run the following command to verify node connectivity between all of the nodes for which your cluster is configured. This command discovers all of the network interfaces available on the cluster nodes and verifies the connectivity between all of the nodes by way of the discovered interfaces. This command also lists all of the interfaces available on the nodes which are suitable for use as VIP addresses.

```
$ cluvfy comp nodecon -n all -verbose
```

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*

Changing Oracle Clusterware Private Network Configuration

You can make changes to the Oracle Clusterware private network configuration.

This section describes the following topics:

- [About Private Networks and Network Interfaces](#)
- [Redundant Interconnect Usage](#)
- [Consequences of Changing Interface Names Using OIFCFG](#)
- [Changing a Network Interface](#)

About Private Networks and Network Interfaces

Oracle Clusterware requires that each node is connected through a private network (in addition to the public network). The private network connection is referred to as the cluster interconnect.

[Table 2-1](#) describes how the network interface card and the private IP address are stored.

Oracle only supports clusters in which all of the nodes use the same network interface connected to the same subnet (defined as a global interface with the `oifcfg` command). You cannot use different network interfaces for each node (node-specific interfaces).

Table 2-1 Storage for the Network Interface, Private IP Address, and Private Host Name

Entity	Stored In...	Comments
Network interface name	Operating system For example: <code>eth1</code>	You can use wildcards when specifying network interface names. For example: <code>eth*</code>
Private network Interfaces	Oracle Clusterware, in the Grid Plug and Play (GPnP) Profile	Configure an interface for use as a private interface during installation by marking the interface as Private , or use the <code>oifcfg setif</code> command to designate an interface as a private interface.

Related Topics

- [Oracle Interface Configuration Tool \(OIFCFG\) Command Reference](#)

Redundant Interconnect Usage

You can define multiple interfaces for Redundant Interconnect Usage by classifying the role of interfaces as private either during installation or after installation using the `oifcfg setif` command.

When you do, Oracle Clusterware creates from one to four (depending on the number of interfaces you define) highly available IP (HAIP) addresses, which Oracle Database and Oracle ASM instances use to ensure highly available and load balanced communications.

The Oracle software (including Oracle RAC, Oracle ASM, and Oracle ACFS, all 11g release 2 (11.2.0.2), or later), by default, uses the HAIP address of the interfaces designated with the private role as the HAIP address for all of its traffic, enabling load balancing across the provided set of cluster interconnect interfaces. If one of the

defined cluster interconnect interfaces fails or becomes non-communicative, then Oracle Clusterware transparently moves the corresponding HAIP address to one of the remaining functional interfaces.

For example, after installation, if you add a new interface to a server named `eth3` with the subnet number `172.16.2.0`, then use the following command to make this interface available to Oracle Clusterware for use as a private interface:

```
$ oifcfg setif -global eth3/172.16.2.0:cluster_interconnect
```

While Oracle Clusterware brings up a HAIP address on `eth3` of `169.254.*.*` (which is the reserved subnet for HAIP), and the database, Oracle ASM, and Oracle ACFS use that address for communication, Oracle Clusterware also uses the `172.16.2.0` address for its own communication.

Caution:

Do not use OIFCFG to classify HAIP subnets (`169.264.*.*`). You can use OIFCFG to record the interface name, subnet, and type (public, cluster interconnect, or Oracle ASM) for Oracle Clusterware. However, you cannot use OIFCFG to modify the actual IP address for each interface.

Note:

Oracle Clusterware uses at most four interfaces at any given point, regardless of the number of interfaces defined. If one of the interfaces fails, then the HAIP address moves to another one of the configured interfaces in the defined set.

When there is only a single HAIP address and multiple interfaces from which to select, the interface to which the HAIP address moves is no longer the original interface upon which it was configured. Oracle Clusterware selects the interface with the *lowest* numeric subnet to which to add the HAIP address.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Consequences of Changing Interface Names Using OIFCFG

The consequences of changing interface names depend on which name you are changing, and whether you are also changing the IP address.

In cases where you are only changing the interface names, the consequences are minor. If you change the name for the public interface that is stored in OCR, then you also must modify the node applications for the cluster. Therefore, you must stop the node applications for this change to take effect.

Changing a Network Interface

You can change a network interface and its associated subnet address by using the OIFCFG command..

This procedure changes the network interface and IP address on each node in the cluster used previously by Oracle Clusterware and Oracle Database.

▲ Caution:

The interface that the Oracle RAC (RDBMS) interconnect uses must be the same interface that Oracle Clusterware uses with the host name. Do not configure the private interconnect for Oracle RAC on a separate interface that is not monitored by Oracle Clusterware.

1. Ensure that Oracle Clusterware is running on all of the cluster nodes by running the following command:

```
$ olsnodes -s
```

The command returns output similar to the following, showing that Oracle Clusterware is running on all of the nodes in the cluster:

```
./olsnodes -s  
myclustera Active  
myclusterc Active  
myclusterb Active
```

2. Ensure that the replacement interface is configured and operational in the operating system on all of the nodes. Use the `ifconfig` command (or `ipconfig` on Windows) for your platform. For example, on Linux, use:

```
$ /sbin/ifconfig
```

3. Add the new interface to the cluster as follows, providing the name of the new interface and the subnet address, using the following command:

```
$ oifcfg setif -global if_name/subnet:cluster_interconnect
```

You can use wildcards with the interface name. For example, `oifcfg setif -global "eth*/192.168.0.0:cluster_interconnect` is valid syntax. However, be careful to avoid ambiguity with other addresses or masks used with other cluster interfaces. If you use wildcards, then you see a warning similar to the following:

```
eth*/192.168.0.0 global cluster_interconnect  
PRIF-29: Warning: wildcard in network parameters can cause mismatch  
among GPNP profile, OCR, and system
```

 **Note:**

Legacy network configuration does not support wildcards; thus wildcards are resolved using current node configuration at the time of the update.

4. If you change the Oracle ASM network, then update the Oracle ASM listener, as follows:

```
$ srvctl update listener -listener listener_name -asm -remove -force
$ srvctl add listener -listener listener_name -asmlistener -subnet
subnet
```

5. After the previous step completes, you can remove the former subnet, as follows, by providing the name and subnet address of the former interface:

```
oifcfg delif -global if_name/subnet
```

For example:

```
$ oifcfg delif -global eth1/10.10.0.0
```

 **Caution:**

This step should be performed only after a replacement interface is committed into the Grid Plug and Play configuration. Simple deletion of cluster interfaces without providing a valid replacement can result in invalid cluster configuration.

6. Verify the current configuration using the following command:

```
oifcfg getif
```

For example:

```
$ oifcfg getif
eth2 10.220.52.0 global cluster_interconnect
eth0 10.220.16.0 global public
```

7. If you change the private network, then stop Oracle Clusterware on all nodes by running the following command as `root` on each node:

```
# crsctl stop crs
```

 **Note:**

If you configured HAIP on eth0 and eth1, and you want to replace eth1 with eth3, then you do not have to stop Oracle Clusterware. If, however, you want to add another set of interfaces, such as eth2 and eth3 to your HAIP configuration, which you already configured on eth0 and eth1, then you must stop Oracle Clusterware.

8. When Oracle Clusterware stops, you can deconfigure the deleted network interface in the operating system using the `ifconfig` command. For example:

```
$ ifconfig down
```

At this point, the IP address from network interfaces for the old subnet is deconfigured from Oracle Clusterware. This command does not affect the configuration of the IP address on the operating system.

You must update the operating system configuration changes, because changes made using `ifconfig` are not persistent.

9. Restart Oracle Clusterware by running the following command on each node in the cluster as the `root` user:

```
# crsctl start crs
```

The changes take effect when Oracle Clusterware restarts.

If you use the `CLUSTER_INTERCONNECTS` initialization parameter, then you must update it to reflect the changes.

Related Topics

- [Oracle Interface Configuration Tool \(OIFCFG\) Command Reference](#)

Creating a Network Using SRVCTL

You can use SRVCTL to create a network for a cluster member node, and to add application configuration information.

Create a network for a cluster member node, as follows:

1. Log in as `root`.
2. Add a node application to the node, using the following syntax, where:

```
srvctl add nodeapps -node node_name -address {vip |  
  addr}/netmask[/if1|if2|...] [-pingtarget "ping_target_list"]
```

In the preceding syntax:

- *node_name* is the name of the node
- *vip* is the VIP name or *addr* is the IP address
- *netmask* is the netmask

- `if1[|if2|...]` is a pipe (|)-delimited list of interfaces bonded for use by the application
- `ping_target_list` is a comma-delimited list of IP addresses or host names to ping

 **Note:**

- Use the `-pingtarget` parameter when link status monitoring does not work as it does in a virtual machine environment.
- Enter the `srvctl add nodeapps -help` command to review other syntax options.

In the following example of using `srvctl add nodeapps` to configure an IPv4 node application, the node name is `node1`, the netmask is `255.255.252.0`, and the interface is `eth0`:

```
# srvctl add nodeapps -node node1 -address node1-  
vip.mycluster.example.com/255.255.252.0/eth0
```

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*

Network Address Configuration in a Cluster

You can configure a network interface for either IPv4, IPv6, or both types of addresses on a given network.

If you configure redundant network interfaces using a third-party technology, then Oracle does not support configuring one interface to support IPv4 addresses and the other to support IPv6 addresses. You must configure network interfaces of a redundant interface pair with the same IP address type. If you use the Oracle Clusterware Redundant Interconnect feature, then you must use IPv4 addresses for the interfaces.

All the nodes in the cluster must use the same IP protocol configuration. Either all the nodes use only IPv4, or all the nodes use only IPv6, or all the nodes use both IPv4 and IPv6. You cannot have some nodes in the cluster configured to support only IPv6 addresses, and other nodes in the cluster configured to support only IPv4 addresses.

The local listener listens on endpoints based on the address types of the subnets configured for the network resource. Possible types are IPV4, IPV6, or both.

Changing Static IPv4 Addresses To Static IPv6 Addresses Using SRVCTL

When you change from IPv4 static addresses to IPv6 static addresses, you add an IPv6 address and modify the network to briefly accept both IPv4 and IPv6 addresses, before switching to using static IPv6 addresses, only.

 **Note:**

If the IPv4 network is in mixed mode with both static and dynamic addresses, then you cannot perform this procedure. You must first transition all addresses to static.

To change a static IPv4 address to a static IPv6 address:

1. Add an IPv6 subnet using the following command as `root` once for the entire network:

```
# srvctl modify network -subnet ipv6_subnet/prefix_length
```

In the preceding syntax *ipv6_subnet/prefix_length* is the subnet of the IPv6 address to which you are changing along with the prefix length, such as `3001::/64`.

2. Add an IPv6 VIP using the following command as `root` once on each node:

```
# srvctl modify vip -node node_name -netnum network_number -address vip_name/netmask
```

In the preceding syntax:

- *node_name* is the name of the node
- *network_number* is the number of the network
- *vip_name/netmask* is the name of a local VIP that resolves to both IPv4 and IPv6 addresses

The IPv4 netmask or IPv6 prefix length that follows the VIP name must satisfy two requirements:

- If you specify a netmask in IPv4 format (such as `255.255.255.0`), then the VIP name resolves to IPv4 addresses (but can also resolve to IPv6 addresses). Similarly, if you specify an IPv6 prefix length (such as `64`), then the VIP name resolves to IPv6 addresses (but can also resolve to IPv4 addresses).
- If you specify an IPv4 netmask, then it should match the netmask of the registered IPv4 network subnet number, regardless of whether the `-iptype` of the network is IPv6. Similarly, if you specify an IPv6 prefix length, then it must match the prefix length of the registered IPv6 network subnet number, regardless of whether the `-iptype` of the network is IPv4.

3. Add the IPv6 network resource to OCR using the following command:

```
$ oifcfg setif -global if_name/subnet:public
```

4. Update the SCAN in DNS to have as many IPv6 addresses as there are IPv4 addresses. Add IPv6 addresses to the SCAN VIPs using the following command as root once for the entire network:

```
# srvctl modify scan -scanname scan_name
```

scan_name is the name of a SCAN that resolves to both IPv4 and IPv6 addresses.

5. Convert the network IP type from IPv4 to both IPv4 and IPv6 using the following command as root once for the entire network:

```
srvctl modify network -netnum network_number -iptype both
```

This command brings up the IPv6 static addresses.

6. Change all clients served by the cluster from IPv4 networks and addresses to IPv6 networks and addresses.
7. Transition the network from using both protocols to using only IPv6 using the following command:

```
# srvctl modify network -iptype ipv6
```

8. Modify the VIP using a VIP name that resolves to IPv6 by running the following command as root:

```
# srvctl modify vip -node node_name -address vip_name -netnum  
network_number
```

Do this once for each node.

9. Modify the SCAN using a SCAN name that resolves to IPv6 by running the following command:

```
# srvctl modify scan -scanname scan_name
```

Do this once for the entire cluster.

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*
- [OIFCFG Command Format](#)

Changing Dynamic IPv4 Addresses To Dynamic IPv6 Addresses Using SRVCTL

You change dynamic IPv4 addresses to dynamic IPv6 addresses by the SRVCTL command.

 **Note:**

If the IPv4 network is in mixed mode with both static and dynamic addresses, then you cannot perform this procedure. You must first transition all addresses to dynamic.

To change dynamic IPv4 addresses to dynamic IPv6 addresses:

1. Add an IPv6 subnet using the `srvctl modify network` command.

To add the IPv6 subnet, log in as `root` and use the following command syntax:

```
# srvctl modify network -netnum network_number -subnet ipv6_subnet/  
ipv6_prefix_length[/interface] -nettype autoconfig
```

In the preceding syntax:

- *network_number* is the number of the network
- *ipv6_subnet* is the subnet of the IPv6 addresses to which you are changing (for example, 2001:db8:122:344:c0:2:2100::)
- *ipv6_prefix_length* is the prefix specifying the IPv6 network addresses (for example, 64)

For example, the following command modifies network 3 by adding an IPv6 subnet, 2001:db8:122:344:c0:2:2100::, and the prefix length 64:

```
# srvctl modify network -netnum 3 -subnet  
2001:db8:122:344:c0:2:2100::/64  
-nettype autoconfig
```

2. Add the IPv6 network resource to OCR using the following command:

```
$ oifcfg setif -global if_name/subnet:public
```

3. Start the IPv6 dynamic addresses, as follows:

```
# srvctl modify network -netnum network_number -iptype both
```

For example, on network number 3:

```
# srvctl modify network -netnum 3 -iptype both
```

4. Change all clients served by the cluster from IPv4 networks and addresses to IPv6 networks and addresses.

At this point, the SCAN in the GNS-delegated domain *scan_name.gns_domain* will resolve to three IPv4 and three IPv6 addresses.

5. Turn off the IPv4 part of the dynamic addresses on the cluster using the following command:

```
# srvctl modify network -iptype ipv6
```

After you run the preceding command, the SCAN (*scan_name.gns_domain*) will resolve to only three IPv6 addresses.

Related Topics

- [OIFCFG Command Format](#)

Changing an IPv4 Network to an IPv4 and IPv6 Network

You can change an IPv4 network to an IPv4 and IPv6 network by adding an IPv6 network to an existing IPv4 network.

This process is described in steps 1 through 5 of the procedure documented in "[Changing Static IPv4 Addresses To Static IPv6 Addresses Using SRVCTL](#)".

After you complete those steps, log in as the Grid user, and run the following command:

```
$ srvctl status scan
```

Review the output to confirm the changes to the SCAN VIPs.

Transitioning from IPv4 to IPv6 Networks for VIP Addresses Using SRVCTL

You use the SRVCTL command to remove an IPv4 address type from a combined IPv4 and IPv6 network.

Enter the following command:

```
# srvctl modify network -iptype ipv6
```

This command starts the removal process of IPv4 addresses configured for the cluster.

Cross-Cluster Dependency Proxies

Cross-cluster dependency proxies are lightweight, fault-tolerant proxy resources on Member Clusters for resources running on a Domain Services Cluster.

 **Note:**

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Member Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

 **Note:**

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

Member Clusters reduce the overhead of having infrastructure resources and it is important to be able to effectively monitor the state of the shared infrastructure resources, such as Oracle Automatic Storage Management (Oracle ASM), on a Domain Services Cluster, so that resources on Member Clusters can properly adjust their states.

Cross-cluster dependency proxies provide this functionality for Domain Services Cluster resources, specifically, and, more generally, to reflect the state of resources running on one cluster, in other clusters. Cross-cluster dependency proxies are configured, by default, on Domain Services Clusters.

3

Policy-Based Cluster and Capacity Management

Oracle Clusterware uses *policy-based management* of servers and resources used by Oracle databases or applications.

This chapter includes the following topics:

- [Overview of Server Pools and Policy-Based Management](#)
- [Default Server Pools](#)
- [Overview of Server Categorization](#)
- [Overview of Cluster Configuration Policies and the Policy Set](#)
- [Server Weight-Based Node Eviction](#)
- [Load-Aware Resource Placement](#)
- [Server Configuration and Server State Attributes](#)
- [Server Category Attributes](#)
- [An Example Policy Set Configuration](#)

Overview of Server Pools and Policy-Based Management

Resources managed by Oracle Clusterware are contained in logical groups of servers called server pools. This was introduced in Oracle Clusterware 11g release 2 (11.2).

[Resources](#) are hosted on a shared infrastructure and are contained within [server pools](#). Examples of resources that Oracle Clusterware manages are database instances, database services, application VIPs, and application components.

In an Oracle Cluster you can use server pools to run particular types of workloads on cluster member nodes, while providing simplified administration options. You can use a [cluster configuration policy set](#) to provide dynamic management of cluster policies across the cluster.

You can continue to manage resources in an Oracle Clusterware standard Cluster by using the Oracle Clusterware 11g release 2 (11.2) server pool model, or you can manually manage resources by using the traditional fixed, non-server pool method.

This section includes the following topics:

- [Server Pools and Server Categorization](#)
- [Server Pools and Policy-Based Management](#)
- [How Server Pools Work](#)
- [Default Server Pools](#)
- [Server Pool Attributes](#)
- [How Oracle Clusterware Assigns New Servers Using Server Pools](#)

- [Managing Server Pools Using Default Attributes](#)

Server Pools and Server Categorization

Administrators can deploy and manage servers dynamically using server pools by identifying servers distinguished by particular attributes, a process called [server categorization](#). In this way, you can create clusters made up of heterogeneous nodes.

Related Topics

- [Overview of Server Categorization](#)
Server categorization enables you to organize servers into particular categories by using attributes such as processor types, memory, and other distinguishing system features.

Server Pools and Policy-Based Management

With policy-based management, administrators specify the server pool (excluding the Generic and Free pools) in which the servers run.

For example, a database administrator uses SRVCTL to create a server pool for servers hosting a database or database service. A clusterware administrator uses CRSCTL to create server pools for non-database use, such as creating a server pool for servers hosting an application.

Policy-based management:

- Enables online server reallocation based on a defined policy to satisfy workload capacity requirements
- Guarantees the allocation of required resources for critical work as defined by the policy
- Ensures isolation where necessary, so that you can provide dedicated servers in a cluster for applications and databases
- Enables policies to be configured to change pools in accordance with business needs or application demand, so that pools provide the required capacity at the right time

Server pools provide resource isolation to prevent applications running in one server pool from accessing resources running in another server pool. Oracle Clusterware provides fine-grained role separation between server pools. This capability maintains required management role separation between these groups in organizations that have clustered environments managed by separate groups.

Oracle Clusterware efficiently allocates servers in the cluster. Server pool attributes, defined when the server pool is created, dictate placement and prioritization of servers based on the `IMPORTANCE` server pool attribute.

Related Topics

- [Oracle Clusterware Resource Reference](#)
- [Overview of Cluster Configuration Policies and the Policy Set](#)
A cluster configuration policy is a document that contains exactly one definition for each server pool managed by the cluster configuration policy set. A cluster configuration policy set is a document that defines the names of all server pools configured for the cluster and definitions for all policies.

How Server Pools Work

Server pools divide the cluster into logical groups of servers hosting both singleton and uniform applications. The application can be a database service or a non-database application. An application is *uniform* when the application workload is distributed over all servers in the server pool. An application is *singleton* when it runs on a single server within the server pool. Oracle Clusterware role-separated management determines access to and use of a server pool.

You manage server pools that contain Oracle RAC databases with the Server Control (SRVCTL) utility. Use the Oracle Clusterware Control (CRSCTL) utility to manage all other server pools. Only cluster administrators have permission to create top-level server pools.

Database administrators use the Server Control (SRVCTL) utility to create and manage server pools that will contain Oracle RAC databases. Cluster administrators use the Oracle Clusterware Control (CRSCTL) utility to create and manage all other server pools, such as server pools for non-database applications. Only cluster administrators have permission to create top-level server pools.

Top-level server pools:

- Logically divide the cluster
- Are always exclusive, meaning that one server can only reside in one particular server pool at a certain point in time

Default Server Pools

When Oracle Clusterware is installed, two internal server pools are created automatically: *Generic* and *Free*. All servers in a new installation are assigned to the Free server pool, initially. Servers move from Free to newly defined server pools automatically.

The Free Server Pool

The Free server pool contains servers that are not assigned to any other server pools. The attributes of the Free server pool are restricted, as follows:

- `SERVER_NAMES`, `MIN_SIZE`, and `MAX_SIZE` cannot be edited by the user
- `IMPORTANCE` and `ACL` can be edited by the user

The Generic Server Pool

The Generic server pool stores any server that is not in a top-level server pool and is not policy managed. Servers that host non-policy-managed applications, such as administrator-managed databases, are statically assigned to the Generic server pool.

The Generic server pool's attributes are restricted, as follows:

- No one can modify configuration attributes of the Generic server pool (all attributes are read-only)
- You can only create administrator-managed databases in the Generic Pool, if the server you want to create the database on is one of the following:

- Online and exists in the Generic server pool
- Online and exists in the Free server pool, in which case Oracle Clusterware moves the server into the Generic server pool
- Online and exists in any other server pool *and* the user is either a cluster administrator or is allowed to use the server pool's servers, in which case, the server is moved into the Generic server pool
- Offline and the user is a cluster administrator

Server Pool Attributes

You can use SRVCTL or CRSCTL to create server pools for databases and other applications, respectively.

If you use SRVCTL to create a server pool, then you can only use a subset of the server pool attributes described in this section. If you use CRSCTL to create server pools, then you can use the entire set of server pool attributes.

Server pool attributes are the attributes that you define to create and manage server pools.

The decision about which utility to use is based upon the type of resource being hosted in the server pool. You must use SRVCTL to create server pools that host Oracle databases. You must use CRSCTL to create server pools that host non-database resources such as middle tiers and applications.

When you use SRVCTL to create a server pool, the server pool attributes available to you are:

```
-category  
-importance  
-min  
-max  
-serverpool  
-servers
```

SRVCTL prepends "ora." to the name of the server pool.

When you use CRSCTL to create a server pool, all attributes listed and described in the following table are available to you.

Table 3-1 Server Pool Attributes

Attribute	Description
ACL	<p>String in the following format:</p> <pre data-bbox="743 407 1263 434">owner:user:rwx,pgrp:group:rwx,other::r-</pre> <p>Defines the owner of the server pool and which privileges are granted to various operating system users and groups. The server pool owner defines the operating system user of the owner, and which privileges that user is granted.</p> <p>The value of this optional attribute is populated at the time a server pool is created based on the ACL of the user creating the server pool, unless explicitly overridden. The value can subsequently be changed, if such a change is allowed based on the existing privileges of the server pool.</p> <p>In the string:</p> <ul data-bbox="743 793 1453 1087" style="list-style-type: none"> • <code>owner</code>: The operating system user of the server pool owner, followed by the privileges of the owner • <code>pgrp</code>: The operating system group that is the primary group of the owner of the server pool, followed by the privileges of members of the primary group • <code>other</code>: Followed by privileges of others • <code>r</code>: Read only • <code>w</code>: Modify attributes of the pool or delete it • <code>x</code>: Assign resources to this pool <p>By default, the identity of the client that creates the server pool is the <code>owner</code>. Also by default, <code>root</code>, and the <code>user</code> specified in <code>owner</code> have full privileges. You can grant required operating system users and operating system groups their privileges by adding the following lines to the <code>ACL</code> attribute:</p>
ACTIVE_SERVERS	<p>A string of server names in the following format:</p> <pre data-bbox="743 1656 1127 1684">server_name1 server_name2 ...</pre> <p>Oracle Clusterware automatically manages this attribute, which contains the space-delimited list of servers that are currently assigned to a server pool.</p>

Table 3-1 (Cont.) Server Pool Attributes

Attribute	Description
EXCLUSIVE_POOLS	<p>This optional attribute indicates if servers assigned to this server pool are shared with other server pools. A server pool can explicitly state that it is mutually exclusive of any other server pool that has the same value for this attribute. Two or more server pools are mutually exclusive when the sets of servers assigned to them do not have a single server in common. For example, server pools A and B must be mutually exclusive if they both have the value of this attribute set to the same string, such as <code>pools_A_B</code>.</p> <p>Top-level server pools are mutually exclusive, by default.</p>
IMPORTANCE	<p>Relative importance of the server pool, expressed as an integer from 0 to 1000, with 0 denoting the lowest level of importance and 1000, the highest. This optional attribute is used to determine how to reconfigure the server pools when a node joins or leaves the cluster. The default value is 0.</p>
MIN_SIZE	<p>The minimum size of a server pool expressed as any nonnegative integer. If the number of servers contained in a server pool is below the number you specify in this attribute, then Oracle Clusterware automatically moves servers from other pools into this one until that number is met.</p> <p>Note: The value of this optional attribute does not set a hard limit. It governs the priority for server assignment whenever the cluster is reconfigured. The default value is 0.</p>
MAX_SIZE	<p>The maximum number of servers a server pool can contain, expressed as any nonnegative integer. This attribute is optional and is set to -1 (no limit), by default.</p> <p>Note: A value of -1 for this attribute spans the entire cluster.</p>
NAME	<p>The name of the server pool, which you must specify when you create the server pool. Server pool names must be unique within the domain of names of user-created entities, such as resources, types, and servers. A server pool name has a 254 character limit and can contain any platform-supported characters except the exclamation point (!), the tilde (~), and spaces. A server pool name cannot begin with a period nor with <i>ora</i>.</p> <p>Note: When you create server pools using SRVCTL, the utility prepends "ora." to the name of the server pool.</p>
PARENT_POOLS	<p>Use of this attribute makes it possible to create nested server pools. Server pools, listed as a string of space-delimited server pool names, in this attribute are referred to as <i>parent</i> server pools. A server pool included in a parent server pool is referred to as a <i>child</i> server pool.</p> <p>Note: If you use SRVCTL to create the server pool, then you cannot specify this attribute.</p>

Table 3-1 (Cont.) Server Pool Attributes

Attribute	Description
SERVER_CATEGORY	<p>The name of a registered server category, used as part of server categorization. Oracle Clusterware standard clusters and Oracle Flex Clusters have a default category of <code>hub</code>. When you create a server pool, if you set a value for <code>SERVER_CATEGORY</code>, then you cannot set a value for <code>SERVER_NAMES</code>. Only one of these parameters may have a non-empty value.</p> <p>Use the <code>SERVER_CATEGORY</code> attribute to classify servers assigned to a server pool based on server attributes. You can organize servers and server pools in a cluster to match specific workload to servers and server pools, based on server attributes that you define.</p>
SERVER_NAMES	<p>A list of candidate node names, expressed as a string of space-delimited server names, that may be associated with a server pool. If you do not provide a set of server names for this optional attribute, then Oracle Clusterware is configured so that any server may be assigned to any server pool, to the extent allowed by values of other attributes, such as <code>PARENT_POOLS</code>.</p> <p>The server names identified as candidate node names are not validated to confirm that they are currently active cluster members. Use this attribute to define servers as candidates that have not yet been added to the cluster.</p> <p>If you set a value for <code>SERVER_NAMES</code>, then you cannot set a value for <code>SERVER_CATEGORY</code>; Only one of these attributes may have a non-empty value.</p> <p>Note: If you set the <code>SERVER_CATEGORY</code> attribute and you need to specify individual servers, then you can list servers by name using the <code>EXPRESSION</code> server category attribute.</p>

Related Topics

- [crsctl add serverpool](#)
- [crsctl status server](#)
- [Server Category Attributes](#)
- *Oracle Real Application Clusters Administration and Deployment Guide*

How Oracle Clusterware Assigns New Servers Using Server Pools

Oracle Clusterware assigns new servers to server pools in the following order:

1. Generic server pool
2. User-created server pool
3. Free server pool

Oracle Clusterware continues to assign servers to server pools until the following conditions are met:

1. Until all server pools are filled in order of importance to their minimum (`MIN_SIZE`).
2. Until all server pools are filled in order of importance to their maximum (`MAX_SIZE`).
3. By default, any servers not placed in a server pool go into the Free server pool.

You can modify the `IMPORTANCE` attribute for the Free server pool. If the value of the `IMPORTANCE` attribute of the Free server pool is greater than one or more of the other server pools, then the Free server pool will receive any remaining servers once the value of their `MIN_SIZE` attribute is met.

When a server joins a cluster, several things occur.

Consider the server pools configured in [Table 3-2](#):

Table 3-2 Sample Server Pool Attributes Configuration

NAME	IMPORTANCE	MIN_SIZE	MAX_SIZE	PARENT_POOLS	EXCLUSIVE_POOLS
sp1	1	1	10		
sp2	3	1	6		
sp3	2	1	2		
sp2_1	2	1	5	sp2	S123
sp2_2	1	1	5	sp2	S123

For example, assume that there are no servers in a cluster; all server pools are empty.

When a server, named `server1`, joins the cluster:

1. Server-to-pool assignment commences.
2. Oracle Clusterware only processes top-level server pools (those that have no parent server pools), first. In this example, the top-level server pools are `sp1`, `sp2`, and `sp3`.
3. Oracle Clusterware lists the server pools in order of `IMPORTANCE`, as follows: `sp2`, `sp3`, `sp1`.
4. Oracle Clusterware assigns `server1` to `sp2` because `sp2` has the highest `IMPORTANCE` value and its `MIN_SIZE` value has not yet been met.
5. Oracle Clusterware processes the remaining two server pools, `sp2_1` and `sp2_2`. The sizes of both server pools are below the value of the `MIN_SIZE` attribute (both server pools are empty and have `MIN_SIZE` values of 1).
6. Oracle Clusterware lists the two remaining pools in order of `IMPORTANCE`, as follows: `sp2_1`, `sp2_2`.
7. Oracle Clusterware assigns `server1` to `sp2_1` but cannot assign `server1` to `sp2_2` because `sp2_1` is configured to be exclusive with `sp2_2`.

After processing, the cluster configuration appears, as follows

Table 3-3 Post Processing Server Pool Configuration

Server Pool Name	Assigned Servers
sp1	
sp2	server1
sp3	
sp2_1	server1
sp2_2	

Servers Moving from Server Pool to Server Pool

If the number of servers in a server pool falls below the value of the `MIN_SIZE` attribute for the server pool (such as when a server fails), based on values you set for the `MIN_SIZE` and `IMPORTANCE` attributes for all server pools, Oracle Clusterware can move servers from other server pools into the server pool whose number of servers has fallen below the value for `MIN_SIZE`. Oracle Clusterware selects servers from other server pools to move into the deficient server pool that meet the following criteria:

- For server pools that have a lower `IMPORTANCE` value than the deficient server pool, Oracle Clusterware can take servers from those server pools even if it means that the number of servers falls below the value for the `MIN_SIZE` attribute.
- For server pools with equal or greater `IMPORTANCE`, Oracle Clusterware only takes servers from those server pools if the number of servers in a server pool is greater than the value of its `MIN_SIZE` attribute.

Managing Server Pools Using Default Attributes

By default, each server pool is configured with the following attribute options for managing server pools:

- `MIN_SIZE`: The minimum number of servers the server pool should contain.
If the number of servers in a server pool is below the value of this attribute, then Oracle Clusterware automatically moves servers from elsewhere into the server pool until the number of servers reaches the attribute value.
- `MAX_SIZE`: The maximum number of servers the server pool should contain.
- `IMPORTANCE`: A number from 0 to 1000 (0 being least important) that ranks a server pool among all other server pools in a cluster.

In addition, you can assign additional attributes to provide more granular management of server pools, as part of a cluster configuration policy. Attributes such as `EXCLUSIVE_POOLS` and `SERVER_CATEGORY` can assist you to create policies for your server pools that enhance performance and build tuning design management into your server pool.

Overview of Server Categorization

Server categorization enables you to organize servers into particular categories by using attributes such as processor types, memory, and other distinguishing system features.

You can configure server pools to restrict eligible members of the pool to a category of servers, which share a particular set of attributes. Originally, server pools were restricted to a set of basic attributes characterizing servers as belonging to a given pool, with no way to distinguish between types of servers; all servers were considered to be equal in relation to their processors, physical memory, and other characteristics. Server categorization now provides a way to distinguish these servers.

Note:

If you create policies with Oracle Database Quality of Service Management (Oracle Database QoS Management), then you categorize servers by setting server pool directive overrides, and CRSCTL commands using the `policy` and `policyset` nouns are disabled. Also if you switch from using Oracle Clusterware policies to using Oracle Database QoS Management policies, then the Oracle Clusterware policies are replaced, because the two policy types cannot coexist. Oracle recommends that you create a backup using `crsctl status policyset -file file_name` before you switch policies.

Related Topics

- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)

Overview of Cluster Configuration Policies and the Policy Set

A cluster configuration policy is a document that contains exactly one definition for each server pool managed by the cluster configuration policy set. A cluster configuration policy set is a document that defines the names of all server pools configured for the cluster and definitions for all policies.

Note:

Oracle Clusterware 11g release 2 (11.2) supports only a single server pool configuration. You must manually make any changes to the server pool configuration when you want the change to take effect.

In Oracle Clusterware 12c, you use the policies defined in the [cluster configuration policy set](#) for server pool specification and management, and Oracle Clusterware manages the server pools according to the policies in the policy set. With a cluster configuration policy set, for example, you can allocate more servers to OLTP workload during weekly business hours to respond to email demand, and on the weekends and

evenings, allocate more servers to batch workloads, and perform transitions of server pool configuration or server allocation, atomically.

At any point in time, only one policy is in effect for the cluster. But you can create several different policies, so that you can configure pools of servers with parameters to reflect differences in requirements for the cluster based on business needs or demand, or based on calendar dates or times of the day.

Related Topics

- [An Example Policy Set Configuration](#)

Load-Aware Resource Placement

You can configure a database so that Oracle Clusterware is aware of the CPU requirements and limits for the given database.

Oracle Clusterware uses this information to place the database resource only on servers that have a sufficient number of CPUs, amount of memory, or both.

If you have configured resources with CPU or memory requirements in Oracle Clusterware, then Oracle Clusterware will only attempt to start those resources on the servers that have that meet those requirements. For database resources, specifically, you can configure the CPU or memory values into the `CPU_COUNT` and `MEMORY_TARGET` instance parameters, respectively.

Note:

Configuring the instance parameters requires the following:

- For CPU (Instance Caging), the Resource Manager must be enabled
- For memory, Automatic Memory Management must be used for the database

When you add or modify a database instance, you can configure load-aware resource placement, as follows:

```
$ srvctl modify database -db db_unique_name -cpucount cpu_count -  
memorytarget memory_target
```

In the preceding example, `cpu_count` refers to the number of workload CPUs and `memory_target` refers to the target memory, in MB, used by the resource.

If the Resource Manager is enabled in the database, then Oracle Clusterware sets the `CPU_COUNT` parameter to the value of `-cpucount`. Also, if Automatic Memory Management is enabled for the database, then Oracle Clusterware sets the `MEMORY_TARGET` database parameter to the value of `-memorytarget`.

Related Topics

- *Oracle Database Reference*
- *Oracle Database Reference*
- *Oracle Database Administrator's Guide*

Server Configuration and Server State Attributes

Oracle Clusterware assigns each server a set of attributes as soon as you add a server to a cluster.

Some of these attributes describe the physical characteristics of the server, while others describe the state conditions of the server. Also, there are other server attributes which you can modify that help further categorize servers. If you remove the server from the cluster, then Oracle Clusterware deletes the server object.

You use server configuration attributes to categorize servers, as part of a server categorization management policy.

The following table lists and describes server configuration attributes.

Table 3-4 Server Configuration Attributes

Attribute	Description
ACTIVE_CSS_ROLE	Role being performed by the server. A server can have the hub role, which is a designated role for a server in an Oracle Flex Cluster or the designated role for any node in an Oracle Clusterware standard Cluster. Note: You cannot configure this attribute.
CONFIGURED_CSS_ROLE	Configured role for the server. A server can have the hub role, which is the designated role for a server in an Oracle Flex Cluster or the designated role for any node in an Oracle Clusterware standard Cluster. Note: You cannot configure this attribute.
CPU_CLOCK_RATE	CPU clock rate in megahertz (MHz)
CPU_COUNT	Number of processors
CPU_EQUIVALENCY	The relative value (expressed as a positive integer greater than or equal to 1) that Oracle Clusterware uses to describe that the CPU power provided by a server may deviate (positively or negatively) from its physical representation using a baseline of 1000, for example. A value lower than 1000 describes that, despite a certain value for the CPU_COUNT and CPU_CLOCK_RATE parameters, the equivalent power provided by this server is respectively lower. Use the following commands to view or modify, respectively, this attribute on the local server: <pre>crsctl get cpu equivalency crsctl set cpu equivalency</pre>
CPU_HYPERTHREADING	Status of hyperthreading for the CPU. A value of 0 signifies that hyperthreading is not in use. A value of 1 signifies that hyperthreading is in use.
MEMORY_SIZE	Memory size in megabytes (MB)
NAME	The name of the server.

Table 3-4 (Cont.) Server Configuration Attributes

Attribute	Description
RESOURCE_USE_ENABLED	<p>A server pool resource management parameter. If the value for this attribute is 1, which is the default, then the server can be used for resource placement. If the value is 0, then Oracle Clusterware disallows starting server pool resources on the server. The server remains in the Free pool.</p> <p>You can review the setting and control this attribute for each cluster member node by using the <code>crsctl get resource use</code> and <code>crsctl set resource use</code> commands.</p>
SERVER_LABEL	<p>An arbitrary value that you can use to label the server. You can use this attribute when setting up server categories. For example, you can specify a location (such as <code>building_A</code> or <code>building_B</code>), which makes it possible to put servers into pools where location is a requirement, by creating an appropriate server category and using it for the server pool.</p> <p>Use the following commands to view or modify, respectively, this attribute on the local server:</p> <pre>crsctl get server label crsctl set server label</pre>

The following table lists and describes read-only server state and configuration attributes:

Table 3-5 Server State Attributes

Attribute	Description
ACTIVE_POOLS	<p>A space-delimited list of the names of the server pools to which a server belongs. Oracle Clusterware manages this list automatically.</p>

Table 3-5 (Cont.) Server State Attributes

Attribute	Description
STATE	<p data-bbox="516 342 997 363">A server can be in one of the following states:</p> <ul data-bbox="516 380 1373 1052" style="list-style-type: none"><li data-bbox="516 380 1373 432">• ONLINE: The server is a member of the cluster and is available for resource placement.<li data-bbox="516 443 1373 495">• OFFLINE: The server is not currently a member of the cluster. Subsequently, it is not available for resource placement.<li data-bbox="516 506 1373 642">• JOINING: When a server joins a cluster, Oracle Clusterware processes the server to ensure that it is valid for resource placement. Oracle Clusterware also checks the state of resources configured to run on the server. Once the validity of the server and the state of the resources are determined, the server transitions out of this state.<li data-bbox="516 653 1373 705">• LEAVING: When a planned shutdown for a server begins, the state of the server transitions to LEAVING, making it unavailable for resource placement.<li data-bbox="516 716 1373 863">• VISIBLE: Servers that have Oracle Clusterware running, but not the Cluster Ready Services daemon (CRSD), are put into the VISIBLE state. This usually indicates an intermittent issue or failure and Oracle Clusterware trying to recover (restart) the daemon. Oracle Clusterware cannot manage resources on servers while the servers are in this state.<li data-bbox="516 873 1373 1052">• RECONFIGURING: When servers move between server pools due to server pool reconfiguration, a server is placed into this state if resources that ran on it in the current server pool must be stopped and relocated. This happens because resources running on the server may not be configured to run in the server pool to which the server is moving. As soon as the resources are successfully relocated, the server is put back into the ONLINE state. <p data-bbox="516 1062 1292 1083">Use the <code>crsctl status server</code> command to obtain server information.</p>

Table 3-5 (Cont.) Server State Attributes

Attribute	Description
STATE_DETAILS	<p>This is a read-only attribute that Oracle Clusterware manages. The attribute provides additional details about the state of a server. Possible additional details about a server state are:</p> <p>Server state: ONLINE:</p> <ul style="list-style-type: none"> AUTOSTARTING RESOURCES Indicates that the resource autostart procedure (performed when a server reboots or the Oracle Clusterware stack is restarted) is in progress for the server. AUTOSTART QUEUED The server is waiting for the resource autostart to commence. Once that happens, the attribute value changes to AUTOSTARTING RESOURCES. <p>Server state: RECONFIGURING:</p> <ul style="list-style-type: none"> STOPPING RESOURCES Resources that are restricted from running in a new server pool are stopping. STARTING RESOURCES Resources that can run in a new server pool are starting. RECONFIG FAILED One or more resources did not stop and thus the server cannot transition into the ONLINE state. At this point, manual intervention is required. You must stop or unregister resources that did not stop. After that, the server automatically transitions into the ONLINE state. <p>Server state: JOINING:</p> <ul style="list-style-type: none"> CHECKING RESOURCES Whenever a server reboots, the Oracle Clusterware stack restarts, or CRSD on a server restarts, the policy engine must determine the current state of the resources on the server. While that procedure is in progress, this value is returned.

Memory Pressure Management for Database Servers

Enterprise database servers can use all available memory due to too many open sessions or runaway workloads. Running out of memory can result in failed transactions or, in extreme cases, a restart of the server and the loss of a valuable resource for your applications. Oracle Database QoS Management detects memory pressure on a server in real time and redirects new sessions to other servers to prevent using all available memory on the stressed server.

When Oracle Database QoS Management is enabled and managing an Oracle Clusterware server pool, Cluster Health Monitor sends a metrics stream that provides real-time information about memory resources for the cluster servers to Oracle Database QoS Management. This information includes the following:

- Amount of available memory
- Amount of memory currently in use

If Oracle Database QoS Management determines that a node is under memory stress, then the database services managed by Oracle Clusterware are stopped on that node, preventing new connections from being created. After the memory stress is relieved,

the services on that node are restarted automatically, and the listener starts sending new connections to that server. Memory pressure can be relieved in several ways (for example, by closing existing sessions or by user intervention).

Rerouting new sessions to different servers protects the existing workloads on the memory-stressed server and enables the server to remain available. Managing the memory pressure for servers adds a new resource protection capability in managing service levels for applications hosted on Oracle RAC databases.

Server Category Attributes

You define servers into named categories, and assign attributes that define servers as members of that category.

Some attributes that you can use to define members of a category describe the state conditions for the server, and others describe the physical characteristics of the server. You can also create your own characteristics to define servers as members of a particular category.



Note:

If you change the value of any of the server attributes listed in the `EXPRESSION` server category attribute, then you must restart the Oracle Clusterware technology stack on the affected servers before the new values take effect.

The following table lists and describes possible server category attributes.

Table 3-6 Server Category Attributes

Attribute	Description
NAME	The name of the server category, which you must specify when you create the server category. Server category names must be unique within the domain of names of user-created entities, such as resources, types, and servers. A server pool name has a 254 character limit and can contain any platform-supported characters except the exclamation point (!) and the tilde (~). A server pool name cannot begin with a period nor with <i>ora</i> .
ACTIVE_CSS_ROLE	Active role for the server, which is <code>hub</code> for a server that is a Hub Node in either an Oracle Flex Cluster or an Oracle Clusterware standard Cluster. This is the default value in either case.

Table 3-6 (Cont.) Server Category Attributes

Attribute	Description
EXPRESSION	<p>A set of server attribute names, values, and conditions that can be evaluated for each server to determine whether it belongs to the category. Table 3-4 lists and describes server attributes.</p> <p>Acceptable comparison operators include:</p> <ul style="list-style-type: none"> =: equal eqi: equal, case insensitive >: greater than <: less than !=: not equal co: contains coi: contains, case insensitive st: starts with en: ends with nc: does not contain nci: does not contain, case insensitive <p>Acceptable boolean operators include:</p> <ul style="list-style-type: none"> AND OR <p>Note: Spaces must surround the operators used in the EXPRESSION string.</p> <p>For example:</p> <pre>EXPRESSION='((NAME = server1) OR (NAME = server2))'</pre>

An Example Policy Set Configuration

In this example, you have a four-node cluster that is used by three different applications, *app1*, *app2*, and *app3*, and that you have created three server pools, *pool1*, *pool2*, and *pool3*. You configure the server pools such that each application is assigned to run in its own server pool, and that *app1* wants to have two servers, and *app2* and *app3* each want one server.

The server pool configurations are as follows:

```
$ crsctl status serverpool pool1 -p
NAME=pool1
IMPORTANCE=0
MIN_SIZE=2
MAX_SIZE=2
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:mjk:rxw,pgrp:g900:rxw,other::r--
SERVER_CATEGORY=

$ crsctl status serverpool pool2 -p
NAME=pool2
IMPORTANCE=0
```

```
MIN_SIZE=1
MAX_SIZE=1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:mjk:rw,grp:g900:rw,other::r--
SERVER_CATEGORY=
```

```
$ crsctl status serverpool pool3 -p
NAME=pool3
IMPORTANCE=0
MIN_SIZE=1
MAX_SIZE=1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:mjk:rw,grp:g900:rw,other::r--
SERVER_CATEGORY=
```

**Note:**

The `crsctl status serverpool` command shown in the preceding examples only functions if you created the server pools using CRSQL.

This configuration, however, does not consider the fact that some applications need server time at different times of the day, week, or month. Email applications, for example, typically use more resources during business hours and use less resources at night and on weekends.

Further assume that *app1* requires two servers during business hours, but only requires one server at night and does not require any servers on weekends. At the same time, *app2* and *app3* each require one server during business hours, while at night, *app2* requires two servers and *app3* requires one. On the weekend, *app2* requires one server and *app3* requires three. This scenario suggests three configurations that you must configure for the cluster:

1. Day Time:

app1 uses two servers
app2 and *app3* use one server, each

2. Night Time:

app1 uses one server
app2 uses two servers
app3 uses one server

3. Weekend:

app1 is not running (0 servers)
app2 uses one server
app3 uses three servers

Policy Set Creation

Given these assumptions, run the `crsctl create policyset` command to create a policy set with a single policy named `Default`, which reflects the configuration displayed by the `crsctl status serverpool` command. You can use the `Default` policy to create other policies to meet the needs assumed in this example. The `crsctl create policyset` command creates a text file similar to [Example 3-1](#).

Example 3-1 Policy Set Text File

```
SERVER_POOL_NAMES=Free pool1 pool2 pool3
POLICY
  NAME=Default
  SERVERPOOL
    NAME=pool1
    IMPORTANCE=0
    MAX_SIZE=2
    MIN_SIZE=2
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool2
    IMPORTANCE=0
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool3
    IMPORTANCE=0
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=
```

Policy Modification

To modify the preceding policy set to meet the needs assumed in this example, edit the text file to define policies for the three scenarios discussed previously, by changing the name of the policy from `Default` to `DayTime`. Then, copy the policy and paste it twice to form two subsequent policies, which you name `NightTime` and `Weekend`, as shown in [Example 3-2](#).

Example 3-2 Modified Policy Set Text File

```
SERVER_POOL_NAMES=Free pool1 pool2 pool3
POLICY
  NAME=DayTime
  SERVERPOOL
    NAME=pool1
    IMPORTANCE=0
    MAX_SIZE=2
    MIN_SIZE=2
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool2
    IMPORTANCE=0
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=
```

```

SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
POLICY
  NAME=NightTime
  SERVERPOOL
    NAME=pool1
    IMPORTANCE=0
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool2
    IMPORTANCE=0
    MAX_SIZE=2
    MIN_SIZE=2
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool3
    IMPORTANCE=0
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=
POLICY
  NAME=Weekend
  SERVERPOOL
    NAME=pool1
    IMPORTANCE=0
    MAX_SIZE=0
    MIN_SIZE=0
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool2
    IMPORTANCE=0
    MAX_SIZE=1
    MIN_SIZE=1
    SERVER_CATEGORY=
  SERVERPOOL
    NAME=pool3
    IMPORTANCE=0
    MAX_SIZE=3
    MIN_SIZE=3
    SERVER_CATEGORY=

```

Notice that, in addition to changing the names of the individual policies, the `MAX_SIZE` and `MIN_SIZE` policy attributes for each of the server pools in each of the policies were also modified according to the needs of the applications.

The following command registers the policy set stored in a file with Oracle Clusterware:

```
$ crsctl modify policyset -file file_name
```

You can achieve the same results as shown in the previous examples by editing the Default policy set, as a whole, using the `crsctl modify policyset` command, and by using the `crsctl modify serverpool` command to change individual server pool attributes for a specific policy.

The following command modifies the Default policy set to manage the three server pools:

```
$ crsctl modify policyset -attr "SERVER_POOL_NAMES=Free pool1 pool2 pool3"
```

The following commands add the three policies:

```
$ crsctl add policy DayTime  
$ crsctl add policy NightTime  
$ crsctl add policy Weekend
```

The following commands configure the three server pools according to the requirements of the policies:

```
$ crsctl modify serverpool pool1 -attr "MIN_SIZE=2,MAX_SIZE=2" -policy  
DayTime  
$ crsctl modify serverpool pool1 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy  
NightTime  
$ crsctl modify serverpool pool1 -attr "MIN_SIZE=0,MAX_SIZE=0" -policy  
Weekend  
  
$ crsctl modify serverpool pool2 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy  
DayTime  
$ crsctl modify serverpool pool2 -attr "MIN_SIZE=2,MAX_SIZE=2" -policy  
NightTime  
$ crsctl modify serverpool pool2 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy  
Weekend  
  
$ crsctl modify serverpool pool3 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy  
DayTime  
$ crsctl modify serverpool pool3 -attr "MIN_SIZE=1,MAX_SIZE=1" -policy  
NightTime  
$ crsctl modify serverpool pool3 -attr "MIN_SIZE=3,MAX_SIZE=3" -policy  
Weekend
```

There are now three distinct policies to manage the server pools to accommodate the requirements of the three applications.

Policy Activation

The policy set is now configured and controlling the three server pools with three different policies. You can activate policies when necessary, prompting Oracle Clusterware to reconfigure a server pool according to each policy's configuration.

The following command activates the `DayTime` policy:

```
$ crsctl modify policyset -attr "LAST_ACTIVATED_POLICY=DayTime"
```

The current status of the resources is as follows:

```
$ crsctl status resource -t
-----
-----
Name          Target  State      Server          State details
-----
-----
Cluster Resources
-----
-----
appl
  1          ONLINE  ONLINE     mjk_has3_2     STABLE
  2          ONLINE  ONLINE     mjk_has3_0     STABLE
app2
  1          ONLINE  ONLINE     mjk_has3_1     STABLE
app3
  1          ONLINE  ONLINE     mjk_has3_3     STABLE
```

The status of the server pools is as follows:

```
$ crsctl stat serverpool
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=

NAME=pool1
ACTIVE_SERVERS=mjk_has3_0 mjk_has3_2

NAME=pool2
ACTIVE_SERVERS=mjk_has3_1

NAME=pool3
ACTIVE_SERVERS=mjk_has3_3
```

The servers are allocated according to the `DayTime` policy and the applications run on their respective servers.

The following command activates the `Weekend` policy (remember, because the server pools have different sizes, as servers move between server pools, some applications will be stopped and others will be started):

```
$ crsctl modify policyset -attr "LAST_ACTIVATED_POLICY=Weekend"
CRS-2673: Attempting to stop 'appl' on 'mjk_has3_2'
CRS-2673: Attempting to stop 'appl' on 'mjk_has3_0'
CRS-2677: Stop of 'appl' on 'mjk_has3_0' succeeded
CRS-2672: Attempting to start 'app3' on 'mjk_has3_0'
```

```
CRS-2677: Stop of 'app1' on 'mjk_has3_2' succeeded
CRS-2672: Attempting to start 'app3' on 'mjk_has3_2'
CRS-2676: Start of 'app3' on 'mjk_has3_2' succeeded
CRS-2676: Start of 'app3' on 'mjk_has3_0' succeeded
```

The current status of the resources is as follows:

```
$ crsctl status resource -t
-----
-----
Name          Target State      Server          State
details
-----
-----
Cluster Resources
-----
-----
app1
  1          ONLINE OFFLINE
  2          ONLINE OFFLINE
app2
  1          ONLINE ONLINE      mjk_has3_1     STABLE
app3
  1          ONLINE ONLINE      mjk_has3_0     STABLE
  2          ONLINE ONLINE      mjk_has3_2     STABLE
  3          ONLINE ONLINE      mjk_has3_3     STABLE
-----
-----
```

The status of the server pools is as follows:

```
$ crsctl status serverpool
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=

NAME=pool1
ACTIVE_SERVERS=

NAME=pool2
ACTIVE_SERVERS=mjk_has3_1

NAME=pool3
ACTIVE_SERVERS=mjk_has3_0 mjk_has3_2 mjk_has3_3
```

Using the `crsctl modify policyset` command, Oracle Clusterware changed server pool configuration, moved servers according to the requirements of the policy, and stopped and started the applications.

Related Topics

- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)

4

Oracle Flex Clusters

An Oracle Flex Cluster scales Oracle Clusterware to large numbers of nodes.

This chapter includes the following topics:

- [Overview of Oracle Flex Clusters](#)
- [Managing Oracle Flex Clusters](#)
- [Oracle Extended Clusters](#)

Overview of Oracle Flex Clusters

Oracle Grid Infrastructure installed in an Oracle Flex Cluster configuration is a scalable, dynamic, robust network of nodes.

Oracle Flex Clusters provide a platform for a variety of applications, including Oracle Real Application Clusters (Oracle RAC) databases with large numbers of nodes. Oracle Flex Clusters also provide a platform for other service deployments that require coordination and automation for high availability. Starting at Oracle Clusterware 12c release 2 (12.2), all clusters are configured as Oracle Flex Clusters.

All nodes in an Oracle Flex Cluster belong to a single Oracle Grid Infrastructure cluster. This architecture centralizes policy decisions for deployment of resources based on application needs, to account for various service levels, loads, failure responses, and recovery.

Oracle Flex Clusters contain [Hub Nodes](#) and any number of other supported nodes. The number of Hub Nodes in an Oracle Flex Cluster can be as many as 64. The number of other nodes can be many more. Hub Nodes can host different types of applications.

Hub Nodes are similar to Oracle Grid Infrastructure nodes in an Oracle Clusterware standard Cluster configuration: they are tightly connected, and have direct access to shared storage. Use Hub Nodes to host read-write database instances.

Other supported nodes in an Oracle Flex Cluster are different from standard Oracle Grid Infrastructure nodes, in that they do not require direct access to shared storage, but instead request data through Hub Nodes. Use other nodes to host read-only database instances.

 **Note:**

Read-write and read-only database instances of the same primary database can coexist in an Oracle Flex Cluster.

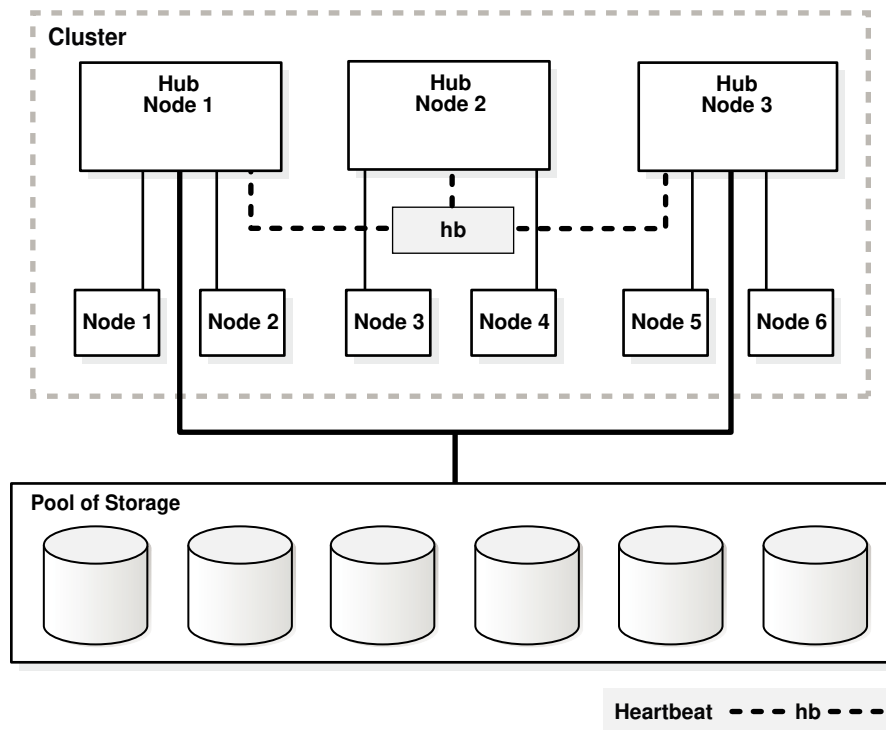
Hub Nodes can run in an Oracle Flex Cluster configuration without having any other nodes as cluster member nodes, but other nodes must be members of a cluster that includes at least one Hub Node.

 **Note:**

If you upgrade an Oracle Flex Cluster, then Oracle recommends that you upgrade the Hub Nodes first, and that you also have any upgraded Hub Nodes up and running as part of the upgrade process.

Oracle Flex Cluster Configuration

Figure 4-1 Oracle Clusterware Configuration



- Hub Nodes are tightly connected, and have direct access to shared storage. They would traditionally be deployed as hosts for Oracle RAC or Oracle RAC One database instances. Other nodes in the cluster differ from Hub Nodes in that they do not require direct access to shared storage, but instead access data through the Hub Nodes.
- All nodes in an Oracle Flex Cluster belong to a single Oracle Grid Infrastructure cluster. This architecture centralizes policy decisions for deployment of resources based on application needs, to account for various service levels, loads, failure responses, and recovery.
- Oracle Flex Clusters consists of Hub Nodes that support other nodes in the cluster. The number of Hub Nodes in an Oracle Flex Cluster must be at least one and can be as many as 64, while the number of other, supported nodes can be many more. Hub Nodes can host different types of applications.
- Oracle Flex Clusters may operate with one or many Hub Nodes, but other nodes are optional and can only exist as members of a cluster that includes at least one Hub Node.

Reader Nodes

You can use other nodes to host Oracle RAC database instances that run in read-only mode, which become [reader nodes](#). You can optimize these nodes for parallel query operations by provisioning nodes with a large amount of memory so that data is cached in the hosting node.

A hosting node sends periodic heartbeat messages to its associated Hub Node, which is different from the heartbeat messages that occur between Hub Nodes. During planned shutdown of the Hub Nodes, a hosting node attempts to connect to another Hub Node, unless the hosting node is connected to only one Hub Node. If the Hub Node is evicted, then the hosting node is also evicted from the cluster.

Managing Oracle Flex Clusters

Use CRSQL to manage Oracle Flex Clusters after successful installation of Oracle Grid Infrastructure for either a small or large cluster.

This section includes the following topics:

- [Changing the Cluster Mode](#)

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- [Oracle Clusterware Control \(CRSQL\) Utility Reference](#)

Changing the Cluster Mode

You can change the mode of an existing Oracle Clusterware standard Cluster to be an Oracle Flex Cluster.

 **Note:**

- Changing the cluster mode requires cluster downtime.
- Oracle does not support changing an Oracle Flex Cluster to an Oracle Clusterware standard Cluster.
- Oracle Flex Cluster requires Grid Naming Service (GNS).
- Zone delegation is not required.

Changing an Oracle Clusterware Standard Cluster to an Oracle Flex Cluster

Use CRSQL to change an existing Oracle Clusterware standard Cluster to an Oracle Flex Cluster.

Perform the following steps:

1. Run the following command to determine the current mode of the cluster:

```
$ crsctl get cluster mode status
```

2. Run the following command to ensure that the Grid Naming Service (GNS) is configured with a fixed VIP:

```
$ srvctl config gns
```

This procedure cannot succeed unless GNS is configured with a fixed VIP. If there is no GNS, then, as `root`, create one, as follows:

```
# srvctl add gns -vip vip_name | ip_address
```

Run the following command as `root` to start GNS:

```
# srvctl start gns
```

3. Use the Oracle Automatic Storage Management Configuration Assistant (ASMCA) to enable Oracle Flex ASM in the cluster before you change the cluster mode.
4. Run the following command as `root` to change the mode of the cluster to be an Oracle Flex Cluster:

```
# crsctl set cluster mode flex
```

5. Stop Oracle Clusterware by running the following command as `root` on each node in the cluster:

```
# crsctl stop crs
```

6. Start Oracle Clusterware by running the following command as `root` on each node in the cluster:

```
# crsctl start crs -wait
```

 **Note:**

Use the `-wait` option to display progress and status messages.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

Oracle Extended Clusters

You can extend an Oracle RAC cluster across two, or more, geographically separate sites, each equipped with its own storage. In the event that one of the sites fails, the other site acts as an active standby.

Both Oracle ASM and the Oracle Database stack, in general, are designed to use enterprise-class shared storage in a data center. Fibre Channel technology, however, enables you to distribute compute and storage resources across two or more data centers, and connect them through ethernet cables and Fibre Channel, for compute and storage needs, respectively.

While you can configure Oracle Extended Clusters when you install Oracle Grid Infrastructure, you can also do so post installation using the `ConvertToExtended` script. You manage your Oracle Extended Cluster using `CRSCTL`.

Configuring Oracle Extended Clusters

This procedure is only supported for clusters that have been installed with or upgraded to Oracle Grid Infrastructure 12c release 2 (12.2), or later, which are typically configured with one site (default site).

Note:

This procedure requires that all nodes in the cluster be accessible. There will also be a cluster outage during which time database access is disrupted.

You can configure an Oracle Extended Cluster with one or many disk groups and with multiple failure groups. Using the `ConvertToExtended` script you can create multiple data sites and associate a node with each data site. All Oracle Flex ASM storage remains associated with the default cluster site because there is no mechanism to convert an existing disk group to an extended disk group. After you convert your cluster to an Oracle Extended Cluster, the voting file membership remains flat, and not hierarchical.

You must also add an extended disk group, and migrate the voting files to the extended disk group to take advantage of a site-specific hierarchical voting file algorithm.

Use `CRSCTL` to query the cluster, as follows, to determine its extended status:

```
$ crsctl get cluster extended
CRS-6579: The cluster is 'NOT EXTENDED'
```

```
$ crsctl query cluster site -all
Site 'crsclus' identified by '7b7b3bef4c1f5ff9ff8765bceb45433a' in state
'ENABLED',
and contains nodes 'node1,node2,node3,node4', and disks ''.
```

The preceding example identifies a cluster called `crsclus` that has four nodes—`node1`, `node2`, `node3`, and `node4`—and a disk group—`datadg`. The cluster has one site configured.

Shut down the Oracle Clusterware stack to prevent some Oracle Flex ASM instances, `cssd`, and `crsd` from considering the cluster mode is not extended, and other components to consider that it is extended. The advantage to keeping the Grid Plug

and Play daemon (gpnpd) online is that the profile gets updated on the remote nodes. When you next start the Oracle Clusterware stack, the cluster mode will be extended.

```
# crsctl stop cluster -all
```

1. On the first node in the cluster, run the following command:

```
$ rootcrs.sh -converttoextended -firstnode -sites site_list -site  
site_name
```

2. Run `crsctl check css` to ensure that Cluster Synchronization Services (CSS) is not running on any remote nodes.

3. Generate the site names and the site GUIDs, using the following command:

```
$ crsctl query cluster site -all
```

The preceding command displays output similar to the following:

```
Site 'SiteA' identified by GUID '7b7b3bef4c1f5ff9ff8765bceb45433a' in  
state  
  'ONLINE' contains nodes 'node1,node2,node3', and disks 'disk1, disk2,  
disk3'.  
Site 'SiteB' identified by GUID '23453bef4c1f5ff9ff8765bceb45433a' in  
state  
  'QUARANTINED' contains nodes 'node4, node5, node6', and disks 'disk4,  
disk5, disk6'.  
Site 'SiteQ' identified by GUID '98763bef4c1f5ff9ff8765bceb45433a' in  
state  
  'ONLINE' contains no nodes and disk 'disk7'.  
Site 'SiteD' identified by GUID '4abcd453c6bc6fc1ffd3a58849d5ba41' in  
state  
  'ONLINE' contains nodes 'cuj1234' and no disk
```

Update the checkpoint with this information, and copy it to the rest of the nodes in the cluster.

4. Update the Grid Plug and Play profile and set `ext_mode` to `TRUE` using the following commands:

```
$ cd $Grid_home/gpnp/host_name/profiles/peer  
  
$ gpnptool getpval -p=profile.xml -prf_sq -o=temp_file //  
run_gpnptool_getpval()  
$ gpnptool edit -p=profile.xml -asm_ext="" prf_sq=seq+1 -  
o=profile_ext.xml
```

 **Note:**

The value of `prf_sq` must be greater than what is in the current profile for the `ProfileSequence` key.

```
$ gpnptool sign -p=profile_ext.xml -o=profile_ext_sign.xml
$ mv profile_ext_sign.xml profile.xml
$ gpnptool put -p=profile.xml
```

5. Add sites to the local configuration, as follows:

```
$ crsctl add crs site site_name -guid site_guid -local
```

6. Add sites to the global configuration, as follows (this requires that OCR be running):

```
$ crsctl add crs site site_name -guid site_guid
```

7. Update the to-site mapping in the local configuration for this node, as follows:

```
$ crsctl modify cluster site site_name -n local_host -local
```

8. Update the node-to-site mapping, as follows:

```
$ crsctl modify cluster site site_name -n host1, host2, ...
```

9. Stop and then start the Oracle High Availability Services stack, as follows:

```
# crsctl stop crs
# crsctl start crs
```

Do the following on the rest of the nodes in the cluster:

1. Extend the cluster to the specific node, as follows:

```
$ rootcrs.sh -converttoextended -site site_name
```

2. Ensure that CSS is not running on any remote nodes.
3. Look up new sites and the site GUIDs using the previous checkpoint information.
4. Add sites to the local configuration, as follows:

```
$ crsctl add crs site site_name -guid site_guid -local
```

5. Update the to-site mapping in the local configuration for this node, as follows:

```
$ crsctl modify cluster site site_name -n local_host -local
```

6. Stop and then start the Oracle High Availability Services stack, as follows:

```
# crsctl stop crs
# crsctl start crs
```

After you finish configuring the Oracle Extended Cluster, run the following command to verify the configuration:

```
$ crsctl get cluster extended
CRS-XXXX: The cluster is 'EXTENDED'

$ crsctl query cluster site -all
Site 'crsclus' identified by '7b7b3bef4c1f5ff9ff8765bceb45433a' is
'ONLINE', and
contains nodes '', and disks ''.
Site 'ny' identified by '888b3bef4c1f5ff9ff8765bceb45433a' is 'ONLINE',
and \
contains nodes 'node1,node2', and disks ''.
Site 'nj' identified by '999b3bef4c1f5ff9ff8765bceb45433a' is 'ONLINE',
and \
contains nodes 'node3,node4', and disks ''.
```

The output in the preceding command examples is similar to what CRSCTL displays when you run the commands.

5

Oracle Fleet Patching and Provisioning

Oracle Fleet Patching and Provisioning is a software lifecycle management method for provisioning and maintaining Oracle homes.

Oracle Fleet Patching and Provisioning (Oracle FPP) enables mass deployment and maintenance of standard operating environments for databases, clusters, and user-defined software types. With Oracle Fleet Patching and Provisioning, you can also install clusters and provision, patch, scale, and upgrade Oracle Grid Infrastructure and Oracle Database 11g release 2 (11.2), and later. Additionally, you can provision applications and middleware.

Note:

Combined Oracle FPP patching for Oracle Grid Infrastructure and Oracle Database is not supported for standalone (Oracle Restart) configurations, only for cluster configurations.

Oracle Fleet Patching and Provisioning is a service in Oracle Grid Infrastructure that you can use in either of the following modes:

- As a central server ([Fleet Patching and Provisioning Server](#)), that stores and manages standardized [images](#), called [gold images](#). You can deploy gold images to any number of nodes across a data center. You can use the deployed homes to create new clusters and databases, and patch, upgrade, and scale existing installations.

The server manages software homes on the cluster hosting the Oracle Fleet Patching and Provisioning Server, itself, Oracle Fleet Patching and Provisioning Clients, and can also manage installations running Oracle Grid Infrastructure 11g release 2 (11.2.0.3 and 11.2.0.4) and 12c release 1 (12.1.0.2). The server can also manage installations running no grid infrastructure.

An Oracle Fleet Patching and Provisioning Server can provision new installations and can manage existing installations without any changes to the existing installations (such as no agent, daemon, or configuration prerequisites). Oracle Fleet Patching and Provisioning Servers also include capabilities for automatically sharing gold images among peer Oracle Fleet Patching and Provisioning Servers to support enterprises with geographically distributed data centers.

- As a client (Oracle Fleet Patching and Provisioning Client), that can be managed from the central Oracle Fleet Patching and Provisioning Server or directly by running commands on the Oracle Fleet Patching and Provisioning Client, itself. As with the Oracle Fleet Patching and Provisioning Server, the Oracle Fleet Patching and Provisioning Client is a service built in to Oracle Grid Infrastructure and is available with Oracle Grid Infrastructure 12c release 2 (12.2.0.1), and later. The Oracle Fleet Patching and Provisioning Client service can retrieve gold images from the Oracle Fleet Patching and Provisioning Server, upload new images based on policy, and apply maintenance operations to itself.

For patching operations, a third option is available with Oracle Database and Oracle Grid Infrastructure 18c, and later. The procedures for updating database and grid infrastructure homes have been modularized into independent *automatons* that are included with Oracle Database and Oracle Grid Infrastructure, and can be run locally without any central Oracle Fleet Patching and Provisioning Server in the architecture. This provides an immediate entry point to the capabilities of Oracle Fleet Patching and Provisioning as soon as you bring up an Oracle Database or cluster.

Managing Oracle software using Oracle Fleet Patching and Provisioning:

- Ensures standardization and enables high degrees of automation with gold images and managed lineage of deployed software.
- Minimizes the maintenance window by deploying new homes (gold images) out-of-place, without disrupting active databases or clusters.
- Simplifies maintenance by providing automatons that are invoked with an API across database versions and deployment models.
- Reduces maintenance risk with built-in validations and a dry-run mode to ensure operations will succeed end-to-end.
- In the event of an issue, commands are resumable and restartable, further reducing risk of maintenance operations.
- Minimizes and, in many cases, eliminates the impact of patching and upgrades, with features that include:
 - Zero-downtime database upgrade: A fully-automated upgrade executed entirely within the deployment with no extra nodes or external storage required.
 - Adaptive management of database sessions and OJVM during rolling patching.
 - Options for fine-grained management of consolidated deployments.

The deployment and maintenance operations are extensible, allowing customizations to include environment-specific actions into the automated workflow.

Oracle Fleet Patching and Provisioning Automatons

- *Zero-downtime database upgrade* automates all of the steps involved in a database upgrade to minimize or even eliminate application downtime while upgrading an Oracle database. It also minimizes resource requirements and provides a fallback path in case the upgrade must be rolled back.
- *Adaptive Oracle RAC Rolling Patching for OJVM Deployments*: In a clustered environment, the default approach for Oracle Fleet Patching and Provisioning for patching a database is Oracle RAC rolling patching. However non-rolling may be required if the patched database home contains OJVM patches. In this case, Oracle Fleet Patching and Provisioning determines whether rolling patching is possible and does so, if applicable.
- *Dry-run command evaluation*: Before running any command, Oracle Fleet Patching and Provisioning checks various preconditions to ensure the command will succeed. However, some conditions cannot be detected prior to a command running. And, while Oracle Fleet Patching and Provisioning allows a failed command to be reverted or resumed after an error condition is corrected, it is preferable to address as many potential issues as possible before the command is run. The command evaluation mode will test the preconditions for a given

command, without making any changes, and report potential problems and correct them before the command is actually run.

- *Independent automatons*: Prior to Oracle Database 18c, performing any Oracle Fleet Patching and Provisioning operation (for example, switching a database home to a later version) required the presence of a central Oracle Fleet Patching and Provisioning Server. Beginning with Oracle Database 18c, key functionality can be performed independently, with no central Oracle Fleet Patching and Provisioning Server in the architecture.

Note:

When you install Oracle Grid Infrastructure, the Oracle Fleet Patching and Provisioning Server is configured, by default, in the *local* mode to support the *local switch home* capability. If you must configure the general Oracle Fleet Patching and Provisioning Server product, then you must remove the current local-mode Oracle Fleet Patching and Provisioning Server, using the following commands, as `root`:

```
# srvctl stop rhpserver
```

Ignore a message similar to "Fleet Patching and Provisioning Server is not running".

```
# srvctl remove rhpserver
```

Global Fleet Standardization and Management

- *Sharing gold images between peer Oracle Fleet Patching and Provisioning Servers*: Large enterprises typically host multiple data centers and, within each data center, there may be separate network segments. In the Oracle Fleet Patching and Provisioning architecture, one Oracle Fleet Patching and Provisioning Server operates on a set of targets within a given data center (or network segment of a data center). Therefore each data center requires at least one Oracle Fleet Patching and Provisioning Server.

While each data center may have some unique requirements in terms of the gold images that target servers will use, the goal of standardization is using the same gold image across all data centers whenever possible. To that end, Oracle Fleet Patching and Provisioning supports peer-to-peer sharing of gold images to easily propagate gold images among multiple Oracle Fleet Patching and Provisioning Servers.

- *Gold image drift detection and aggregation*: After you provision a software home from a gold image, you may have to apply a patch directly to the deployed home. At this point the deployed home has drifted from the gold image. Oracle Fleet Patching and Provisioning provides two capabilities for monitoring and reporting drift:
 - Oracle Fleet Patching and Provisioning compares a specific home to its parent gold image and lists any patches that are applied to the home but that are not in the gold image.

- Oracle Fleet Patching and Provisioning compares a specific gold image to all of its descendant homes and lists the aggregation of all patches applied to those homes that are not in the gold image. This provides a build specification for a new gold image that could be applied to all of the descendants of the original gold image, such that no patches will be lost from any of those deployments.
- *Configuration collection and reporting*: The Oracle Fleet Patching and Provisioning Server can collect and retain operating system configuration and the root file system contents of specified Oracle Fleet Patching and Provisioning Clients. If an Oracle Fleet Patching and Provisioning Client node is rendered unusable (for example, a user accidentally deletes or changes operating system configuration or the root file system), then it can be difficult to determine the problem and correct it. This feature automates the collection of relevant information, enabling simple restoration in the event of node failure.

Flexibility and Extensibility

- *RESTful API*: Oracle Fleet Patching and Provisioning provides a RESTful API for many common operations, including provisioning, patching, upgrading, and query operations.

See Also:

Oracle Database REST API Reference

- *Customizable authentication*: Host-to-host authentication in certain environments, particularly in compliance-conscious industries, such as financials and e-commerce, often uses technologies and products that are not supported, natively, by Oracle Fleet Patching and Provisioning. This feature allows integrating Oracle Fleet Patching and Provisioning authentication with the mechanisms in use at your data center.
- *Command scheduler*: The ability to schedule and bundle automated tasks is essential for maintenance of a large database estate. Oracle Fleet Patching and Provisioning supports scheduling tasks such as provisioning software homes, switching to a new home, and scaling a cluster. Also, you can add a list of clients to a command, facilitating large-scale operations.
- *Configurable connectivity*: As security concerns and compliance requirements increase, so do the restrictions on connectivity across the intranets of many enterprises. You can configure the small set ports used for communication between the Oracle Fleet Patching and Provisioning Server and its Clients, allowing low-impact integration into firewalled or audit-conscious environments.

Other Oracle Fleet Patching and Provisioning Features

- *Zero-downtime upgrade*: Automation of all of upgrade steps involved minimizes or even eliminates application downtime while upgrading an Oracle Database. It also minimizes resource requirements and provides a fallback path in case you must roll back the upgrade. You can run a zero-downtime upgrade on certain versions of Oracle RAC and Oracle RAC One Node databases.
- *Provision new server pools*: The Oracle Fleet Patching and Provisioning Server can install and configure Oracle Grid Infrastructure (11g release 2 (11.2.0.4), and 12c release 1 (12.1.0.2) and release 2 (12.2)) on nodes that have no Oracle

software inventory and can then manage those deployments with the full complement of Oracle Fleet Patching and Provisioning functionality.

- *Provision and manage any software home:* Oracle Fleet Patching and Provisioning enables you to create a gold image from any software home. You can then provision that software to any Oracle Fleet Patching and Provisioning Client or target as a working copy of a gold image. The software may be any binary that you will run on an Oracle Fleet Patching and Provisioning Client or target.
- *Provision, scale, patch, and upgrade Oracle Grid Infrastructure:* The Oracle Fleet Patching and Provisioning Server can provision Oracle Grid Infrastructure 11g release 2 (11.2.0.4) homes, and later, add or delete nodes from an Oracle Grid Infrastructure configuration, and can also be used to patch and upgrade Oracle Grid Infrastructure homes. In addition, there is a rollback capability that facilitates undoing a failed patch procedure. While patching Oracle Grid Infrastructure, you can use Oracle Fleet Patching and Provisioning to optionally patch any database homes hosted on the cluster.
- *Provision, scale, patch, and upgrade Oracle Database:* You can use Oracle Fleet Patching and Provisioning, you can provision, scale, and patch Oracle Database 11g release 2 (11.2.0.4), and later. You can also upgrade Oracle Databases from 11g release 2 (11.2.0.3) to 11g release 2 (11.2.0.4); from 11g release 2 (11.2.0.4) to 12c release 1 (12.1.0.2); and from 11g release 2 (11.2.0.4) or 12c release 1 (12.1.0.2) to 12c release 2 (12.2).

When you provision such software, Oracle Fleet Patching and Provisioning offers additional features for creating various types of databases (such as Oracle RAC, single instance, and Oracle Real Application Clusters One Node (Oracle RAC One Node) databases) on different types of storage, and other options, such as using templates and creating container databases (CDBs). The Oracle Fleet Patching and Provisioning Server can add nodes to an Oracle RAC configuration, and remove nodes from an Oracle RAC configuration. Oracle Fleet Patching and Provisioning also improves and makes more efficient patching of database software, allowing for rapid and remote patching of the software, in most cases, without any downtime for the database.

- *Support for single-instance databases:* You can use Oracle Fleet Patching and Provisioning to provision, patch, and upgrade single-instance databases running on clusters or Oracle Restart, or on single, standalone nodes.
- *Advanced patching capabilities:* When patching an Oracle Grid Infrastructure or Oracle Database home, Oracle Fleet Patching and Provisioning offers a batch mode that speeds the patching process by patching some or all nodes of a cluster in parallel, rather than sequentially.

For Oracle Database homes, you can define disjoint sets of nodes. Each set of nodes is updated sequentially. By defining sets with reference to the database instances running on them, you can minimize the impact of rolling updates by ensuring that services are never taken completely offline. A “smartmove” option is available to help define the sets of batches to meet this goal.

Integration with Application Continuity is another enhancement to help eliminate the impact of maintenance. This provides the ability to gracefully drain and relocate services within a cluster, completely masking the maintenance from users.

- *Notifications:* The Oracle Fleet Patching and Provisioning Server is the central repository for the software homes available to the data center. Therefore, it is essential that administrators throughout the data center be aware of changes to the inventory which might impact their areas of responsibility.

Oracle Fleet Patching and Provisioning enables you and other users to subscribe to image series events. Anyone subscribed will be notified by email of any changes to the images available in a particular image series. Also, users can be notified by email when a working copy of a gold image is added to or deleted from a client.

- *Custom workflow support:* You can create actions for various Oracle Fleet Patching and Provisioning operations, such as importing images, adding or deleting working copies of the gold images, and managing a software home. You can define different actions for each operation, and further differentiate by the type of image to which the operation applies. Actions that you define can be executed before or after the given operation, and are executed on the deployment the operation applies to, whether it is the Oracle Fleet Patching and Provisioning Server, a target that is not running an Oracle Fleet Patching and Provisioning Client, or a target that is running an Oracle Fleet Patching and Provisioning Client.
- *Resume failed operations:* If an operation, such as adding an image, provisioning a working copy of a gold image, or performing a scale, patch or upgrade fails, then Oracle Fleet Patching and Provisioning reports the error and stops. After the problem is corrected (for example, a directory permissions or ownership misconfiguration on a target node), you can rerun the RHPCTL command that failed, and it will resume from the point of failure. This avoids redoing any work that may have been completed prior to the failure.
- *Audit command:* The Oracle Fleet Patching and Provisioning Server records the execution of all Oracle Fleet Patching and Provisioning operations and also records their outcome (whether success or failure). An audit mechanism enables you to query the audit log in a variety of dimensions, and also to manage its contents and size.

 **Note:**

- Oracle does not support Oracle Fleet Patching and Provisioning on HP-UX or Windows operating systems.
- The Oracle Fleet Patching and Provisioning Server does not manage operating system images.

This section includes the following topics:

- [Fleet Patching and Provisioning Architecture](#)
- [Fleet Patching and Provisioning Implementation](#)
- [Oracle Grid Infrastructure Management](#)
- [Oracle Database Software Management](#)
- [Zero-Downtime Upgrade](#)
- [Persistent Home Path During Patching](#)
- [Managing Fleet Patching and Provisioning Clients](#)

Related Topics

- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*

Fleet Patching and Provisioning Architecture

Conceptual information regarding Fleet Patching and Provisioning architecture.

The Fleet Patching and Provisioning architecture consists of a Fleet Patching and Provisioning Server and any number of Fleet Patching and Provisioning Clients and targets. Oracle recommends deploying the Fleet Patching and Provisioning Server in a multi-node cluster so that it is highly available.

The Fleet Patching and Provisioning Server cluster is a repository for all data, of which there are primarily two types:

- Gold images
- Metadata related to users, roles, permissions, and identities

The Fleet Patching and Provisioning Server acts as a central server for provisioning Oracle Database homes, Oracle Grid Infrastructure homes, and other application software homes, making them available to the cluster hosting the Fleet Patching and Provisioning Server and to the Fleet Patching and Provisioning Client clusters and targets.

Users operate on the Fleet Patching and Provisioning Server or Fleet Patching and Provisioning Client to request deployment of Oracle homes or to query gold images. When a user makes a request for an Oracle home, specifying a gold image, the Fleet Patching and Provisioning Client communicates with the Fleet Patching and Provisioning Server to pass on the request. The Fleet Patching and Provisioning Server processes the request by taking appropriate action to instantiate a copy of the gold image, and to make it available to the Fleet Patching and Provisioning Client cluster using available technologies such as Oracle Automatic Storage Management Cluster File System (Oracle ACFS) and local file systems.

The Fleet Patching and Provisioning Server communicates with Fleet Patching and Provisioning Clients (running Oracle Grid Infrastructure 12c release 2 (12.2.0.1) or later), and targets (a target is an installation with Oracle Grid Infrastructure 12c release 1 (12.1.0.2) or release 11.2, or with no Oracle Grid Infrastructure installed) using the following ports, several of which you can configure, as described in [Table 5-1](#). Additionally, differences in ports used when communicating with Fleet Patching and Provisioning Clients versus targets are noted.

Table 5-1 Fleet Patching and Provisioning Communication Ports

Client, Target, or Both	Protocol	Port	Purpose	Description
Target	TCP	22	SSH	Authentication-based operations involving clientless targets.
Client	TCP	22	SSH	Provisioning of Oracle Grid Infrastructure 12c release 2 (12.2.0.1) or later requires an SSH port. (Subsequent Fleet Patching and Provisioning Server/Client interactions use the JMX path.)

Table 5-1 (Cont.) Fleet Patching and Provisioning Communication Ports

Client, Target, or Both	Protocol	Port	Purpose	Description
Client	TCP	8896	JMX registry and server port	<p>For establishing Fleet Patching and Provisioning Server communication with Fleet Patching and Provisioning Clients.</p> <p>You can configure this port using the <code>srvctl modify rhpserver -port port_number</code> command.</p> <p>Note: The preceding command requires you to restart the service.</p>
Client	UDP	Port 53 must be open only on the Fleet Patching and Provisioning Server to accept incoming connections.	GNS port	<p>GNS is used for Fleet Patching and Provisioning Clients to locate the Fleet Patching and Provisioning Server.</p> <p>You can configure GNS with or without zone delegation.</p>
Both	TCP	<p>Specify one port for each RHPCTL command from ephemeral port range, or one port as specified with the <code>srvctl modify rhpserver -pl_port port_number</code> command.</p> <p>Progress listener port must be open to accept incoming connections on the server where RHPCTL is run (whether a Fleet Patching and Provisioning Server or Client).</p>	Command progress listener	<p>Fleet Patching and Provisioning Server opens a random port from the ephemeral range to monitor progress on the client or target, or uses the fixed port you specify with the <code>srvctl modify rhpserver -pl_port port_number</code> and multiplexes across clients/targets.</p>

Table 5-1 (Cont.) Fleet Patching and Provisioning Communication Ports

Client, Target, or Both	Protocol	Port	Purpose	Description
Clients running Oracle Database 18c or Oracle Database 12c release 2 (12.2.0.1) with the January 2018 RU, or later	TCP	You must have six ports for gold image provisioning on a Fleet Patching and Provisioning Client server to accept incoming connections.	Gold image transfers to Oracle Database 18c clients and clients with the 2018 January RU (running the <code>rhpcctl add workingcopy</code> command on the Fleet Patching and Provisioning Server)	Transferring copies of gold images from the Fleet Patching and Provisioning Server to clients uses six ports chosen from the ephemeral range, or six ports from the range you define using the <code>srvctl modify rhpserver - port_range port_number_range</code> . In this case, specify a range to accommodate the anticipated degree of parallelism (concurrent transfers) you will use. If the <code>port_range</code> value is specified and if RHPS manages 12.2 clients, then the client must be patched to 12.2.0.1.180717OCWJUL2018 RU or later.
Both	TCP/UDP	Fixed and configurable ports, as described in Table 5-2	NFS NFS is not required if RHPC clusters are at release 18c (18.4) or later.	Fleet Patching and Provisioning Server transfers software homes to targets and Oracle Database 12c release 2 (12.2.0.1) pre-January 2018 RU clients using temporary NFS mount points. NFS is also used for remote command execution support on clients and targets (all versions).

Table 5-2 Ports for NFS

Port	Description
2049, 111	Fixed NFS ports
Six ports chosen from ephemeral range (default) or defined in <code>/etc/sysconfig/nfs</code>	Configurable NFS ports, with each item followed by its configuration file entry: <pre>rpc: STATD_PORT rpc status out: STATD_OUTGOING_PORT rpc quota: RQUOTAD_PORT rpc mountd: MOUNTD_PORT rpc lockd tcp: LOCKD_TCPPORT rpc lockd udp: LOCKD_UDPPORT</pre>

Fleet Patching and Provisioning Server

The Fleet Patching and Provisioning Server is a highly available software provisioning system that uses Oracle Automatic Storage Management (Oracle ASM), Oracle

Automatic Storage Management Cluster File System (Oracle ACFS), Grid Naming Service (GNS), and other components.

The Fleet Patching and Provisioning Server primarily acts as a central server for provisioning Oracle homes, making them available to Fleet Patching and Provisioning Client and targets.

Features of the Fleet Patching and Provisioning Server:

- Efficiently stores gold images and image series for the managed homes, including separate binaries, and metadata related to users, roles, and permissions.
- Provides a list of available homes to clients upon request.
- Patch a software home once and then deploy the home to any Fleet Patching and Provisioning Client or any other target, instead of patching every site.
- Provides the ability to report on existing deployments.
- Deploys homes on physical servers and virtual machines.
- Notifies subscribers of changes to image series.
- Maintains an audit log of all RHPCTL commands run.

Fleet Patching and Provisioning Targets

Computers of which Fleet Patching and Provisioning is aware are known as targets.

Fleet Patching and Provisioning Servers can create new targets, and can also install and configure Oracle Grid Infrastructure on targets with only an operating system installed. Subsequently, Fleet Patching and Provisioning Server can provision database and other software on those targets, perform maintenance, scale the target cluster, in addition to many other operations. All Fleet Patching and Provisioning commands are run on the Fleet Patching and Provisioning Server. Targets running the Fleet Patching and Provisioning Client in Oracle Clusterware 12c release 2 (12.2), and later, may also run many of the Fleet Patching and Provisioning commands to request new software from the Fleet Patching and Provisioning Server and initiate maintenance themselves, among other tasks.

Note:

If you have targets running the Fleet Patching and Provisioning Client in Oracle Clusterware prior to 12c release 2 (12.2), then you can import images using the RHPCTL utility without making them clients. For clients running in Oracle Clusterware 12c release 2 (12.2), and later, you must configure and enable Fleet Patching and Provisioning Clients to simplify connectivity and credential issues.

Fleet Patching and Provisioning Clients

The Fleet Patching and Provisioning Client is part of the Oracle Grid Infrastructure. Users operate on a Fleet Patching and Provisioning Client to perform tasks such as requesting deployment of Oracle homes and listing available gold images.

When a user requests an Oracle home specifying a gold image, the Fleet Patching and Provisioning Client communicates with the Fleet Patching and Provisioning Server

to pass on the request. The Fleet Patching and Provisioning Server processes the request by instantiating a working copy of the gold image and making it available to the Fleet Patching and Provisioning Client using Oracle ACFS or a different local file system.

The Fleet Patching and Provisioning Client:

- Can use Oracle ACFS to store working copies of gold images which can be rapidly provisioned as local homes; new homes can be quickly created or undone using Oracle ACFS snapshots.

 **Note:**

Oracle supports using other local file systems besides Oracle ACFS.

- Provides a list of available homes from the Fleet Patching and Provisioning Server.
- Has full functionality in Oracle Clusterware 12c release 2 (12.2) and can communicate with Fleet Patching and Provisioning Servers from Oracle Clusterware 12c release 2 (12.2), or later.

Related Topics

- [Creating a Fleet Patching and Provisioning Client](#)
Users operate on a Fleet Patching and Provisioning Client to perform tasks such as requesting deployment of Oracle homes and querying gold images.

Authentication Options for Fleet Patching and Provisioning Operations

Some RHPCTL commands show authentication choices as an optional parameter.

Specifying an authentication option is not required when running an RHPCTL command on a Fleet Patching and Provisioning Client, nor when running an RHPCTL command on the Fleet Patching and Provisioning Server and operating on a Fleet Patching and Provisioning Client, because the server and client establish a trusted relationship when the client is created, and authentication is handled internally each time a transaction takes place. (The only condition for server/client communication under which an authentication option must be specified is when the server is provisioning a new Oracle Grid Infrastructure deployment—in this case, the client does not yet exist.)

To operate on a target that is not a Fleet Patching and Provisioning Client, you must provide the Fleet Patching and Provisioning Server with information allowing it to authenticate with the target. The options are as follows:

- Provide the `root` password (on `stdin`) for the target
- Provide the `sudo` user name, `sudo` binary path, and the password (`stdin`) for target
- Provide a password (either `root` or `sudouser`) non-interactively from local encrypted store (using the `-cred` authentication parameter)
- Provide a path to the identity file stored on the Fleet Patching and Provisioning Server for SSL-encrypted passwordless authentication (using the `-auth sshkey` option)

Passwordless Authentication Details

The Fleet Patching and Provisioning Server can authenticate to targets over SSH using a key pair. To enable this option, you must establish user equivalence between the `crsusr` on the Fleet Patching and Provisioning Server and `root` or a `sudouser` on the target.



Note:

The steps to create that equivalence are platform-dependent and so not shown in detail here. For Linux, see commands `ssh-keygen` to be run on the target and `ssh-copy-id` to be run on the Fleet Patching and Provisioning Server.

For example, assuming that you have established user equivalency between `crsusr` on the Fleet Patching and Provisioning Server and `root` on the target node, `nonRHPCClient4004.example.com`, and saved the key information on the Fleet Patching and Provisioning Server at `/home/oracle/rhp/ssh-key/key` `-path`, then the following command will provision a copy of the specified gold image to the target node with passwordless authentication:

```
$ rhpctl add workingcopy -workingcopy db12102_160607wc1 -image
db12102_160607
  -targetnode nonRHPCClient4004.example.com -path /u01/app/oracle/12.1/rhp/
dbhome_1
  -oraclebase /u01/app/oracle -auth sshkey -arg1 user:root -arg2
  identity_file:/home/oracle/rhp/ssh-key/key
```

For equivalency between `crsusr` on the Fleet Patching and Provisioning Server and a privileged user (other than `root`) on the target, the `-auth` portion of the command would be similar to the following:

```
-auth sshkey -arg1 user:ssh_user -arg2
identity_file:path_to_identity_file_on_RHPS
  -arg3 sudo_location:path_to_sudo_binary_on_target
```

Related Topics

- [rhpctl add credentials](#)
- [rhpctl delete credentials](#)
- [rhpctl add workingcopy](#)
Creates a working copy on a client cluster.

Fleet Patching and Provisioning Roles

An administrator assigns roles to Fleet Patching and Provisioning users with access-level permissions defined for each role. Users on Fleet Patching and Provisioning Clients are also assigned specific roles. Fleet Patching and Provisioning includes basic built-in and composite built-in roles.

Basic Built-In Roles

The basic built-in roles and their functions are:

- **GH_ROLE_ADMIN:** An administrative role for everything related to roles. Users assigned this role are able to run `rhptcl verb role` commands.
- **GH_SITE_ADMIN:** An administrative role for everything related to Fleet Patching and Provisioning Clients. Users assigned this role are able to run `rhptcl verb client` commands.
- **GH_SERIES_ADMIN:** An administrative role for everything related to image series. Users assigned this role are able to run `rhptcl verb series` commands.
- **GH_SERIES_CONTRIB:** Users assigned this role can add images to a series using the `rhptcl insertimage series` command, or delete images from a series using the `rhptcl deleteimage series` command.
- **GH_WC_ADMIN:** An administrative role for everything related to working copies of gold images. Users assigned this role are able to run `rhptcl verb workingcopy` commands.
- **GH_WC_OPER:** A role that enables users to create a working copy of a gold image for themselves or others using the `rhptcl add workingcopy` command with the `-user` option (when creating for others). Users assigned this role do not have administrative privileges and can only administer the working copies of gold images that they create.
- **GH_WC_USER:** A role that enables users to create a working copy of a gold image using the `rhptcl add workingcopy` command. Users assigned this role do not have administrative privileges and can only delete working copies that they create.
- **GH_IMG_ADMIN:** An administrative role for everything related to images. Users assigned this role are able to run `rhptcl verb image` commands.
- **GH_IMG_USER:** A role that enables users to create an image using the `rhptcl add | import image` commands. Users assigned this role do not have administrative privileges and can only delete images that they create.
- **GH_IMG_TESTABLE:** A role that enables users to add a working copy of an image that is in the `TESTABLE` state. Users assigned this role must also be assigned either the `GH_WC_ADMIN` role or the `GH_WC_USER` role to add a working copy.
- **GH_IMG_RESTRICT:** A role that enables users to add a working copy from an image that is in the `RESTRICTED` state. Users assigned this role must also be assigned either the `GH_WC_ADMIN` role or the `GH_WC_USER` role to add a working copy.
- **GH_IMG_PUBLISH:** Users assigned this role can promote an image to another state or retract an image from the `PUBLISHED` state to either the `TESTABLE` or `RESTRICTED` state.
- **GH_IMG_VISIBILITY:** Users assigned this role can modify access to promoted or published images using the `rhptcl allow | disallow image` commands.

Composite Built-In Roles

The composite built-in roles and their functions are:

- **GH_SA:** The Oracle Grid Infrastructure user on a Fleet Patching and Provisioning Server automatically inherits this role.

The GH_SA role includes the following basic built-in roles: GH_ROLE_ADMIN, GH_SITE_ADMIN, GH_SERIES_ADMIN, GH_SERIES_CONTRIB, GH_WC_ADMIN, GH_IMG_ADMIN, GH_IMG_TESTABLE, GH_IMG_RESTRICT, GH_IMG_PUBLISH, and GH_IMG_VISIBILITY.

- **GH_CA:** The Oracle Grid Infrastructure user on a Fleet Patching and Provisioning Client automatically inherits this role.

The GH_CA role includes the following basic built-in roles: GH_SERIES_ADMIN, GH_SERIES_CONTRIB, GH_WC_ADMIN, GH_IMG_ADMIN, GH_IMG_TESTABLE, GH_IMG_RESTRICT, GH_IMG_PUBLISH, and GH_IMG_VISIBILITY.

- **GH_OPER:** This role includes the following built-in roles: GH_WC_OPER, GH_SERIES_ADMIN, GH_IMG_TESTABLE, GH_IMG_RESTRICT, and GH_IMG_USER. Users assigned this role can delete only images that they have created.

Consider a gold image called `G1` that is available on the Fleet Patching and Provisioning Server.

Further consider that a user, `U1`, on a Fleet Patching and Provisioning Client, `CL1`, has the `GH_WC_USER` role. If `U1` requests to provision an Oracle home based on the gold image `G1`, then `U1` can do so, because of the permissions granted by the `GH_WC_USER` role. If `U1` requests to delete `G1`, however, then that request would be denied because the `GH_WC_USER` role does not have the necessary permissions.

The Fleet Patching and Provisioning Server can associate user-role mappings to the Fleet Patching and Provisioning Client. After the Fleet Patching and Provisioning Server delegates user-role mappings, the Fleet Patching and Provisioning Client can then modify user-role mappings on the Fleet Patching and Provisioning Server for all users that belong to the Fleet Patching and Provisioning Client. This is implied by the fact that only the Fleet Patching and Provisioning Server qualifies user IDs from a Fleet Patching and Provisioning Client site with the client cluster name of that site. Thus, the Fleet Patching and Provisioning Client `CL1` will not be able to update user mappings of a user on `CL2`, where `CL2` is the cluster name of a different Fleet Patching and Provisioning Client.

Fleet Patching and Provisioning Images

By default, when you create a gold image using either `rhptl import image` or `rhptl add image`, the image is ready to provision working copies. However, under certain conditions, you may want to restrict access to images and require someone to test or validate the image before making it available for general use.

You can also create a set of gold images on the Fleet Patching and Provisioning Server that can be collectively categorized as a gold image series which relate to each other, such as identical release versions, gold images published by a particular user, or images for a particular department within an organization.

Related Topics

- [Image State](#)
- [Image Series](#)
An image series is a convenient way to group different gold images into a logical sequence.
- [Image Type](#)
When you add or import a gold image, you must specify an image type.

Gold Image Distribution Among Fleet Patching and Provisioning Servers

Fleet Patching and Provisioning can automatically share and synchronize gold images between Fleet Patching and Provisioning Servers.

In the Fleet Patching and Provisioning architecture, one Fleet Patching and Provisioning Server manages a set of Fleet Patching and Provisioning Clients and targets within a given data center (or network segment of a data center). If you have more than one data center or a segmented data center, you must have more than one Fleet Patching and Provisioning Server.

In the Fleet Patching and Provisioning architecture, one Fleet Patching and Provisioning Server manages a set of Fleet Patching and Provisioning Clients and targets within a given data center (or network segment of a data center). If you have more than one data center or a segmented data center, then you must have more than one Fleet Patching and Provisioning Server to facilitate large-scale standardization across multiple estates.

Fleet Patching and Provisioning Servers retain the ability to create and manage gold images private to their scope, so local customizations are seamlessly supported.

You must first establish a peer relationship between two Fleet Patching and Provisioning Servers. Registration uses the names of the Fleet Patching and Provisioning Server clusters. The names of the two clusters can be the same but there is one naming restriction: a Fleet Patching and Provisioning Server (`RHPS_1`, for example) cannot register a peer Fleet Patching and Provisioning Server if that peer has the same name as a Fleet Patching and Provisioning Client or target within the management domain of `RHPS_1`.

1. On one Fleet Patching and Provisioning Server, create a file containing the server's configuration information, as follows:

```
$ rhpctl export server -serverdata file_path
```

2. Copy the file to a second Fleet Patching and Provisioning Server.
3. Complete the registration of the second Fleet Patching and Provisioning Server, as follows:

```
$ rhpctl register server -server server_cluster_name -serverdata file {-  
root  
| {-sudouser user_name -sudopath sudo_path}}
```

The credentials are the login credentials for the first Fleet Patching and Provisioning Server.

Once you register a Fleet Patching and Provisioning Server as a peer, the following command displays the peer (or peers) of the server:

```
$ rhpctl query peerserver
```

You can inspect the images on a peer Fleet Patching and Provisioning Server, as follows:

```
$ rhpctl query image -server server_cluster_name
```

The preceding command displays all images on a specific peer Fleet Patching and Provisioning Server. Additionally, you can specify a peer server along with the `-image image_name` parameter to display details of a specific image on a specific peer server.

A Fleet Patching and Provisioning Server can have multiple peers. Oracle does not support chained relationships between peers, however, such as, if RHPS_1 is a peer of RHPS_2, and RHPS_2 is also a peer of RHPS_3, then no relationship is established or implied between RHPS_1 and RHPS_3, although you can make them peers if you want.

Retrieve a copy or copies of gold images from a peer Fleet Patching and Provisioning Server, as follows:

```
$ rhpctl instantiate image -server server_cluster_name
```

Running the `rhpctl instantiate image` command activates an auto-update mechanism. From that point on, when you create gold images on a peer Fleet Patching and Provisioning Server (such as RHPS_2), they are candidates for being automatically copied to the Fleet Patching and Provisioning Server that performed the instantiate operation (such as RHPS_1). Whether a new gold image is automatically copied depends on that image's relevance to any instantiate parameters that you may include in the command:

- `-all`: Creates an automatic push for all gold images created on RHPS_2 to RHPS_1
- `-image image_name`: Creates an automatic push for all new descendant gold images of the named image created on RHPS_2 to RHPS_1. A descendant of the named image is an image that is created on RHPS_2 using the `rhpctl add image` command.
- `-series series_name`: Creates an automatic push for all gold images added to the named series on RHPS_2 to RHPS_1
- `-imagetype image_type`: Creates an automatic push for all gold images created of the named image type on RHPS_2 to RHPS_1

To stop receiving updates that were established by the `rhpctl instantiate image` command, run `rhpctl unstantiate image` and specify the peer Fleet Patching and Provisioning Server and one of the following: all, image name, image series name, or image type.

End the peer relationship, as follows, on any one of the Fleet Patching and Provisioning Servers:

```
$ rhpctl unregister server -server server_cluster_name
```

Related Topics

- [rhpctl export server](#)
- [rhpctl register server](#)
- [rhpctl query peerserver](#)
- [rhpctl query image](#)
- [rhpctl instantiate image](#)
- [rhpctl unstantiate image](#)
- [rhpctl unregister server](#)

Fleet Patching and Provisioning Server Auditing

The Fleet Patching and Provisioning Server records the execution of all Fleet Patching and Provisioning operations, and also records whether those operations succeeded or failed.

An audit mechanism enables administrators to query the audit log in a variety of dimensions, and also to manage its contents and size.

Fleet Patching and Provisioning Notifications

The Fleet Patching and Provisioning Server is the central repository for the software homes available to the data center. Therefore, it is essential for administrators throughout the data center to be aware of changes to the inventory that may impact their areas of responsibility.

You can create subscriptions to image series events. Fleet Patching and Provisioning notifies a subscribed role or number of users by email of any changes to the images available in the series, including addition or removal of an image. Each series may have a unique group of subscribers.

Also, when a working copy of a gold image is added to or deleted from a target, the owner of the working copy and any additional users can be notified by email. If you want to enable notifications for additional Fleet Patching and Provisioning events, you can create a user-defined action as described in the next section.

Fleet Patching and Provisioning Implementation

Implementing Fleet Patching and Provisioning involves creating a Fleet Patching and Provisioning Server, adding gold images to the server, and creating working copies of gold images to provision software.

After you install and configure Oracle Clusterware, you can configure and start using Fleet Patching and Provisioning. You must create a Fleet Patching and Provisioning Server where you create and store gold images of database and other software homes.

Creating a Fleet Patching and Provisioning Server

The Fleet Patching and Provisioning Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and targets.

To create a Fleet Patching and Provisioning Server:

1. Use the Oracle ASM configuration assistant (ASMCA) to create an Oracle ASM disk group on the Fleet Patching and Provisioning Server to store software, as follows:

```
$ Grid_home/bin/asmca
```

Because this disk group is used to store software, Oracle recommends a minimum of 50 GB for this disk group.

 **Note:**

You must set Oracle ASM Dynamic Volume Manager (Oracle ADVM) compatibility settings for this disk group to 12.1.

2. Provide a mount path that exists on all nodes of the cluster. The Fleet Patching and Provisioning Server uses this path to mount gold images.

```
$ mkdir -p storage_path/images
```

3. As root, create the Fleet Patching and Provisioning Server resource, as follows:

```
# Grid_home/bin/srvctl add rhpserver -storage storage_path  
-diskgroup disk_group_name
```

4. Start the Fleet Patching and Provisioning Server, as follows:

```
$ Grid_home/bin/srvctl start rhpserver
```

After you start the Fleet Patching and Provisioning Server, use the Fleet Patching and Provisioning Control (RHPCTL) utility to further manage Fleet Patching and Provisioning.

Related Topics

- [Oracle Automatic Storage Management Administrator's Guide](#)
- [Fleet Patching and Provisioning Control \(RHPCTL\) Command Reference](#)
Use the Fleet Patching and Provisioning Control (RHPCTL) utility to manage Fleet Patching and Provisioning in your cluster.
- [Server Control \(SRVCTL\) Command Reference](#)
Use the Server Control (SRVCTL) utility to manage various components and applications in your cluster.

Adding Gold Images to the Fleet Patching and Provisioning Server

Use RHPCTL to add gold images for later provisioning of software.

The Fleet Patching and Provisioning Server stores and serves gold images of software homes. These images must be instantiated on the Fleet Patching and Provisioning Server.

 **Note:**

Images are read-only, and you cannot run programs from them. To create a usable software home from an image, you must create a working copy of a gold image. You cannot directly use images as software homes. You can, however, use images to create working copies (software homes).

You can import software to the Fleet Patching and Provisioning Server using any one of the following methods:

- You can import an image from an installed home on the Fleet Patching and Provisioning Server using the following command:

```
rhpcctl import image -image image_name -path path_to_installed_home
  [-imagetype ORACLEDBSOFTWARE | ORACLEGISoftware | ORACLEGGSoftware |
  SOFTWARE]
```

- You can import an image from an installed home on a Fleet Patching and Provisioning Client, using the following command run from the Fleet Patching and Provisioning Client:

```
rhpcctl import image -image image_name -path path_to_installed_home
```

- You can create an image from an existing working copy using the following command:

```
rhpcctl add image -image image_name -workingcopy working_copy_name
```

Use the first two commands in the preceding list to seed the image repository, and to add additional images over time. Use the third command on the Fleet Patching and Provisioning Server as part of the workflow for creating a gold image that includes patches applied to a pre-existing gold image.

The preceding three commands also create an Oracle ACFS file system in the Fleet Patching and Provisioning root directory, similar to the following:

```
/u01/rhp/images/images/RDBMS_121020617524
```

Related Topics

- [Patching Oracle Database Software](#)
To patch an Oracle database, you move the database home to a new home, which includes the patches you want to implement.

- [RHPCTL Command Reference](#)
This section describes RHPCTL command usage information, and lists and describes RHPCTL commands.

Image State

You can set the state of an image to `TESTABLE` or `RESTRICTED` so that only users with the `GH_IMG_TESTABLE` or `GH_IMG_RESTRICT` roles can provision working copies from this image. Once the image has been tested or validated, you can change the state and make the image available for general use by running the `rhpcctl promote image -image image_name -state PUBLISHED` command. The default image state is `PUBLISHED` when you add a new gold image, but you can optionally specify a different state with the `rhpcctl add image` and `rhpcctl import image` commands.

Image Series

An image series is a convenient way to group different gold images into a logical sequence.

Fleet Patching and Provisioning treats each image as an independent entity with respect to other images. No relationship is assumed between images, even if they follow some specific nomenclature. The image administrator may choose to name images in a logical manner that makes sense to the user community, but this does not create any management grouping within the Fleet Patching and Provisioning framework.

Use the `rhpcctl add series` command to create an image series and associate one or more images to this series. The list of images in an image series is an ordered list. Use the `rhpcctl insertimage series` and `rhpcctl deleteimage series` to add and delete images in an image series. You can also change the order of images in a series using these commands.

The `insertimage` and `deleteimage` commands do not instantiate or delete actual gold images but only change the list. Also, an image can belong to more than one series (or no series at all).

Image Type

When you add or import a gold image, you must specify an image type.

Oracle Clusterware provides the following built-in base image types:

```
ORACLEDBSOFTWARE
ORACLEGISoftware
ORACLEGGSoftware
SOFTWARE
```

Every gold image must have an image type, and you can create your own image types. A new image type must be based on one of the built-in types. The image type directs Fleet Patching and Provisioning to apply its capabilities for managing Oracle Grid Infrastructure and Oracle Database homes. Fleet Patching and Provisioning also uses image type to organize the custom workflow support framework.

Creating a Custom Image Type

Use the `rhpcctl add imagetype` command to create custom image types.

For example, to create an image type called DBTEST, which is based on the ORACLEDBSOFTWARE image type:

```
$ rhpctl add imagetype -imagetype DBTEST -basetype ORACLEDBSOFTWARE
```

 **Note:**

When you create an image type that is based on an existing image type, the new image type does not inherit any user actions (for custom workflow support) from the base type.

Provisioning Copies of Gold Images

Use RHPCTL to provision copies of gold images to Fleet Patching and Provisioning Servers, Clients, and targets.

After you create and import a gold image, you can provision software by adding a copy of the gold image (called a working copy) on the Fleet Patching and Provisioning Server, on a Fleet Patching and Provisioning Client, or a target. You can run the software provisioning command on either the Server or a Client.

- To create a working copy on the Fleet Patching and Provisioning Server:

```
$ rhpctl add workingcopy -workingcopy working_copy_name -image  
image_name
```

- To create a working copy in a local file system on a Fleet Patching and Provisioning Client:

```
$ rhpctl add workingcopy -workingcopy working_copy_name -image  
image_name  
-storagetype LOCAL -path path_to_software_home
```

- To create a working copy on a Fleet Patching and Provisioning Client from the Fleet Patching and Provisioning Server:

```
$ rhpctl add workingcopy -workingcopy working_copy_name -image  
image_name  
-client client_cluster_name
```

 **Note:**

- The directory you specify in the `-path` parameter must be empty.
- You can re-run the provisioning command in case of an interruption or failure due to system or user errors. After you fix the reported errors, re-run the command and it will resume from the point of failure.

Related Topics

- [Storage Options for Provisioned Software](#)
Choose one of three storage options where Fleet Patching and Provisioning stores working copies of gold images.

User Group Management in Fleet Patching and Provisioning

When you create a working copy of a gold image as part of a move or upgrade operation, Fleet Patching and Provisioning configures the operating system groups in the new working copy to match those of the source software home (either the unmanaged or the managed home from which you move or upgrade).

When you create a gold image of SOFTWARE image type, any user groups in the source are not inherited and images of this type never contain user group information. When you provision a working copy from a SOFTWARE gold image using the `rhpcctl add workingcopy` command, you can, optionally, configure user groups in the working copy using the `-groups` parameter.

The `rhpcctl move database`, `rhpcctl move gihome`, `rhpcctl upgrade database`, and `rhpcctl upgrade gihome` commands all require you to specify a source home (either an unmanaged home or a managed home (working copy) that you provisioned using Fleet Patching and Provisioning), and a destination home (which must be a working copy).

When you have provisioned the destination home using the `rhpcctl add workingcopy` command, prior to performing a move or upgrade operation, you must ensure that the groups configured in the source home match those in the destination home. Fleet Patching and Provisioning configures the groups as part of the add operation.

When you create a gold image of either the ORACLEGISOFTWARE or the ORACLEDBSOFTWARE image type from a source software home (using the `rhpcctl import image` command) or from a working copy (using the `rhpcctl add image` command), the gold image inherits the Oracle user groups that were configured in the source. You cannot override this feature.

You can define user groups for ORACLEGISOFTWARE and ORACLEDBSOFTWARE working copies using the `rhpcctl add workingcopy` command, depending on the image type and user group, as discussed in the subsequent sections.

This section describes how Fleet Patching and Provisioning manages user group configuration, and how the `-groups` command-line option of `rhpcctl add workingcopy` functions.

ORACLEGISOFTWARE (Oracle Grid Infrastructure 11g release 2 (11.2), and 12c release 1 (12.1) and release 2 (12.2))

When you provision an Oracle Grid Infrastructure working copy of a gold image, the groups are set in the working copy according to the type of provisioning (whether regular provisioning or software only, and with or without the `-local` parameter), and whether you specify the `-groups` parameter with `rhpcctl add workingcopy`. You can define OSDBA and OSASM user groups in Oracle Grid Infrastructure software with either the `-softwareonly` command parameter or by using a response file with the `rhpcctl add workingcopy` command.

If you are provisioning only the Oracle Grid Infrastructure software using the `-softwareonly` command parameter, then you cannot use the `-groups` parameter, and

Fleet Patching and Provisioning obtains OSDBA and OSASM user group information from the active Grid home.

If you use the `-local` command parameter (which is only valid when you use the `-softwareonly` command parameter) with `rhpcctl add workingcopy`, then Fleet Patching and Provisioning takes the values of the groups from the command line (using the `-groups` parameter) or uses the default values, which Fleet Patching and Provisioning obtains from the `osdbagrps` binary of the gold image.

If none of the preceding applies, then Fleet Patching and Provisioning uses the installer default user group.

If you are provisioning and configuring a working copy using information from a response file, then Fleet Patching and Provisioning:

1. Uses the value of the user group from the command line, if provided, for OSDBA or OSASM, or both.
2. If you provide no value on the command line, then Fleet Patching and Provisioning retrieves the user group information defined in the response file.

If you are defining the OSOPER Oracle group, then, again, you can either use the `-softwareonly` command parameter or use a response file with the `rhpcctl add workingcopy` command.

If you use the `-softwareonly` command parameter, then you can provide the value on the command line (using the `-groups` parameter) or leave the user group undefined.

If you are provisioning and configuring a working copy of a gold image using information from a response file, then you can provide the value on the command line, use the information contained in the response file, or leave the OSOPER Oracle group undefined.

ORACLEDBSOFTWARE (Oracle Database 11g release 2 (11.2), and 12c release 1 (12.1) and release 2 (12.2))

If you are provisioning a working copy of Oracle Database software and you want to define Oracle groups, then use the `-groups` command parameter with the `rhpcctl add workingcopy` command. Oracle groups available in the various Oracle Database releases are as follows:

- Oracle Database 11g release 2 (11.2)
 - OSDBA
 - OSOPER
- Oracle Database 12c release 1 (12.1)
 - OSDBA
 - OSOPER
 - OSBACKUP
 - OSDG
 - OSKM
- Oracle Database 12c release 2 (12.2)
 - OSDBA
 - OSOPER
 - OSBACKUP
 - OSDG

OSKM
OSRAC

Regardless of which of the preceding groups you are defining (except for OSOPER), Fleet Patching and Provisioning takes the values of the groups from the command line (using the `-groups` parameter) or uses the default values, which Fleet Patching and Provisioning obtains from the `osdbagrp` binary of the gold image.

If any group picked up from the `osdbagrp` binary is not in the list of groups to which the database user belongs (given by the `id` command), then Fleet Patching and Provisioning uses the installer default user group. Otherwise, the database user is the user running the `rhptcl add workingcopy` command.

Storage Options for Provisioned Software

Choose one of three storage options where Fleet Patching and Provisioning stores working copies of gold images.

When you provision software using the `rhptcl add workingcopy` command, you can choose from three storage options where Fleet Patching and Provisioning places that software:

- In an Oracle ACFS shared file system managed by Fleet Patching and Provisioning (for database homes only)
- In a local file system not managed by Fleet Patching and Provisioning

Using the `rhptcl add workingcopy` command with the `-storagetype` and `-path` parameters, you can choose where you store provisioned working copies. The applicability of the parameters depends on whether the node (or nodes) to which you are provisioning the working copy is a Fleet Patching and Provisioning Server, Fleet Patching and Provisioning Client, or a non-Fleet Patching and Provisioning client. You can choose from the following values for the `-storagetype` parameter:

- `RHP_MANAGED`: Choosing this value, which is available for Fleet Patching and Provisioning Servers and Fleet Patching and Provisioning Clients, stores working copies in an Oracle ACFS shared file system. The `-path` parameter is not used with this option because Fleet Patching and Provisioning manages the storage option.

 **Notes:**

- You cannot store Oracle Grid Infrastructure homes in `RHP_MANAGED` storage.
- Oracle recommends using the `RHP_MANAGED` storage type, which is available on Fleet Patching and Provisioning Servers, and on Clients configured with an Oracle ASM disk group.
- If you provision working copies on a Fleet Patching and Provisioning Server, then you do not need to specify the `-storagetype` option because it will default to `RHP_MANAGED`.
- If you choose to provision working copies on a Fleet Patching and Provisioning Client, and you do not specify the `-path` parameter, then the storage type defaults to `RHP_MANAGED` only if there is an Oracle ASM disk group on the client. Otherwise the command will fail. If you specify a location on the client for the `-path` parameter, then the storage type defaults to `LOCAL` with or without an Oracle ASM disk group.

- `LOCAL`: Choosing this value stores working copies in a local file system that is not managed by Fleet Patching and Provisioning. You must specify a path to the file system on the Fleet Patching and Provisioning Server, Fleet Patching and Provisioning Client, or non-Fleet Patching and Provisioning client, or to the Oracle ASM disk group on the Fleet Patching and Provisioning Client.

In cases where you specify the `-path` parameter, if the file system is shared among all of the nodes in the cluster, then the working copy gets created on this shared storage. If the file system is not shared, then the working copy gets created in the location of the given path on every node in the cluster.

 **Note:**

The directory you specify in the `-path` parameter must be empty.

Related Topics

- [rhpctl add workingcopy](#)
Creates a working copy on a client cluster.

Provisioning for a Different User

If you want a different user to provision software other than the user running the command, then use the `-user` parameter of the `rhpctl add workingcopy` command.

When the provisioning is completed, all files and directories of the provisioned software are owned by the user you specified. Permissions on files on the remotely provisioned software are the same as the permissions that existed on the gold image from where you provisioned the application software.

Propagating Images Between Fleet Patching and Provisioning Servers

With automatic image propagation, you can set up automated copies of software images across different peer Fleet Patching and Provisioning Servers. Gold images that you register at one site are copied to peer Fleet Patching and Provisioning Servers.

In a peer-to-peer relationship between two Fleet Patching and Provisioning Servers, one server is the source for software images and the other is the destination for software images.

The criteria for copying software images between servers can be based on certain policies, such as software image type, image series, or all software images. When you register an image that meets the criteria for software image propagation, a copy of this image propagates to all registered peer servers.

The following example procedure establishes a relationship between two Fleet Patching and Provisioning Server sites, RHPS-A and RHPS-B, where RHPS-A is the source and RHPS-B is the destination for software images.

1. Run the following command on RHPS-B:

```
$ rhpctl export server -server RHPS-A -serverdata file_path
```

The preceding command creates a file named RHPS-B.xml in the directory path you specify for the `-serverdata` parameter.

2. Run the following command to register a peer server to the current Fleet Patching and Provisioning Server:

```
$ rhpctl register server -server RHPS-B -serverdata /tmp/RHPS-B.xml
```

The preceding command copies the RHPS-B.xml file that was created in the previous step to a location on the server where you run the command, which is `/tmp/RHPS-B.xml`, in this case.

The cluster in which you run the preceding command is the source site, and the server that you specify on the command line is the destination site to which software images are copied.

Use the `rhpctl unregister server` command to remove the peer-to-peer relationship.

3. Run the following command on RHPS-B to propagate software images from RHPS-A to RHPS-B:

```
$ rhpctl instantiate image -server RHPS-A -all
```

The preceding command propagates all images from RHPS-A to RHPS-B. If an image already exists on RHPS-B, then it is not propagated again.

 **Note:**

Propagation of images is based on the image name. RHPCTL does not compare the content of the images, themselves, to determine whether they are different. If an image you want to propagate has a non-unique name, then RHPCTL assumes that the images are identical and does not propagate the image.

Use the `rhpcctl unstantiate image` command to cancel the propagation of a particular image. Any propagation already in progress is not affected by this command, and will continue until complete. The `-all` parameter removes any values for the `-imagetype` and `-series` parameters.

Related Topics

- [rhpcctl export server](#)
- [rhpcctl register server](#)
- [rhpcctl unregister server](#)
- [rhpcctl instantiate image](#)
- [rhpcctl unstantiate image](#)

Oracle Grid Infrastructure Management

The Fleet Patching and Provisioning Server provides an efficient and secure platform for the distribution of Oracle Grid Infrastructure homes to targets and Fleet Patching and Provisioning Clients.

Also, Fleet Patching and Provisioning Clients have the ability to fetch Oracle Grid Infrastructure homes from the Fleet Patching and Provisioning Server.

Oracle Grid Infrastructure homes are distributed in the form of working copies of gold images. After a working copy has been provisioned, Fleet Patching and Provisioning can optionally configure Oracle Grid Infrastructure. This gives Fleet Patching and Provisioning the ability to create an Oracle Grid Infrastructure installation on a group of one or more nodes that initially do not have Oracle Grid Infrastructure installed.

Fleet Patching and Provisioning also has commands for managing Oracle Grid Infrastructure homes, such as switching to a patched home or upgrading to a new Oracle Grid Infrastructure version. These are both single commands that orchestrate the numerous steps involved. Reverting to the original home is just as simple. Also, Fleet Patching and Provisioning can add or delete nodes from an Oracle Grid Infrastructure configuration.

About Deploying Oracle Grid Infrastructure Using Oracle Fleet Patching and Provisioning

Oracle Fleet Patching and Provisioning (Oracle FPP) is a software lifecycle management method for provisioning and maintaining Oracle homes. Oracle Fleet Patching and Provisioning enables mass deployment and maintenance of standard operating environments for databases, clusters, and user-defined software types.

 **Note:**

Starting with Oracle Grid Infrastructure 19c, the feature formerly known as Rapid Home Provisioning (RHP) is now Oracle Fleet Patching and Provisioning (Oracle FPP).

Oracle Fleet Patching and Provisioning enables you to install clusters, and provision, patch, scale, and upgrade Oracle Grid Infrastructure, Oracle Restart, and Oracle Database homes. The supported versions are 11.2, 12.1, 12.2, 18c, and 19c. You can also provision applications and middleware using Oracle Fleet Patching and Provisioning.

Oracle Fleet Patching and Provisioning is a service in Oracle Grid Infrastructure that you can use in either of the following modes:

- **Central Oracle Fleet Patching and Provisioning Server**

The Oracle Fleet Patching and Provisioning Server stores and manages standardized images, called gold images. Gold images can be deployed to any number of nodes across the data center. You can create new clusters and databases on the deployed homes and can use them to patch, upgrade, and scale existing installations.

The Oracle Fleet Patching and Provisioning Server can manage the following types of installations:

- Software homes on the cluster hosting the Oracle Fleet Patching and Provisioning Server itself.
- Oracle Fleet Patching and Provisioning Clients running Oracle Grid Infrastructure 12c Release 2 (12.2), 18c, and 19c.
- Installations running Oracle Grid Infrastructure 11g Release 2 (11.2) and 12c Release 1 (12.1).
- Installations running without Oracle Grid Infrastructure.

The Oracle Fleet Patching and Provisioning Server can provision new installations and can manage existing installations without requiring any changes to the existing installations. The Oracle Fleet Patching and Provisioning Server can automatically share gold images among peer servers to support enterprises with geographically distributed data centers.

- **Oracle Fleet Patching and Provisioning Client**

The Oracle Fleet Patching and Provisioning Client can be managed from the Oracle Fleet Patching and Provisioning Server, or directly by executing commands on the client itself. The Oracle Fleet Patching and Provisioning Client is a service built into the Oracle Grid Infrastructure and is available in Oracle Grid Infrastructure 12c Release 2 (12.2) and later releases. The Oracle Fleet Patching and Provisioning Client can retrieve gold images from the Oracle Fleet Patching and Provisioning Server, upload new images based on the policy, and apply maintenance operations to itself.

Oracle Fleet Patching and Provisioning

Deploying Oracle software using Oracle Fleet Patching and Provisioning has the following advantages:

- Ensures standardization and enables high degrees of automation with gold images and managed lineage of deployed software.
- Minimizes downtime by deploying new homes as images (called gold images) out-of-place, without disrupting active databases or clusters.
- Simplifies maintenance by providing automations which are invoked with a simple, consistent API across database versions and deployment models.
- Reduces maintenance risk with built-in validations and a “dry run” mode to test the operations.
- Enables you to resume or restart the commands in the event of an unforeseen issue, reducing the risk of maintenance operations.
- Minimizes and often eliminates the impact of patching and upgrades, with features that include:
 - Zero-downtime database upgrade with fully automated upgrade, executed entirely within the deployment without requiring any extra nodes or external storage.
 - Adaptive management of database sessions and OJVM during rolling patching.
 - Options for management of consolidated deployments.
- The deployment and maintenance operations enable customizations to include environment-specific actions into the automated workflow.

Related Topics

- *Oracle Clusterware Administration and Deployment Guide*

Provisioning Oracle Grid Infrastructure Software

Fleet Patching and Provisioning has several methods to provision and, optionally, configure Oracle Grid Infrastructure and Oracle Restart grid infrastructure homes.

Fleet Patching and Provisioning can provision and configure Oracle Grid Infrastructure on one or more nodes that do not currently have a Grid home, and then configure Oracle Grid Infrastructure to form a single-node or multi-node Oracle Grid Infrastructure installation.

Use the `rhpcctl add workingcopy` command to install and configure Oracle Grid Infrastructure, and to enable simple and repeatable creation of standardized deployments.

The Fleet Patching and Provisioning Server can also provision an Oracle Grid Infrastructure home to a node or cluster that is currently running Oracle Grid Infrastructure. The currently running Grid home can be a home that Fleet Patching and Provisioning did not provision (an *unmanaged* home) or a home that Fleet Patching and Provisioning did provision (a *managed* home).

You can also provision an Oracle Restart grid infrastructure to a node in the cluster.

In either case, use the `-softwareonly` parameter of the `rhpcctl add workingcopy` command. This provisions but does not activate the new Grid home, so that when you are ready to switch to that new home, you can do so with a single command.

- To inform Fleet Patching and Provisioning the nodes on which to install Oracle Grid Infrastructure, and to configure Oracle Grid Infrastructure, you provide directions in a response file, as in the following example:

```
$ rhpctl add workingcopy -workingcopy GI_HOME_11204_WCPY -image  
GI_HOME_11204 -responsefile /u01/app/rhpinfo/GI_11204_install.txt  
  {authentication_option}
```

The preceding command provisions the GI_HOME_11204_WCPY working copy based on the GI_HOME_11204 gold image to a target specified in the GI_11204_install.txt response file. In addition to identifying the target nodes, the response file specifies information about the Oracle Grid Infrastructure configuration, such as Oracle ASM and GNS parameters.

 **Note:**

The `oracle.install.crs.rootconfig.executeRootScript=xxx` response file parameter is overridden and always set to `false` for Fleet Patching and Provisioning, regardless of what you specify in the response file.

- To provision an Oracle Grid Infrastructure home to a node or cluster that is currently running Oracle Grid Infrastructure:

```
$ rhpctl add workingcopy -workingcopy GI_HOME_12201_PATCHED_WCPY -image  
GI_HOME_12201_PSU1 -client CLUST_002 -softwareonly
```

The preceding command provisions a new working copy based on the GI_HOME_12201_PSU1 gold image to the Fleet Patching and Provisioning Client (that is running Oracle Grid Infrastructure 12c release 2 (12.2)) named CLUST_002. When you provision to a target that is not running Oracle Grid Infrastructure 12c release 2 (12.2) (such as, a target running Oracle Grid Infrastructure 12c release 1 (12.1) or Oracle Grid Infrastructure 11g release 2 (11.2)), use the `-targetnode` parameter instead of `-client`.

- Specify a target node on which you want to provision an Oracle Restart grid infrastructure, as follows:

```
$ rhpctl add workingcopy -workingcopy SIHA_GI -image goldimage -  
targetnode remote_node_name -resposnefile Oracle_Restart_resposne_file  
  {authentication_option}
```

Related Topics

- [Authentication Options for Fleet Patching and Provisioning Operations](#)
Some RHPCTL commands show authentication choices as an optional parameter.

Patching Oracle Grid Infrastructure Software

Fleet Patching and Provisioning provides three methods to patch Oracle Grid Infrastructure software homes: rolling, non-rolling, and in batches.

Patching Oracle Grid Infrastructure software involves moving the Grid home to a patched version of the current Grid home. When the patching operation is initiated by a Fleet Patching and Provisioning Server or Client, the patched version must be a working copy of a gold image. The working copy to which you are moving the Grid home can be at a lower patch level than the current home. This facilitates rollback if any problems occur after moving to the higher-level patched home.

You can also perform this operation using the independent automaton in an environment where no Fleet Patching and Provisioning Server is present. In this case, the source and destination homes are not working copies of gold images, but are two installed homes that you deployed with some method other than using Fleet Patching and Provisioning.

This section includes the following topics:

- [Patching Oracle Grid Infrastructure Using the Rolling Method](#)
- [Patching Oracle Grid Infrastructure Using the Non-Rolling Method](#)
- [Patching Oracle Grid Infrastructure Using Batches](#)
- [Combined Oracle Grid Infrastructure and Oracle Database Patching](#)

Related Topics

- [rhpctl add workingcopy](#)
- [rhpctl move gihome](#)

Patching Oracle Grid Infrastructure Using the Rolling Method

The rolling method for patching Oracle Grid Infrastructure is the default method.

You use the `rhpctl move gihome` command (an atomic operation), which returns after the Oracle Grid Infrastructure stack on each node has been restarted on the new home. Nodes are restarted sequentially, so that only one node at a time will be offline, while all other nodes in the cluster remain online.

- Move the Oracle Grid Infrastructure home to a working copy of the same release level, as follows:

```
$ rhpctl move gihome -client cluster_name -sourcewc Grid_home_1 -destwc  
Grid_home_2
```

The preceding command moves the running Oracle Grid Infrastructure home from the current managed home (the `sourcewc`) to the patched home (`destwc`) on the specific client cluster. The patched home must be provisioned on the client.

- If the move operation fails at some point before completing, then you can rerun the operation by running the command again and the operation will resume where it left off. This enables you to fix whatever problem caused the failure and resume

processing from the point of failure. Or you can undo the partially completed operation and return the configuration to its initial state, as follows:

```
$ rhpctl move gihome -destwc destination_workingcopy_name -revert  
[authentication_option]
```

You can use the `-revert` parameter with an un-managed home.

Notes:

- You cannot move the Grid home to a home that Fleet Patching and Provisioning does not manage. Therefore, rollback (to the original home) applies only to moves between two working copies. This restriction does not apply when using the independent automaton since it operates on unmanaged homes only.
- You can delete the source working copy at any time after moving a Grid home. Once you delete the working copy, however, you cannot perform a rollback. Also, use the `rhpctl delete workingcopy` command (as opposed to `rm`, for example) to remove the source working copy to keep the Fleet Patching and Provisioning inventory correct.
- If you use the `-abort` parameter to terminate the patching operation, then Fleet Patching and Provisioning does not clean up or undo any of the patching steps. The cluster, databases, or both may be in an inconsistent state because all nodes are not patched.

Patching Oracle Grid Infrastructure Using the Non-Rolling Method

You can use the `-nonrolling` parameter with the `rhpctl move gihome` command, which restarts the Oracle Grid Infrastructure stack on all nodes in parallel.

As with the rolling method, this is an atomic command which returns after all nodes are online.

- Use the following command to patch Oracle Grid Infrastructure in a non-rolling fashion:

```
$ rhpctl move gihome -client cluster_name -sourcewc Grid_home_1 -destwc  
Grid_home_2 -nonrolling
```

Patching Oracle Grid Infrastructure Using Batches

The third patching method is to sequentially process batches of nodes, with a number of nodes in each batch being restarted in parallel.

This method maximizes service availability during the patching process. When you patch Oracle Grid Infrastructure 12c release 2 (12.2.x) software homes, you can define the batches on the command line or choose to have Fleet Patching and Provisioning generate the list of batches based on its analysis of the database services running in the cluster.

There are two methods for defining batches:

- [User-Defined Batches](#)
- [Fleet Patching and Provisioning-Defined Batches](#)

User-Defined Batches

When you use this method of patching, the first time you run the `rhpcctl move gihome` command, you must specify the source home, the destination home, the batches, and other options, as needed. The command terminates after the first node restarts.

To patch Oracle Grid Infrastructure using batches that you define:

1. Define a list of batches on the command line and begin the patching process, as in the following example:

```
$ rhpcctl move gihome -sourcewc wc1 -destwc wc2 -batches "(n1),(n2,n3),  
(n4)"
```

The preceding command example initiates the move operation, and terminates and reports successful when the Oracle Grid Infrastructure stack restarts in the first batch. Oracle Grid Infrastructure restarts the batches in the order you specified in the `-batches` parameter.

In the command example, node `n1` forms the first batch, nodes `n2` and `n3` form the second batch, and node `n4` forms the last batch. The command defines the source working copy as `wc1` and the patched (destination) working copy as `wc2`.

Notes:

You can specify batches such that singleton services (policy-managed singleton services or administrator-managed services running on one instance) are relocated between batches and non-singleton services remain partially available during the patching process.

2. You must process the next batch by running the `rhpcctl move gihome` command, again, as follows:

```
$ rhpcctl move gihome -destwc wc2 -continue
```

The preceding command example restarts the Oracle Grid Infrastructure stack on the second batch (nodes `n2` and `n3`). The command terminates by reporting that the second batch was successfully patched.

3. Repeat the previous step until you have processed the last batch of nodes. If you attempt to run the command with the `-continue` parameter after the last batch has been processed, then the command returns an error.

If the `rhpcctl move gihome` command fails at any time during the above sequence, then, after determining and fixing the cause of the failure, rerun the command with the `-continue` option to attempt to patch the failed batch. If you want to skip the failed batch and continue with the next batch, use the `-continue` and `-skip` parameters. If you attempt to skip over the last batch, then the move operation is terminated.

Alternatively, you can reissue the command using the `-revert` parameter to undo the changes that have been made and return the configuration to its initial state.

You can use the `-abort` parameter instead of the `-continue` parameter at any point in the preceding procedure to terminate the patching process and leave the cluster in its current state.

Notes:

- Policy-managed services hosted on a server pool with one active server, and administrator-managed services with one preferred instance and no available instances cannot be relocated and will go OFFLINE while instances are being restarted.
- If a move operation is in progress, then you cannot initiate another move operation from the same source home or to the same destination home.
- After the move operation has ended, services may be running on nodes different from the ones they were running on before the move and you will have to manually relocate them back to the original instances, if necessary.
- If you use the `-abort` parameter to terminate the patching operation, then Fleet Patching and Provisioning does not clean up or undo any of the patching steps. The cluster, databases, or both may be in an inconsistent state because all nodes are not patched.
- Depending on the start dependencies, services that were offline before the move began could come online during the move.

Fleet Patching and Provisioning-Defined Batches

Using Fleet Patching and Provisioning to define and patch batches of nodes means that you need only run one command, as shown in the following command example, where the source working is `wc1` and the destination working copy is `wc2`:

```
$ rhpctl move gihome -sourcewc wc1 -destwc wc2 -smartmove -saf Z+ [-eval]
```

There is no need for you to do anything else unless you used the `-separate` parameter with the command. In that case, the move operation waits for user intervention to proceed to the next batch, and then you will have to run the command with the `-continue` parameter after each batch completes.

If the move operation fails at some point before completing, then you can either rerun the operation by running the command again, or you can undo the partially completed operation, as follows:

```
$ rhpctl move gihome -destwc destination_workingcopy_name -revert  
[authentication_option]
```

You can use the `-revert` parameter with an un-managed home.

The parameters used in the preceding example are as follows:

- `-smartmove`: This parameter restarts the Oracle Grid Infrastructure stack on disjoint sets of nodes so that singleton resources are relocated before Oracle Grid Infrastructure starts.

 **Note:**

If the server pool to which a resource belongs contains only one active server, then that resource will go offline as relocation cannot take place.

The `-smartmove` parameter:

- Creates a map of services and nodes on which they are running.
- Creates batches of nodes. The first batch will contain only the Hub node, if the configuration is an Oracle Flex Cluster. For additional batches, a node can be merged into a batch if:
 - * The availability of any non-singleton service, running on this node, does not go below the specified service availability factor (or the default of 50%).
 - * There is a singleton service running on this node and the batch does not contain any of the relocation target nodes for the service.
- Restarts the Oracle Grid Infrastructure stack batch by batch.
- Service availability factor (`-saf Z+`): You can specify a positive number, as a percentage, that will indicate the minimum number of database instances on which a database service must be running. For example:
 - If you specify `-saf 50` for a service running on two instances, then only one instance can go offline at a time.
 - If you specify `-saf 50` for a service running on three instances, then only one instance can go offline at a time.
 - If you specify `-saf 75` for a service running on two instances, then an error occurs because the target can never be met.
 - The service availability factor is applicable for services running on at least two instances. As such, the service availability factor can be 0% to indicate a non-rolling move, but not 100%. The default is 50%.
 - If you specify a service availability factor for singleton services, then the parameter will be ignored because the availability of such services is 100% and the services will be relocated.
- `-eval`: You can optionally use this parameter to view the auto-generated batches. This parameter also shows the sequence of the move operation without actually patching the software.

Related Topics

- [rhpctl move gihome](#)
Moves the Oracle Grid Infrastructure software stack from one home to another.

Combined Oracle Grid Infrastructure and Oracle Database Patching

When you patch an Oracle Grid Infrastructure deployment, Fleet Patching and Provisioning enables you to simultaneously patch the Oracle Database homes on the cluster, so you can patch both types of software homes in a single maintenance operation.

Note:

You cannot patch both Oracle Grid Infrastructure and Oracle Database in combination, with the independent automaton.

The following optional parameters of the `ohpctl move gihome` command are relevant to the combined Oracle Grid Infrastructure and Oracle Database patching use case:

- `-auto`: Automatically patch databases along with patching Oracle Grid Infrastructure
- `-dbhomes mapping_of_Oracle_homes`: Mapping of source and destination working copies in the following format:

```
sourcewc1=destwc1,...,source_oracle_home_path=destwcN
```

- `-dblist db_name_list`: Patch only the specified databases
- `-excludedblist db_name_list`: Patch all databases except the specified databases
- `-nodatapatch`: Indicates that `datapatch` is not be run for databases being moved

As an example, assume that a Fleet Patching and Provisioning Server with Oracle Grid Infrastructure 12c release 2 (12.2) has provisioned the following working copies on an Oracle Grid Infrastructure 12c release 1 (12.1.0.2) target cluster which includes the node `test_749`:

- `GI121WC1`: The active Grid home on the Oracle Grid Infrastructure 12c release 1 (12.1.0.2) cluster
- `GI121WC2`: A software-only Grid home on the Oracle Grid Infrastructure 12c release 1 (12.1.0.2) cluster
- `DB121WC1`: An Oracle RAC 12c release 1 (12.1.0.2.0) database home running database instances
- `DB121025WC1`: An Oracle RAC 12c release 1 (12.1.0.2.5) database home with no database instances (this is the patched home)
- `DB112WC1`: An Oracle RAC 11g release 2 (11.2.0.4.0) database home running database instances
- `DB112045WC1`: An Oracle RAC 11g release 2 (11.2.0.4.5) database home with no database instances (this is the patched home)

Further assume that you want to simultaneously move

- Oracle Grid Infrastructure from working copy `GI121WC1` to working copy `GI121WC2`

- Oracle RAC Database db1 from working copy DB121WC1 to working copy DB121025WC1
- Oracle RAC Database db2 in working copy DB112WC1 to working copy DB112045WC1

The following single command accomplishes the moves:

```
$ rhpctl move gihome -sourcewc GI121WC1 -destwc GI121WC2 -auto  
-dbhomes DB121WC1=DB121025WC1,DB112WC1=DB112045WC1 -targetnode test_749  
{authentication_option}
```

Notes:

- If you have an existing Oracle home that is not currently a working copy, then specify the Oracle home path instead of the working copy name for the source home. In the preceding example, if the Oracle home path for an existing 12.1.0.2 home is `/u01/app/prod/12.1.0.2/dbhome1`, then replace `DB121WC1=DB121025WC1` with `/u01/app/prod/12.1.0.2/dbhome1=DB121025WC1`.
- If the move operation fails at some point before completing, then you can either resolve the cause of the failure and resume the operation by rerunning the command, or you can undo the partially completed operation by issuing the following command, which restores the configuration to its initial state:

```
$ rhpctl move gihome -destwc GI121WC2 -revert  
{authentication_option}
```

In the preceding command example, the Oracle Grid Infrastructure 12c release 1 (12.1.0.2) Grid home moves from working copy GI121WC1 to working copy GI121WC2, databases running on working copy DB121WC1 move to working copy DB121025WC1, and databases running on working copy DB112WC1 move to working copy DB112045WC1.

For each node in the client cluster, RHPCTL:

1. Runs any configured pre-operation user actions for moving the Oracle Grid Infrastructure (`move gihome`).
2. Runs any configured pre-operation user actions for moving the database working copies (`move database`).
3. Stops services running on the node, and applies drain and disconnect options.
4. Performs the relevant patching operations for Oracle Clusterware and Oracle Database.
5. Runs any configured post-operation user actions for moving the database working copies (`move database`).
6. Runs any configured post-operation user actions for moving the Oracle Grid Infrastructure working copy (`move gihome`).

Related Topics

- [rhpctl move gihome](#)
Moves the Oracle Grid Infrastructure software stack from one home to another.

Zero-Downtime Oracle Grid Infrastructure Patching

Use Fleet Patching and Provisioning to patch Oracle Grid Infrastructure without bringing down Oracle RAC database instances.

Current methods of patching the Oracle Grid Infrastructure require that you bring down all Oracle RAC database instances on the node where you are patching the Oracle Grid Infrastructure home. This issue is addressed in the Grid Infrastructure layer where by the database instances can continue to run during the grid infrastructure patching.

- To enable zero-downtime Oracle Grid Infrastructure patching, use the `rhpctl move gihome` command in a manner similar to the following:

```
rhpctl move gihome -tgip
```

Patching Oracle Grid Infrastructure Using Local-Mode Configuration

When you install Oracle Grid Infrastructure or when you upgrade an older version to this current version, the Fleet Patching and Provisioning Server is configured automatically in local mode.

The local-mode operation enables you to perform grid infrastructure and database patching operations on the local cluster in a simplified environment without having to register or deploy gold images. Deploy either the grid infrastructure or the database patched home and run the patch operation using either the `rhpctl move gihome` or `rhpctl move database` command, specifying the source and destination paths instead of working copy names.

Note:

In 20c, the Fleet Patching and Provisioning Server is started automatically when patching command is invoked and the dependent resources are set up during the automatic startup process.

To switch the Fleet Patching and Provisioning Server from local mode to the regular, central mode (to manage remote targets), you must delete the current Fleet Patching and Provisioning Server in local mode, as follows:

```
$ srvctl stop rhpserver  
$ srvctl remove rhpserver
```

Proceed with the steps described in "Creating a Fleet Patching and Provisioning Server" to create the central-mode Fleet Patching and Provisioning Server.

- The independent automaton for patching Oracle Grid Infrastructure performs all of the steps necessary to switch from one home to another. Because the automaton

is not aware of gold images, moving the database requires two home paths, as follows:

```
$ rhpctl move gihome -sourcehome Oracle_home_path -destinationhome  
Oracle_home_path
```

Use the following `rhpctl move gihome` command parameters for the patching operation:

- `-node`: If the home you are moving is an Oracle Grid Infrastructure home installed on more than one node, then the default operation is a rolling update on all nodes. To apply a patch to just one node, specify the name of that node with this parameter.
- `-nonrolling`: If the home you are moving is an Oracle Grid Infrastructure home installed on more than one node, then the default operation is a rolling update on all nodes. To patch all nodes in a nonrolling manner, use this parameter instead of the `-node` parameter.
- `-ignorewcpatches`: By default, Fleet Patching and Provisioning will not perform the move operation if the destination home is missing any patches present in the source home. You can override this functionality by using this parameter, for example, to move back to a previous source home if you must undo an update.

Related Topics

- [Creating a Fleet Patching and Provisioning Server](#)
The Fleet Patching and Provisioning Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and targets.

Error Prevention and Automated Recovery Options

Fleet Patching and Provisioning has error prevention and automated recovery options to assist you during maintenance operations.

During maintenance operations, errors must be avoided whenever possible and, when they occur, you must have automated recovery paths to avoid service disruption.

Error Prevention

Many RHPCTL commands include the `-eval` parameter, which you can use to run the command and evaluate the current configuration without making any changes to determine if the command can be successfully run and how running the command will impact the configuration. Commands that you run using the `-eval` parameter run as many prerequisite checks as possible without changing the configuration. If errors are encountered, then RHPCTL reports them in the command output. After you correct any errors, you can run the command again using `-eval` to validate the corrections. Running the command successfully using `-eval` provides a high degree of confidence that running the actual command will succeed.

You can test commands with the `-eval` parameter outside of any maintenance window, so the full window is available for the maintenance procedure, itself.

Automated Recovery Options

During maintenance operations, errors can occur either in-flight (for example, partway through either an `rhpctl move database` or `rhpctl move gihome` command) or after a

successful operation (for example, after an `rhpcctl move database` command, you encounter performance or behavior issues).

In-Flight Errors

Should in-flight errors occur during move operations:

- Correct any errors that RHPCTL report and rerun the command, which will resume running at the point of failure.

If rerunning the command succeeds and the move operation has a post-operation user action associated with it, then the user action is run. If there is a pre-operation user action, however, then RHPCTL does not rerun the command.

- Run a new move command, specifying only the destination from the failed move (working copy or unmanaged home), an authentication option, if required, and use the `-revert` parameter. This will restore the configuration to its initial state.

No user actions associated with the operation are run.

- Run a new move command, specifying only the destination from the failed move (working copy or unmanaged home), an authentication option if required, and the `-abort` parameter. This leaves the configuration in its current state. Manual intervention is required at this point to place the configuration in a final state.

No user actions associated with the operation are run.

Post-Update Issues

Even after a successful move operation to a new database or Oracle Grid Infrastructure home, you still may need to undo the change and roll back to the prior home. You can do this by rerunning the command with the source and destination homes reversed. This is, effectively, a fresh move operation performed without reference to the previous move operation.

Note:

For the independent automatons, the source and destination homes are always unmanaged homes (those homes not provisioned by Fleet Patching and Provisioning). When the move operation is run on a Fleet Patching and Provisioning Server or Fleet Patching and Provisioning Client, the destination home must be a managed home that was provisioned by Fleet Patching and Provisioning.

Upgrading Oracle Grid Infrastructure Software

If you are using Fleet Patching and Provisioning, then you can use a single command to upgrade an Oracle Grid Infrastructure home.

Fleet Patching and Provisioning supports upgrades to Oracle Grid Infrastructure 12c release 1 (12.1.0.2) from 11g release 2 (11.2.0.3 and 11.2.0.4). Upgrading to Oracle Grid Infrastructure 12c release 2 (12.2.0.1) is supported from 11g release 2 (11.2.0.3 and 11.2.0.4) and 12c release 1 (12.1.0.2). The destination for the upgrade can be a working copy of a gold image already provisioned or you can choose to create the working copy as part of this operation.

As an example, assume that a target cluster is running Oracle Grid Infrastructure on an Oracle Grid Infrastructure home that was provisioned by Fleet Patching and Provisioning. This Oracle Grid Infrastructure home is 11g release 2 (11.2.0.4) and the working copy is named accordingly.

After provisioning a working copy version of Oracle Grid Infrastructure 12c release 2 (12.2.0.1) (named GIOH12201 in this example), you can upgrade to that working copy with this single command:

```
$ rhpctl upgrade gihome -sourcewc GIOH11204 -destwc GIOH12201
```

Fleet Patching and Provisioning is able to identify the cluster to upgrade based on the name of the source working copy. If the target cluster was running on an unmanaged Oracle Grid Infrastructure home, then you would specify the path of the source home rather than providing a source working copy name, and you must also specify the target cluster.

 **Note:**

You can delete the source working copy at any time after completing an upgrade. Once you delete the working copy, however, you cannot perform a rollback. Also, use the `rhpctl delete workingcopy` command (as opposed to `rm`, for example) to remove the source working copy to keep the Fleet Patching and Provisioning inventory correct.

Oracle Database Software Management

The Fleet Patching and Provisioning Server provides an efficient and secure platform for the distribution of Oracle Database Homes to targets and Fleet Patching and Provisioning Clients.

Also, Fleet Patching and Provisioning Clients have the ability to fetch database homes from the Fleet Patching and Provisioning Server.

Oracle Database homes are distributed in the form of working copies of gold images. Database instances (one or more) can then be created on the working copy.

Fleet Patching and Provisioning also has commands for managing existing databases, such as switching to a patched home or upgrading to a new database version. These are both single commands which orchestrate the numerous steps involved. Reverting to the original home is just as simple.

Provisioning a Copy of a Gold Image of a Database Home

Use the `rhpctl add workingcopy` command to provision a working copy of a database home on a Fleet Patching and Provisioning Server, Client, or target.

- Run the `rhpcctl add workingcopy` command on a Fleet Patching and Provisioning Server, similar to the following example:

```
$ rhpcctl add workingcopy -image db12c -path /u01/app/dbusr/product/  
12.2.0/db12201  
-client client_007 -oraclebase /u01/app/dbusr/ -workingcopy wc_db122_1
```

The preceding command example creates a working copy named `wc_db122_1` on all nodes of the Fleet Patching and Provisioning Client cluster named `client_007`. The gold image `db12c` is the source of the workingcopy. The directory path locations that you specify in the command must be empty.

Related Topics

- [rhpcctl add workingcopy](#)

Creating an Oracle Database on a Copy of a Gold Image

Create an Oracle Database on a working copy.

The Fleet Patching and Provisioning Server can add a database on a working copy that is on the Fleet Patching and Provisioning Server, itself, a Fleet Patching and Provisioning Client, or a non-Fleet Patching and Provisioning Client target. A Fleet Patching and Provisioning Client can create a database on a working copy that is running on the Fleet Patching and Provisioning Client, itself.

- After you create a working copy of a gold image and provision that working copy to a target, you can create an Oracle Database on the working copy using the `rhpcctl add database` command, similar to the following command example, which creates an Oracle Real Application Clusters (Oracle RAC) database called `db12201` on a working copy called `wc_db122_1`:

```
$ rhpcctl add database -workingcopy wc_db122_1 -dbname db12201 -node  
client_007_node1,client_007_node2 -dbtype RAC -datafileDestination  
DATA007_DG
```

The preceding example creates an administrator-managed Oracle RAC database on two nodes in a client cluster. The data file destination is an Oracle ASM disk group that was created prior to running the command. Additionally, you can create Oracle RAC One Node and non-cluster databases.

Note:

When you create a database using Fleet Patching and Provisioning, the feature uses random passwords for both the SYS and SYSTEM schemas in the database and you cannot retrieve these passwords. A user with the DBA or operator role must connect to the database, locally, on the node where it is running and reset the passwords to these two accounts.

Patching Oracle Database Software

To patch an Oracle database, you move the database home to a new home, which includes the patches you want to implement.

Use the `rhpcctl move database` command to move one or more database homes to a working copy of the same database release level. The databases may be running on a working copy, or on an Oracle Database home that is not managed by Fleet Patching and Provisioning.

When the move operation is initiated by a Fleet Patching and Provisioning Server or Client, the version moved to must be a working copy of a gold image. You can also perform this operation using the independent automaton in an environment where no Fleet Patching and Provisioning Server is present. In this case, the source and destination homes are not working copies of gold images, but are two installed homes that you deployed with some method other than using Fleet Patching and Provisioning.

The working copy to which you are moving the database can be at a lower patch level than the current database home. This facilitates rollback in the event that you encounter any problems after moving to the higher level patched home.

The working copy to which you are moving the database home can be at the same patch level as the original working copy. This is useful if you are moving a database home from one storage location to another, or if you wish to convert an unmanaged home to a managed home while staying at the same patch level.

Fleet Patching and Provisioning applies all patches out-of-place, minimizing the downtime necessary for maintenance. Fleet Patching and Provisioning also preserves the current configuration, enabling the rollback capability previously described. By default, Fleet Patching and Provisioning applies patches in a rolling manner, which reduces, and in many cases eliminates, service downtime. Use the `-nonrolling` option to perform patching in non-rolling mode. The database is then completely stopped on the old `ORACLE_HOME`, and then restarted to make it run from the newly patched `ORACLE_HOME`.

Note:

Part of the patching process includes applying Datapatch. When you move an Oracle Database 12c release 1 (12.1) or higher, Fleet Patching and Provisioning completes this step for you. When you move to a version previous to Oracle Database 12c release 1 (12.1), however, you must run Datapatch manually. Fleet Patching and Provisioning is Oracle Data Guard-aware, and will not apply Datapatch to Oracle Data Guard standbys.

Workflow for Database Patching

Assume that a database named `myorcldb` is running on a working copy that was created from an Oracle Database 12c release 2 (12.2) gold image named `DB122`. The typical workflow for patching an Oracle Database home is as follows:

1. Create a working copy of the Oracle Database that you want to patch, in this case `DB122`.
2. Apply the patch to the working copy you created.
3. Test and validate the patched working copy.

4. Use the `rhpcctl add image` command to create a gold image (for example, `DB122_PATCH`) from the patched working copy.

 **Note:**

The working copy you specify in the preceding command must be hosted on the Fleet Patching and Provisioning Server in Fleet Patching and Provisioning-managed storage.

5. Delete the patched working copy with the patched Oracle Database using the `rhpcctl delete workingcopy` command.

 **Note:**

Do not remove directly using the `rm` command or some other method, because this does not update the Fleet Patching and Provisioning inventory information.

6. Create a working copy from the patched gold image, (`DB122_PATCH`).
7. Move `myorclpdb` to the working copy you just created.
8. When you are confident that you will not need to roll back to the working copy on which the database was running at the beginning of the procedure, delete that working copy using the `rhpcctl delete workingcopy` command.

Patching Oracle Database Using Batches

During database patching, Fleet Patching and Provisioning can sequentially process batches of nodes, with a number of nodes in each batch being restarted in parallel. This method maximizes service availability during the patching process. You can define the batches on the command line or choose to have Fleet Patching and Provisioning generate the list of batches based on its analysis of the database services running in the cluster.

Adaptive Oracle RAC-Rolling Patching for OJVM Deployments

In a clustered environment, the default approach for applying database maintenance with Fleet Patching and Provisioning is Oracle RAC rolling. However, non-rolling may be required if the new (patched) database home contains OJVM patches. In this case, Fleet Patching and Provisioning determines whether the rolling approach is possible, and rolls when applicable. (See MOS Note 2217053.1 for details.)

Related Topics

- [Patching Oracle Grid Infrastructure Software](#)
- [Provisioning Copies of Gold Images](#)
- [Adding Gold Images to the Fleet Patching and Provisioning Server](#)
- [RHPCTL Command Reference](#)

Patching Oracle Database with the Independent Automaton

The independent local-mode automaton updates Oracle Database homes, including Oracle Database single-instance databases in a cluster or standalone (with no Oracle Grid Infrastructure), an Oracle RAC database, or an Oracle RAC One Node database.

In a clustered environment, the database source home can be Oracle Database 11g (11.2.0.4), 12c (12.1.0.2), 12c (12.2.0.1), or later.

In a standalone (non-clustered) environment, the database home must be Oracle Database 18c or later.

The automaton runs locally on the deployment and does not require a Fleet Patching and Provisioning Server in the architecture. (If there is a Fleet Patching and Provisioning Server in the architecture, then the automaton does not communicate with it, and the Fleet Patching and Provisioning Server cannot interact with the independent automaton.)

- The independent automaton for Oracle Database patching performs all of the steps necessary to switch from one home to another. Because the automaton is not aware of gold images, moving the database requires two home paths, as follows:

```
$ rhpctl move database -sourcehome Oracle_home_path -destinationhome  
Oracle_home_path
```

Use the following `rhpctl move database` command parameters for any of the patching scenarios:

- `-dbname`: If the database home is hosting more than one database, you can move specific databases by specifying a comma-delimited list with this parameter. Databases not specified are not moved. If you do not use this parameter, then RHPCTL moves all databases.

Note:

If you are moving a non-clustered (single-instance) database, then, for the value of the `-dbname` parameter, you must specify the SID of the database instead of the database name.

- `-ignorewcpatches`: By default, Fleet Patching and Provisioning will not perform the move operation if the destination home is missing any patches present in the source home. You can override this functionality by using this parameter, for example, to move back to a previous source home if you must undo an update.

The following parameters apply only to clustered environments:

- `-node`: If the home you are moving is a database home installed on more than one node, then the default operation is a rolling update on all nodes. To apply a patch to just one node, specify the name of that node with this parameter.
- `-nonrolling`: If the home you are moving is a database home installed on more than one node, then the default operation is a rolling update on all nodes. To patch all nodes in a nonrolling manner, use this parameter instead of the `-node` parameter.

- `-disconnect` and `-noreplay`: Applies to single-instance Oracle Databases, and Oracle RAC and Oracle RAC One Node database instances. Use the `-disconnect` parameter to disconnect all sessions before stopping or relocating services. If you choose to use `-disconnect`, then you can choose to use the `-noreplay` parameter to disable session replay during disconnection.

- `-drain_timeout`: Applies to single-instance Oracle Databases, and Oracle RAC, and Oracle RAC One Node database instances. Use this parameter to specify the time, in seconds, allowed for resource draining to be completed from each node. Accepted values are an empty string (""), 0, or any positive integer. The default value is an empty string, which means that this parameter is not set. This is applicable to older versions to maintain traditional behavior. If it is set to 0, then the stop option is applied immediately.

The draining period is intended for planned maintenance operations. During the draining period, on each node in succession, all current client requests are processed, but new requests are not accepted.

- `-stopoption`: Applies to single-instance Oracle Databases, and Oracle RAC, and Oracle RAC One Node database instances. Specify a stop option for the database. Stop options include: ABORT, IMMEDIATE, NORMAL, TRANSACTIONAL, and TRANSACTIONAL_LOCAL.

**Note:**

The `rhpcctl move database` command is Oracle Data Guard-aware, and will not run Datapatch if the database is an Oracle Data Guard standby.

Related Topics

- [rhpcctl move database](#)
Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.

Patching Oracle Exadata Software

In addition to Oracle Grid Infrastructure and Oracle Database homes, Fleet Patching and Provisioning supports patching the Oracle Exadata components: database nodes, storage cells, and InfiniBand switches.

The first time you patch an Oracle Exadata system using Fleet Patching and Provisioning, run the `rhpcctl add workingcopy` command, which stores the Oracle Exadata system information (list of nodes and the images with which they were last patched) on the Fleet Patching and Provisioning Server, before patching the desired Oracle Exadata nodes.

For subsequent patching, run the `rhpcctl update workingcopy` command. After patching, Fleet Patching and Provisioning updates the images of the nodes.

When you run the `rhpcctl query workingcopy` command for a working copy based on the EXAPATCHSOFTWARE image type, the command returns a list of nodes and their images.

To use Fleet Patching and Provisioning to patch Oracle Exadata:

1. Import an image of the EXAPATCHSOFTWARE image type, using the `rhpcctl import image` command, similar to the following:

```
$ rhpcctl import image -image EXA1 -imagetype EXAPATCHSOFTWARE -
path /tmp/ExadataPatchBundle
-version 12.1.2.2.3.160720
```

 **Note:**

You must rename the patch zip files to include the words *storage*, *database* and, *iso*, so that Fleet Patching and Provisioning can distinguish among them.

2. If this is the first time you are using Fleet Patching and Provisioning to patch Oracle Exadata, then use the `rhpcctl add workingcopy` command, as follows:

```
$ rhpcctl add workingcopy -image image_name -root {[-dbnodes dbnode_list]
[-cells cell_list] [-ibswitches ibswitch_list]} [-fromnode node_name]
[-unkey] [-smtpfrom "address"] [-smtpsto "addresses"] [-precheckonly]
[-modifyatprereq] [-resetforce] [-force]
```

The preceding command stores the list of nodes (database nodes, cells, and InfiniBand switches) and the version of each node in Fleet Patching and Provisioning on the working copy, in addition to the type of node and the type of image.

3. For subsequent patching operations, after you create a gold image with updated database, storage cell, and switch files, run the `rhpcctl update workingcopy` command to patch one, two, or all three Oracle Exadata components, as follows:

```
$ rhpcctl update workingcopy -image image_name -root {[-dbnodes
dbnode_list] [-cells cell_list]
[-ibswitches ibswitch_list]} [-fromnode node_name] [-unkey] [-
smtpfrom "address"]
[-smtpsto "addresses"] [-precheckonly] [-modifyatprereq] [-resetforce]
[-force]
```

 **Note:**

The name of the working copy remains the same throughout the life cycle of patching the given Oracle Exadata target.

You can choose to patch only the database nodes, cells, or InfiniBand switches or any combination of the three. Patching occurs in the following order: InfiniBand switches, cells, database nodes.

4. Display the list of nodes and their images, as follows:

```
$ rhpcctl query workingcopy -workingcopy working_copy_name
```

5. Delete the working copy, as follows:

```
rhpctl delete workingcopy -workingcopy working_copy_name
```

Related Topics

- [rhpctl import image](#)
Creates an image on the Fleet Patching and Provisioning Server.
- [rhpctl add workingcopy](#)
Creates a working copy on a client cluster.
- [rhpctl update workingcopy](#)

Upgrading Oracle Database Software

Fleet Patching and Provisioning provides two options for upgrading Oracle Database. Both options are performed with a single command.

The `rhpctl upgrade database` command performs a traditional upgrade incurring downtime. The `rhpctl zdtupgrade database` command performs an Oracle RAC or Oracle RAC One Node upgrade with minimal or no downtime.

Use the `rhpctl upgrade database` command to upgrade to Oracle Database 12c release 1 (12.1.0.2) from Oracle Database 11g release 2 (11.2.0.3 and 11.2.0.4). Upgrading to Oracle Database 12c release 2 (12.2.0.1) is supported from Oracle Database 11g release 2 (11.2.0.3 and 11.2.0.4) and Oracle Database 12c release 1 (12.1.0.2).



Note:

The version of Oracle Grid Infrastructure on which the pre-upgrade database is running must be the same version or higher than the version of the database to which you are upgrading.

The destination for the upgrade can be a working copy already provisioned, or you can choose to create the working copy of gold image as part of this operation.

The pre-upgrade database can be running on a working copy (a managed home that was provisioned by Fleet Patching and Provisioning) or on an unmanaged home. In the first case, you can roll back the upgrade process with a single RHPCTL command.



Note:

You can delete the source working copy at any time after completing an upgrade. Once you delete the working copy, however, you cannot perform a rollback. Also, use the `rhpctl delete workingcopy` command (as opposed to `rm`, for example) to remove the source working copy to keep the Fleet Patching and Provisioning inventory correct.

Related Topics

- [rhpctl upgrade database](#)

- [rhpctl zdtupgrade database](#)

Zero-Downtime Upgrade

Using Fleet Patching and Provisioning, which automates and orchestrates database upgrades, you can upgrade an Oracle RAC or Oracle RAC One Node database with no disruption in service.

The zero-downtime upgrade process is resumable, restartable, and recoverable should any errors interrupt the process. You can fix the issue then re-run the command, and Fleet Patching and Provisioning continues from the error point. Oracle also provides hooks at the beginning and end of the zero-downtime upgrade process, allowing call outs to user-defined scripts, so you can customize the process.

Note:

A multitenant container database is the only supported architecture in Oracle Database 20c. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

You can use the zero-downtime upgrade process to upgrade databases that meet the following criteria:

- **Database upgrade targets:** Oracle RAC and Oracle RAC One Node, with the following upgrade paths:
 - 11.2.0.4 to 12.1.0.2
 - 11.2.0.4 to 12.2
 - 11.2.0.4 to 18c
 - 12.1.0.2 to 12.2
 - 12.1.0.2 to 18c
 - 12.1.0.2 to 19c
 - 12.1.0.2 to 20c
 - 12.2 to 18c
 - 12.2 to 19c
 - 12.2 to 20c
 - 18c to 19c
 - 18c to 20c
 - 19c to 20c
- **Fleet Patching and Provisioning management:** The source database home can either be unmanaged (not provisioned by Fleet Patching and Provisioning service) or managed (provisioned by Fleet Patching and Provisioning service)
- **Database state:** The source database must be in archive log mode

Upgrading Container Databases

You can use Fleet Patching and Provisioning to upgrade CDBs but Fleet Patching and Provisioning does not support converting a non-CDB to a CDB during upgrade. To prepare for a zero-downtime upgrade, you complete configuration steps and validation

checks. When you run a zero-downtime upgrade using Fleet Patching and Provisioning, you can stop the upgrade and resume it, if necessary. You can recover from any upgrade errors, and you can restart the upgrade. You also have the ability to insert calls to your own scripts during the upgrade, so you can customize your upgrade procedure.

Zero-Downtime Upgrade Environment Prerequisites

- **Server environment:** Oracle Grid Infrastructure 18c with Fleet Patching and Provisioning
- **Database hosts:** Databases hosted on one of the following platforms:
 - Oracle Grid Infrastructure 18c Fleet Patching and Provisioning Client
 - Oracle Grid Infrastructure 18c Fleet Patching and Provisioning Server
 - Oracle Grid Infrastructure 12c (12.2.0.1) Fleet Patching and Provisioning Client
 - Oracle Grid Infrastructure 12c (12.1.0.2) target cluster
- **Database-specific prerequisites for the environment:** During an upgrade, Fleet Patching and Provisioning manages replication to a local data file to preserve transactions applied to the new database when it is ready. There are two possibilities for the local data file:
 - Snap clone, which is available if the database data files and redo and archive redo logs are on Oracle ACFS file systems
 - Full copy, for all other cases
- Fleet Patching and Provisioning requires either Oracle GoldenGate or Oracle Data Guard during a zero-downtime database upgrade. As part of the upgrade procedure, Fleet Patching and Provisioning configures and manages the Oracle GoldenGate deployment.

Running a Zero-Downtime Upgrade Using Oracle GoldenGate for Replication

Run a zero-downtime upgrade using Oracle GoldenGate for replication.

1. Prepare the Fleet Patching and Provisioning Server.

Create gold images of the Oracle GoldenGate software in the image library of the Fleet Patching and Provisioning Server.

Note:

You can download the Oracle GoldenGate software for your platform from Oracle eDelivery. The Oracle GoldenGate 12.3 installable kit contains the required software for both Oracle Database 11g and Oracle Database 12c databases.

If you download the Oracle GoldenGate software, then extract the software home and perform a *software only* installation on the Fleet Patching and Provisioning Server.

Create gold images of the Oracle GoldenGate software for both databases, as follows:

```
$ rhpctl import image -image 112ggimage -path path -imagetype
ORACLEGGSOFTWARE
$ rhpctl import image -image 12ggimage -path path -imagetype
ORACLEGGSOFTWARE
```

In both of the preceding commands, *path* refers to the location of the Oracle GoldenGate software home on the Fleet Patching and Provisioning Server for each release of the database.

2. Prepare the target database.

Provision working copies of the Oracle GoldenGate software to the cluster hosting the database, as follows:

```
$ rhpctl add workingcopy -workingcopy GG_Wcopy_11g -image 112ggimage -
user
  user_name -node 12102_cluster_node -path path {-root | -sudouser
user_name
  -sudopath sudo_bin_path}
$ rhpctl add workingcopy -workingcopy GG_Wcopy_12c -image 12ggimage -
user
  user_name -node 12102_cluster_node -path path {-root | -sudouser
user_name
  -sudopath sudo_bin_path}
```

If the database is hosted on the Fleet Patching and Provisioning Server, itself, then neither the `-targetnode` nor `-client` parameters are required.

Note:

Working copy names must be unique, therefore you must use a different working copy name on subsequent targets. You can create unique working copy names by including the name of the target/client cluster name in the working copy name.

3. Provision a working copy of the Oracle Database 12c software home to the target cluster.

Note:

You can do this preparation ahead of the maintenance window without disrupting any operations taking place on the target.

- You can run the upgrade command on the Fleet Patching and Provisioning Server to upgrade a database hosted on the server, an Oracle Database 12c release 1 (12.1.0.2) target cluster, or a database hosted on a Fleet Patching and Provisioning Client 12c release 2 (12.2.0.1) or 18c. You can also run the command

a Fleet Patching and Provisioning Client 18c to upgrade a database hosted on the client, itself.

Use the upgrade command similar to the following:

```
$ rhpctl zdtupgrade database -dbname sierra -destwc DB_Wcopy_121 -  
gsrcwc  
GG_Wcopy_11g -ggdstwc GG_Wcopy_12c -targetnode 12102_cluster_node -  
root
```

In the preceding command, *12102_cluster_node* refers to the Oracle Grid Infrastructure 12c release 1 (12.1.0.2) cluster hosting the database you want to upgrade.

Related Topics

- [rhpctl import image](#)
Creates an image on the Fleet Patching and Provisioning Server.
- [rhpctl add workingcopy](#)
Creates a working copy on a client cluster.
- [rhpctl zdtupgrade database](#)

Running a Zero-Downtime Upgrade Using Oracle Data Guard for Replication

Run a zero-downtime upgrade using Oracle Data Guard for replication.

You can run the zero-downtime upgrade command using Oracle Data Guard's transient logical standby (TLS) feature. All of the steps involved are orchestrated by the zero-downtime upgrade command.

After you provision the destination database Home, the following prerequisites must be met:

- Data Guard Broker is not enabled
- Flash recovery area (FRA) is configured
- The following example of a zero-downtime upgrade using Oracle Data Guard upgrades an Oracle Database 11g release 2 (11.2.0.4), *sierra*, running on the target cluster, which includes a node, *targetclust003*, to an Oracle Database 12c release 1 (12.1.0.2) (the destination working copy, which was provisioned from a Gold Image stored on the Fleet Patching and Provisioning Server named *rhps.example.com*):

```
$ rhpctl zdtupgrade database -dbname sierra -destwc WC121DB4344 -  
clonedatadg DBDATA -targetnode node90743 -root
```

```
Enter user "root" password:  
node90753.example.com: starting zero downtime upgrade operation ...  
node90753.example.com: verifying patches applied to Oracle homes ...  
node90753.example.com: verifying if database "sierra" can be upgraded  
with zero downtime ...  
node90743: 15:09:10.459: Verifying whether database "sierra" can be  
cloned ...
```

```
node90743: 15:09:10.462: Verifying that database "sierra" is a primary
database ...
node90743: 15:09:14.672: Verifying that connections can be created to
database "sierra" ...
< ... >
node90743: 15:14:58.015: Starting redo apply ...
node90743: 15:15:07.133: Configuring primary database "sierra" ...
#####
node90753.example.com: retrieving information about database
"xmvtokvd" ...
node90753.example.com: creating services for snapshot database
"xmvtokvd" ...
#####
node90743: 15:15:33.640: Macro step 1: Getting information and
validating setup ...
< ... >
node90743: 15:16:02.844: Macro step 2: Backing up user environment in
case upgrade is aborted ...
node90743: 15:16:02.848: Stopping media recovery for database
"xmvtokvd" ...
node90743: 15:16:05.858: Creating initial restore point
"NZDRU_0000_0001" ...
< ... >
node90743: 15:16:17.611: Macro step 3: Creating transient logical
standby from existing physical standby ...
node90743: 15:16:18.719: Stopping instance "xmvtokvd2" of database
"xmvtokvd" ...
node90743: 15:16:43.187: Verifying that database "sierra" is a primary
database ...
< ... >
node90743: 15:19:27.158: Macro step 4: Upgrading transient logical
standby database ...
node90743: 15:20:27.272: Disabling service "sierrasvc" of database
"xmvtokvd" ...
node90743: 16:36:54.684: Macro step 5: Validating upgraded transient
logical standby database ...
node90743: 16:37:09.576: Creating checkpoint "NZDRU_0301" for database
"xmvtokvd" during stage "3" and task "1" ...
node90743: 16:37:09.579: Stopping media recovery for database
"xmvtokvd" ...
node90743: 16:37:10.792: Creating restore point "NZDRU_0301" for
database "xmvtokvd" ...
node90743: 16:37:11.998: Macro step 6: Switching role of transient
logical standby database ...
node90743: 16:37:12.002: Verifying that database "sierra" is a primary
database ...
< ... >
node90743: 16:39:07.425: Macro step 7: Flashback former primary
database to pre-upgrade restore point and convert to physical
standby ...
node90743: 16:39:08.833: Stopping instance "sierra2" of database
"sierra" ...
< ... >
node90743: 16:41:17.138: Macro step 8: Recovering former primary
database ...
```

```

node90743: 16:41:19.045: Verifying that database "sierra" is mounted ...
< ... >
node90743: 17:20:21.378: Macro step 9: Switching back ...
< ... >
#####
node90753.example.com: deleting snapshot database "xmvotkvd" ...

```

Customizing Zero-Downtime Upgrades

You can customize zero-downtime upgrades using the user-action framework of Fleet Patching and Provisioning.

To use the user-action framework, you can provide a separate script for any or all of the points listed in the overall process.

Table 5-3 Zero-Downtime Upgrade Plugins

Plugin Type	Pre or Post	Plugin runs...
ZDTUPGRADE_DATABASE	Pre	Before Fleet Patching and Provisioning starts zero-downtime upgrade.
	Post	After Fleet Patching and Provisioning completes zero-downtime upgrade.
ZDTUPGRADE_DATABASE_SNAPDB	Pre	Before creating the snapshot or full-clone database.
	Post	After starting the snapshot or full-clone database (but before switching over).
ZDTUPGRADE_DATABASE_DBUA	Pre	Before running DBUA (after switching over).
	Post	After DBUA completes.
ZDTUPGRADE_DATABASE_SWITCHBACK	Pre	Before switching back users to the upgraded source database.
	Post	After switching back users to the upgraded source database (before deleting snapshot or full-clone database).

- To register a plugin to be run during a zero-downtime upgrade, run the following command:

```

$ rhpctl add useraction -useraction user_action_name -actionscript
script_name
  {-pre | -post} -optype {ZDTUPGRADE_DATABASE |
ZDTUPGRADE_DATABASE_SNAPDB |
ZDTUPGRADE_DATABASE_DBUA | ZDTUPGRADE_DATABASE_SWITCHBACK}

```

You can specify run-time input to the plugins using the `-useractiondata` option of the `rhpctl zdtupgrade database` command.

Persistent Home Path During Patching

Oracle recommends out-of-place patching when applying updates.

Out-of-place patching involves deploying the patched environment in a new directory path and then switching the software home to the new path. This approach allows for non-disruptive software distribution because the existing home remains active while the new home is provisioned, and also facilitates rollback because the unpatched software home is available should any issues arise after the switch. Additionally, out-of-place patching for databases enables you to choose to move a subset of instances to the new home if more than one instance is running on the home, whereas with in-place patching, you must patch all instances at the same time.

A potential impediment to traditional out-of-place patching is that the software home path changes. While Fleet Patching and Provisioning manages this internally and transparently for Oracle Database and Oracle Grid Infrastructure software, some users have developed scripts which depend on the path. To address this, Fleet Patching and Provisioning uses a file system feature that enables separation of gold image software from the site-specific configuration changes, so the software home path is persistent throughout updates.

This feature is available with Oracle Database 12c release 2 (12.2) and Oracle Grid Infrastructure 12c release 2 (12.2) working copies provisioned in local storage. Also, if you provision an Oracle Database 12c release 2 (12.2) or an Oracle Grid Infrastructure 12c release 2 (12.2) home without using this feature, then, during a patching operation using either the `rhpcctl move database` or `rhpcctl move gihome` command, you can convert to this configuration and take advantage of the feature.

 **Note:**

You can only patch Oracle Grid Infrastructure on a Fleet Patching and Provisioning Client with a home that is based on a persistent home path from a Fleet Patching and Provisioning Server.

Managing Fleet Patching and Provisioning Clients

Management tasks for Fleet Patching and Provisioning Clients include creation, enabling and disabling, creating users and assigning roles to those users, and managing passwords.

Using SRVCTL and RHPCTL, you can perform all management tasks for a Fleet Patching and Provisioning Client.

Creating a Fleet Patching and Provisioning Client

Users operate on a Fleet Patching and Provisioning Client to perform tasks such as requesting deployment of Oracle homes and querying gold images.

To create a Fleet Patching and Provisioning Client:

1. If there is no highly available VIP (HAVIP) on the Fleet Patching and Provisioning Server, then, as the `root` user, create an HAVIP, as follows:

```
# srvctl add havip -id id -address {host_name | ip_address}
```

You can specify either a host name or IPv4 or IPv6 IP address. The IP address that you specify for HAVIP or the address that is resolved from the specified host name *must not* be in use when you run this command.

 **Note:**

The highly available VIP must be in the same subnet as the default network configured in the Fleet Patching and Provisioning Server cluster. You can obtain the subnet by running the following command:

```
$ srvctl config network -netnum network_number
```

2. On the Fleet Patching and Provisioning Server as the Grid home owner, create the client data file, as follows:

```
$ rhpctl add client -client client_cluster_name -toclientdata path
```

RHPCTL creates the client data file in the directory path you specify after the `-toclientdata` flag. The name of the client data file is *client_cluster_name.xml*.

 **Note:**

The *client_cluster_name* must be unique and it must match the cluster name of the client cluster where you run step 4.

3. Copy the client data file that you created in the previous step to a directory on the client cluster that has read/write permissions to the Grid home owner on the Fleet Patching and Provisioning Client.
4. Create the Fleet Patching and Provisioning Client by running the following command as `root` on the client cluster:

```
# srvctl add rhpclient -clientdata path_to_client_data  
[-diskgroup disk_group_name -storage base_path]
```

If you want to provision working copies to Oracle ACFS storage on this cluster, and you have already created a disk group for this purpose, then specify this disk group in the preceding command. In this case, also specify a storage path which will be used as a base path for all mount points when creating Oracle ACFS file systems for storing working copies.

 **Note:**

Once you configure a disk group on a Fleet Patching and Provisioning Client, you cannot remove it from or change it in the Fleet Patching and Provisioning Client configuration. The only way you can do either (change or remove) is to completely remove the Fleet Patching and Provisioning Client using the `srvctl remove client` command, and then add it back with a different disk group, if necessary. Before you remove a Fleet Patching and Provisioning Client, ensure that you remove all registered users from this cluster and all working copies provisioned on this cluster.

5. Start the Fleet Patching and Provisioning Client, as follows:

```
$ srvctl start rhpclient
```

6. Check the status of the Fleet Patching and Provisioning Client, as follows:

```
$ srvctl status rhpclient
```

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*
- [Fleet Patching and Provisioning Control \(RHPCTL\) Command Reference](#)
Use the Fleet Patching and Provisioning Control (RHPCTL) utility to manage Fleet Patching and Provisioning in your cluster.

Enabling and Disabling Fleet Patching and Provisioning Clients

On the Fleet Patching and Provisioning Server, you can enable or disable a Fleet Patching and Provisioning Client.

Fleet Patching and Provisioning Clients communicate with the Fleet Patching and Provisioning Server for all actions. You cannot run any RHPCTL commands without a connection to a Fleet Patching and Provisioning Server.

To enable or disable a Fleet Patching and Provisioning Client, run the following command from the Fleet Patching and Provisioning Server cluster:

```
$ rhpctl modify client -client client_name -enabled TRUE | FALSE
```

To enable a Fleet Patching and Provisioning Client, specify `-enabled TRUE`. Conversely, specify `-enabled FALSE` to disable the client. When you disable a Fleet Patching and Provisioning Client cluster, all RHPCTL commands from that client cluster will be rejected by the Fleet Patching and Provisioning Server, unless and until you re-enable the client.

**Note:**

Disabling a Fleet Patching and Provisioning Client cluster does not disable any existing working copies on the client cluster. The working copies will continue to function and any databases in those working copies will continue to run.

Deleting a Fleet Patching and Provisioning Client

Use the following procedure to delete a Fleet Patching and Provisioning Client.

1. Before deleting the Fleet Patching and Provisioning Client, you must first delete the working copies and users on the Fleet Patching and Provisioning Server, as follows:
 - a. Query the list of working copies that have been provisioned on the Fleet Patching and Provisioning Client cluster.
Run the following command:

```
$ rhpctl query workingcopy -client client_name
```
 - b. Delete each of the working copies listed in the output of the preceding command.
Run the following command for each working copy and specify the name of the working copy you want to delete:

```
$ rhpctl delete workingcopy -workingcopy working_copy_name
```
 - c. Query the list of users from the Fleet Patching and Provisioning Client cluster.
Run the following command:

```
$ rhpctl query user -client client_name
```
 - d. Delete the users listed in the output of the preceding command, as follows:
Run the following command and specify the name of the user you want to delete and the name of the client:

```
$ rhpctl delete user -user user_name -client client_name
```
2. On the Fleet Patching and Provisioning Client cluster, delete the client, as follows:
 - a. Stop the Fleet Patching and Provisioning Client daemon.
Run the following command:

```
$ srvctl stop rhpclient
```
 - b. Delete the Fleet Patching and Provisioning Client configuration.

Run the following command:

```
$ srvctl remove rhpclient
```

3. Delete the client site configuration on the Fleet Patching and Provisioning Server cluster.

Run the following command and specify the name of the client:

```
$ rhpctl delete client -client client_name
```

Creating Users and Assigning Roles for Fleet Patching and Provisioning Client Cluster Users

When you create a Fleet Patching and Provisioning Client with the `rhpctl add client` command, you can use the `-maproles` parameter to create users and assign roles to them. You can associate multiple users with roles, or you can assign a single user multiple roles with this command.

After the client has been created, you can add and remove roles for users using the `rhpctl grant role` command and the `rhpctl revoke role`, respectively.

Managing the Fleet Patching and Provisioning Client Password

The Fleet Patching and Provisioning Client uses a password stored internally to authenticate itself with the RHP server. You cannot query this password, however, if for some reason, you are required to reset this password, then you can do so, as follows, on the RHP server cluster:

1. Run the following command on the Fleet Patching and Provisioning Server cluster to generate a new password and store it in the client credential:

```
$ rhpctl modify client -client client_name -password
```

2. Run the following command on the Fleet Patching and Provisioning Server cluster to generate a credential file:

```
$ rhpctl export client -client client_name -clientdata file_path
```

For example, to generate a credential file for a Fleet Patching and Provisioning Client named `mjk9394`:

```
$ rhpctl export client -client mjk9394 -clientdata /tmp/mjk9394.xml
```

3. Continuing with the preceding example, transport the generated credential file securely to the Fleet Patching and Provisioning Client cluster and then run the following command on any node in the Fleet Patching and Provisioning Client cluster:

```
$ srvctl modify rhpclient -clientdata path_to_mjk9394.xml
```

4. Restart the Fleet Patching and Provisioning Client daemon by running the following commands on the Fleet Patching and Provisioning Client cluster:

```
$ srvctl stop rhpclient  
$ srvctl start rhpclient
```


User-Defined Actions

You can create actions for various Fleet Patching and Provisioning operations, such as import image, add and delete working copy, and add, delete, move, and upgrade a software home.

You can create actions for various Fleet Patching and Provisioning operations, such as import image, add and delete working copy of a gold image, and add, delete, move, and upgrade a software home. You can define different actions for each operation, which can be further differentiated by the type of image to which the operation applies. User-defined actions can be run before or after a given operation, and are run on the deployment on which the operation is run, whether it be a Fleet Patching and Provisioning Server, a Fleet Patching and Provisioning Client (12c release 2 (12.2), or later), or a target that is not running a Fleet Patching and Provisioning Client.

User-defined actions are shell scripts which are stored on the Fleet Patching and Provisioning Server. When a script runs, it is given relevant information about the operation on the command line. Also, you can associate a file with the script. The Fleet Patching and Provisioning Server will copy that file to the same location on the Client or target where the script is run.

For example, perhaps you want to create user-defined actions that are run after a database upgrade, and you want to define different actions for Oracle Database 11g and 12c. This requires you to define new image types, as in the following example procedure.

1. Create a new image type, (`DB11IMAGE`, for example), based on the `ORACLEDBSOFTWARE` image type, as follows:

```
$ rhpctl add imagetype -imagetype DB11IMAGE -basetype ORACLEDBSOFTWARE
```

When you add or import an Oracle Database 11g gold image, you specify the image type as `DB11IMAGE`.

2. Define a user action and associate it with the `DB11IMAGE` image type and the upgrade operation. You can have different actions that are run before or after upgrade.
3. To define an action for Oracle Database 12c, create a new image type (`DB12IMAGE`, for example) that is based on the `ORACLEDBSOFTWARE` image type, as in the preceding step, but with the `DB12IMAGE` image type.

Note:

If you define user actions for the base type of a user-defined image type (in this case the base type is `ORACLEDBSOFTWARE`), then Fleet Patching and Provisioning performs those actions before the actions for the user-defined image type.

You can modify the image type of an image using the `rhpctl modify image` command. Additionally, you can modify, add, and delete other actions. The following two tables, [Table 5-4](#) and [Table 5-5](#), list the operations you can customize and the parameters you can use to define those operations, respectively.

Table 5-4 Fleet Patching and Provisioning User-Defined Operations

Operation	Parameter List
IMPORT_IMAGE	RHP_OPTYPE, RHP_PHASE, RHP_PATH, RHP_PATHOWNER, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ADD_WORKINGCOPY	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_PATH, RHP_STORAGETYPE, RHP_USER, RHP_NODES, RHP_ORACLEBASE, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ADD_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
DELETE_WORKINGCOPY	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
DELETE_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
MOVE_GIHOME	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
MOVE_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_DATAPATCH, RHP_USERACTIONDATA
UPGRADE_GIHOME	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA

Table 5-4 (Cont.) Fleet Patching and Provisioning User-Defined Operations

Operation	Parameter List
UPGRADE_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ADDNODE_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
DELETENODE_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ADDNODE_GIHOME	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
DELETENODE_GIHOME	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ADDNODE_WORKINGCOPY	RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ZDTUPGRADE_DATABASE	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ZDTUPGRADE_DATABASE_SNAPDB	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA
ZDTUPGRADE_DATABASE_DBUA	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA

Table 5-4 (Cont.) Fleet Patching and Provisioning User-Defined Operations

Operation	Parameter List
ZDTUPGRADE_DATABASE_ SWITCHBACK	RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA

Table 5-5 User-Defined Operations Parameters

Parameter	Description
RHP_OPTYPE	The operation type for which the user action is being executed, as listed in the previous table.
RHP_PHASE	This parameter indicates whether the user action is executed before or after the operation (is either PRE or POST).
RHP_SOURCEWC	The source working copy name for a patch of upgrade operation.
RHP_SOURCEPATH	The path of the source working copy home.
RHP_DESTINATIONWC	The destination working copy name for a patch or upgrade operation.
RHP_DESTINATIONPATH	The path of the destination working copy home.
RHP_SRCGGWC	The name of the version of the Oracle GoldenGate working copy from which you want to upgrade.
RHP_SRCGGPATH	The absolute path of the version of the Oracle GoldenGate software home from which you want to upgrade.
RHP_DESTGGWC	The name of the version of the Oracle GoldenGate working copy to which you want to upgrade.
RHP_DESTGGPATH	The absolute path of the version of the Oracle GoldenGate software home to which you want to upgrade.
RHP_PATH	This is the path to the location of the software home. This parameter represents the path on the local node from where the RHPCTL command is being run for an <code>IMPORT_IMAGE</code> operation. For all other operations, this path is present on the site where the operation is taking place.
RHP_PATHOWNER	The owner of the path for the gold image that is being imported.
RHP_PROGRESSLISTENERHOST	The host on which the progress listener is listening. You can use this parameter, together with a progress listener port, to create a TCP connection to print output to the console on which the RHPCTL command is being run.
RHP_PROGRESSLISTENERPORT	The port on which the progress listener host is listening. You can use this parameter, together with a progress listener host name, to create a TCP connection to print output to the console on which the RHPCTL command is being run.

Table 5-5 (Cont.) User-Defined Operations Parameters

Parameter	Description
RHP_IMAGE	The image associated with the operation. In the case of a move operation, it will reflect the name of the destination image.
RHP_IMAGETYPE	The image type of the image associated with the operation. In the case of a move operation, it will reflect the name of the destination image.
RHP_VERSION	The version of the Oracle Grid Infrastructure software running on the Fleet Patching and Provisioning Server.
RHP_CLI	The exact command that was run to invoke the operation.
RHP_STORAGE_TYPE	The type of storage for the home (either LOCAL or RHP_MANAGED)
RHP_USER	The user for whom the operation is being performed.
RHP_NODES	The nodes on which a database will be created.
RHP_ORACLEBASE	The Oracle base location for the provisioned home.
RHP_DBNAME	The name of the database to be created.
RHP_CLIENT	The name of the client cluster.
RHP_DATAPATCH	This parameter is set to TRUE at the conclusion of the user action on the node where the SQL patch will be run after the move database operation is complete.
RHP_USERACTIONDATA	This parameter is present in all of the operations and is used to pass user-defined items to the user action as an argument during runtime.

Example of User-Defined Action

Suppose there is an image type, APACHESW, to use for provisioning and managing Apache deployments. Suppose, too, that there is a Gold Image of Apache named apacheinstall. The following example shows how to create a user action that will run prior to provisioning any copy of our Apache Gold Image.

The following is a sample user action script named addapache_useraction.sh:

```
$ cat /scratch/apacheadmin/addapache_useraction.sh
#!/bin/sh

#refer to arguments using argument names
touch /tmp/SAMPLEOUT.txt;
for i in "$@"
do
    export $i
done

echo "OPTYPE = $RHP_OPTYPE" >> /tmp/SAMPLEOUT.txt;
echo "PHASE = $RHP_PHASE" >> /tmp/SAMPLEOUT.txt;
echo "WORKINGCOPY = $RHP_WORKINGCOPY" >> /tmp/SAMPLEOUT.txt;
echo "PATH = $RHP_PATH" >> /tmp/SAMPLEOUT.txt;
echo "STORAGE_TYPE = $RHP_STORAGE_TYPE" >> /tmp/SAMPLEOUT.txt;
```

```
echo "USER = $RHP_USER" >> /tmp/SAMPLEOUT.txt;
echo "NODES = $RHP_NODES" >> /tmp/SAMPLEOUT.txt;
echo "ORACLEBASE = $RHP_ORACLEBASE" >> /tmp/SAMPLEOUT.txt;
echo "DBNAME = $RHP_DBNAME" >> /tmp/SAMPLEOUT.txt;
echo "PROGRESSLISTENERHOST = $RHP_PROGRESSLISTENERHOST" >> /tmp/
SAMPLEOUT.txt;
echo "PROGRESSLISTENERPORT = $RHP_PROGRESSLISTENERPORT" >> /tmp/
SAMPLEOUT.txt;
echo "IMAGE = $RHP_IMAGE" >> /tmp/SAMPLEOUT.txt;
echo "IMAGETYPE = $RHP_IMAGETYPE" >> /tmp/SAMPLEOUT.txt;
echo "RHPVERSION = $RHP_VERSION" >> /tmp/SAMPLEOUT.txt;
echo "CLI = $RHP_CLI" >> /tmp/SAMPLEOUT.txt;
echo "USERACTIONDATA = $RHP_USERACTIONDATA" >> /tmp/SAMPLEOUT.txt;
$
```

The script is registered to run at the start of `rhpcctl add workingcopy` commands. The `add working copy` operation aborts if the script fails.

The following command creates a user action called `addapachepre`:

```
$ rhpcctl add useraction -optype ADD_WORKINGCOPY -pre -onerror ABORT -
useraction
  addapachepre -actionscrip /scratch/apacheadmin/addapache_useraction.sh
  -runscope ONENODE
```

The following command registers the user action for the `APACHESW` image type:

```
$ rhpcctl modify imagetype -imagetype APACHESW -useractions addapachepre
```

The registered user action is invoked automatically at the start of commands that deploy a working copy of any image of the `APACHESW` type, such as the following:

```
$ rhpcctl add workingcopy -workingcopy apachecopy001 -image apacheinstall
  -path /scratch/apacheadmin/apacheinstallloc -sudouser apacheadmin -
sudopath
  /usr/local/bin/sudo -node targetnode003 -user apacheadmin -
useractiondata "sample"
```

The sample script creates the `/tmp/SAMPLEOUT.txt` output file. Based on the example command, the output file contains:

```
$ cat /tmp/SAMPLEOUT.txt
OPTYPE = ADD_WORKINGCOPY
PHASE = PRE
WORKINGCOPY = apachecopy001
PATH = /scratch/apacheadmin/apacheinstallloc
STORAGETYPE =
USER = apacheadmin
NODES = targetnode003
ORACLEBASE =
DBNAME =
PROGRESSLISTENERHOST = mds11042003.my.company.com
PROGRESSLISTENERPORT = 58068
```

```
IMAGE = apacheinstall
IMAGETYPE = APACHESW
RHPVERSION = 12.2.0.1.0
CLI = rhpctl__add__workingcopy__-image__apacheinstall__-path__/_scratch/
apacheadmin
/apacheinstallloc__-node__targetnode003__-useractiondata__sample__
-sudopath__/_usr/_local/_bin/_sudo__-workingcopy__apachecopy__-
user__apacheadmin__
-sudouser__apacheadmin__USERACTIONDATA = sample
$
```

 **Notes:**

- In the preceding output example empty values terminate with an equals sign (=).
- The spaces in the command-line value of the `RHP_CLI` parameter are replaced by two underscore characters (`__`) to differentiate this from other parameters.

Job Scheduler for Operations

The Fleet Patching and Provisioning job scheduler provides you with a mechanism to submit operations at a scheduled time instead of running the command immediately, querying the metadata of the job, and then deleting the job from the repository.

The Fleet Patching and Provisioning job scheduler includes the following features:

- Enables you to schedule a command to run at a specific point in time by providing the time value
- Performs the job and stores the metadata for the job, along with the current status of the job
- Stores the logs for each of the jobs that have run or are running
- Enables you to query job details (for all jobs or for specific jobs, based on the user roles)
- Deletes jobs
- Authorizes the running, querying, and deleting of jobs, based on role-based access for users

Use the `-schedule timer_value` command parameter with any of the following RHPCTL commands to schedule certain Rapid Hope Provisioning operations:

- `rhpctl add workingcopy`
- `rhpctl import image`
- `rhpctl delete image`
- `rhpctl add database`
- `rhpctl move gihome`
- `rhpctl move database`

- `rhpcctl upgrade database`
- `rhpcctl addnode database`
- `rhpcctl deletenode database`
- `rhpcctl delete workingcopy`

For example:

```
$ rhpcctl add workingcopy -workingcopy 18_3 -image 18_3_Base -  
oraclebase /u01/app/oracle -schedule 2016-12-21T19:13:17+05
```

All commands are run in reference with the time zone of the server, according to the ISO-8601 value, and RHPCTL displays the command result by specifying the same time zone.

Command Results

RHPCTL stores any command that is run from the command queue on the Fleet Patching and Provisioning Server. When you query a command result by specifying the command identifier, then RHPCTL returns the path to the job output file, along with the results.

Job Operation

When you run an RHPCTL command with the `-schedule` parameter, the operation creates a job with a unique job ID that you can query to obtain the status of the job.

Job Status

At any point in time, a job can be in any of the following states:

EXECUTED | TIMER_RUNNING | EXECUTING | UNKNOWN | TERMINATED

- **EXECUTED:** The job is complete.
- **TIMER_RUNNING:** The timer for the job is still running.
- **EXECUTING:** The timer for the job has expired and is running.
- **UNKNOWN:** There is an unexpected failure due to issues such as a target going down, nodes going down, or any resource failures.
- **TERMINATED:** There is an abrupt failure or the operation has stopped.

Related Topics

- [rhpcctl delete job](#)
- [rhpcctl query job](#)

Oracle Restart Patching and Upgrading

You can use Fleet Patching and Provisioning to patch and upgrade Oracle Restart using gold images.

You can move the target of single-node Oracle Restart to an Oracle home that you provision from a gold image that includes any patches. Fleet Patching and

Provisioning ensures the copying of the configuration files, such as listener.ora, to the new Oracle home.

You can also use Fleet Patching and Provisioning to upgrade Oracle Restart using gold images. Upgrade the Oracle Restart environment by upgrading the Oracle home on the target Oracle home that you provision from a higher-level gold image. Fleet Patching and Provisioning updates the configuration and inventory settings.

Use the RHPCTL utility similar to the following:

```
rhpcctl add workingcopy -workingcopy Oracle_Restart_working_copy -  
resposnefile Oracle_Restart_resposne_file -targetnode  
node_on_which_Oracle_Restart_is_provisioned {superuser credentials}
```

Fleet Patching and Provisioning Use Cases

Review these topics for step-by-step procedures to provision, patch, and upgrade your software using Fleet Patching and Provisioning.

Fleet Patching and Provisioning is a software lifecycle management solution and helps standardize patching, provisioning, and upgrade of your standard operating environment.

Creating an Oracle Grid Infrastructure 12c Release 2 Deployment

Provision Oracle Grid Infrastructure software on two nodes that do not currently have a Grid home, and then configure Oracle Grid Infrastructure to form a multi-node Oracle Grid Infrastructure installation.

Before You Begin

Note:

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Members Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

Note:

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

Provide configuration details for storage, network, users and groups, and node information for installing Oracle Grid Infrastructure in a response file. You can store the response file in any location on the Fleet Patching and Provisioning Server.

You can provision an Oracle Standalone Cluster, Oracle Application Clusters, Oracle Domain Services Cluster, or Oracle Member Clusters. Ensure that the response file has the required cluster configuration details.

Ensure that you have storage, network, and operating system requirements configured as stated in the *Oracle Grid Infrastructure Installation Guide*.

Procedure

- From the Fleet Patching and Provisioning Server, run the command:

```
$ rhpctl add workingcopy -workingcopy GI122 -image GI_HOME_12201 -  
responsefile /u01/app/rhpinfo/GI_12201_install.rsp  
{authentication_option}
```

GI122 is the working copy based on the image GI_HOME_12201

/u01/app/rhpinfo/GI_12201_install.rsp is the response file location.

Cluster Verification Utility checks for preinstallation configuration as per requirements. Fleet Patching and Provisioning configures Oracle Grid Infrastructure.

Oracle Grid Infrastructure 12c release 2 is provisioned as per the settings in the same response file.

During provisioning, if an error occurs, the procedure stops and allows you to fix the error. After fixing the error, you can resume the provisioning operation from where it last stopped.

Watch a video  [Video](#)

Provisioning an Oracle Database Home and Creating a Database

This procedure provisions Oracle Database 12c release 2 (12.2) software and creates Oracle Database instances.

Procedure

1. From the Fleet Patching and Provisioning Server, provision the Oracle Database home software:

```
$ rhpctl add workingcopy -image db12201 -path /u01/app/dbusr/product/  
12.2.0/db12201  
-client client_001 -oraclebase /u01/app/dbusr/ -workingcopy db122
```

The command provisions the working copy db122 to the specified path on the cluster client_001, from the image db12201 .

2. Create the database instance:

```
$ rhpctl add database -workingcopy db122 -dbname db -dbtype RAC
```

The command creates an Oracle RAC database instance `db`. You can use the `add database` command repeatedly to create more instances on the working copy.

Watch a video  [Video](#)

Provisioning a Pluggable Database

You can provision a pluggable database (PDB) on an existing container database (CDB) running in a provisioned database working copy.

After you create a working copy of a gold image, provision that working copy to a target, and create a database as a multitenant CDB, you can add a PDB to the CDB using the `rhpcctl addpdb database` command.

- The following command example creates a PDB called `pdb183` on a CDB called `raccdb183`, which is on a working copy called `wc_db183`:

```
$ rhpcctl addpdb database -workingcopy wc_db183 -cdbName raccdb183 -  
pdbName pdb183
```

- Use the `rhpcctl deletepdb database` command to delete a PDB from an existing CDB on a working copy.

The following command example deletes a PDB called `pdb183` on a CDB called `raccdb183`, which is on a working copy called `wc_db183`:

```
$ rhpcctl deletepdb database -workingcopy wc_db183 -cdbName raccdb183 -  
pdbName pdb183
```

Upgrading to Oracle Grid Infrastructure 12c Release 2

This procedure uses Fleet Patching and Provisioning to upgrade your Oracle Grid Infrastructure cluster from 11g release 2 (11.2.0.4) to 12c release 2 (12.2).

Before You Begin

To upgrade to Oracle Grid Infrastructure 12c release 2 (12.2.0.1), your source must be Oracle Grid Infrastructure 11g release 2 (11.2.0.3 or 11.2.0.4), or Oracle Grid Infrastructure 12c release 2 (12.1.0.2).

Ensure that groups configured in the source home match those in the destination home.

Ensure that you have an image `GI_HOME_12201` of the Oracle Grid Infrastructure 12c release 2 (12.2.0.1) software to provision your working copy.

`GI_11204` is the active Grid Infrastructure home on the cluster being upgraded. It is a working copy because in this example, Fleet Patching and Provisioning provisioned the cluster. Fleet Patching and Provisioning can also upgrade clusters whose Grid Infrastructure homes are unmanaged that is, homes that Fleet Patching and Provisioning did not provision.

Procedure

1. Provision a working copy of the Oracle Grid Infrastructure 12c release 2 (12.2.0.1) software:

```
$ rhpctl add workingcopy -workingcopy GI122 -image GI_HOME_12201  
{authentication_option}
```

GI122 is the working copy based on the image GI_HOME_12201.

2. Upgrade your target cluster to the GI122 working copy:

```
rhpctl upgrade gihome -sourcewc GI1204 -destwc GI122
```

Rapid Home Provisioning identifies the cluster to upgrade based on the name of the source working copy, and upgrades to the working copy GI122.

Patching Oracle Grid Infrastructure Without Changing the Grid Home Path

This procedure explains how to patch Oracle Grid Infrastructure without changing the Grid home path.

Before You Begin

- Ensure that the gold image containing the Grid home is imported and exists on the Fleet Patching and Provisioning Server.
- Ensure that the directory you provide in the `path` option is not an existing directory.
- The source Grid home must be a managed home (provisioned by Fleet Patching and Provisioning). It does not need to be an Oracle Layered File System (OLFS)-compliant home.
- The Grid home must be Oracle Grid Infrastructure 12c (12.2.0.1) or later.

Procedure for Patching

- To move a non-OLFS-compliant Grid home to an OLFS Grid home, from the Fleet Patching and Provisioning Server, run two commands similar to the following:

```
$ rhpctl add workingcopy -workingcopy GI_HOME_12201_PSU1 -aupath  
/u01/app/orabase/product/12.2.0.1/aupath -image image_name  
-client client_name -softwareonly
```

```
$ rhpctl move gihome -srcwc GI_HOME_12201 -destwc GI_HOME_12201_PSU1 -  
agpath  
/u01/app/orabase/product/12.2.0.1/agpath -path  
/u01/app/orabase/product/12.2.0.1/grid
```

- To move an OLFS-compliant Grid home to a patched version, run two commands similar to the following:

```
$ rhpctl add workingcopy -workingcopy gihome12201_psu1_lpm -aupath  
/u01/app/orabase/product/12.2.0.1/aupath -image image_name -client  
client_name -softwareonly
```

```
$ rhpctl move gihome -sourcewc gihome12201_lpm -destwc  
gihome12201_psu1_lpm
```

Patching Oracle Grid Infrastructure and Oracle Databases Simultaneously

This procedure patches Oracle Grid Infrastructure and Oracle Databases on the cluster to the latest patch level without cluster downtime.

Before You Begin

In this procedure, Oracle Grid Infrastructure 12c release 2 (12.2.0.1) is running on the target cluster. Working copy `GI_HOME_12201_WCPY` is the active Grid home on this cluster. Working copy `DB_HOME_12201_WCPY` runs an Oracle RAC 12c release 2 (12.2.0.1) Database with running database instance `db1`. Working copy `DB_HOME_12102_WCPY` runs an Oracle RAC 12c release 1 (12.1.0.2) Database with running database instance `db2`.

Ensure that you have images `GI_HOME_12201_PSU1`, `DB_HOME_12201_PSU1`, `DB_HOME_12102_PSU5` with the required patches for Oracle Grid Infrastructure and Oracle RAC Database on the Fleet Patching and Provisioning Server.

The groups configured in the source home must match with those in the destination home.

Procedure

1. Prepare target Oracle homes as follows:

- a. Provision software-only Grid home on the cluster to be patched:

```
$ rhpctl add workingcopy -workingcopy GI_HOME_12201_PATCHED_WCPY  
-image GI_HOME_12201_PSU1 -client CLUSTER_005 -softwareonly
```

- b. Provision each release Database home, without database instances, to be patched:

```
$ rhpctl add workingcopy -workingcopy DB_HOME_12201_PATCHED_WCPY  
-image DB_HOME_12201_PSU1  
$ rhpctl add workingcopy -workingcopy DB_HOME_12102_PATCHED_WCPY  
-image DB_HOME_12102_PSU5
```

2. Patch Oracle Grid Infrastructure and all Oracle RAC Databases on node1 as follows:

```
$ rhpctl move gihome -sourcewc GI_HOME_12201_WCPY -destwc  
GI_HOME_12201_PATCHED_WCPY -auto  
-dbhomes  
DB_HOME_12102_WCPY=DB_HOME_12102_PATCHED_WCPY,DB_HOME_12201_WCPY=DB_HOME  
_12201_PATCHED_WCPY -targetnode node1 {authentication_option}
```

When you run the command, you move your active Oracle Grid Infrastructure from working copy GI_HOME_12201_WCPY to GI_HOME_12201_PATCHED_WCPY, Oracle RAC Database db1 from DB_HOME_12201_WCPY to DB_HOME_12201_PATCHED_WCPY, and Oracle RAC Database db2 from DB_HOME_12102_WCPY to DB_HOME_12102_PATCHED_WCPY.

Patching Oracle Database 12c Release 1 Without Downtime

This procedure explains how to patch Oracle Database 12c release 1 (12.1.0.2) with the latest patching without bringing down the database.

Before You Begin

You have an Oracle Database db12102 that you want to patch to the latest patch level.

Ensure that the working copy db12102_psu based on the image DB12102_PSU contains the latest patches and is available.

Procedure

From the Fleet Patching and Provisioning Server, run one of the following commands as per your source and destination database:

1. To patch an Oracle Database home managed by Fleet Patching and Provisioning, and there exist working copies of the source and destination databases, run:

```
rhpctl move database -sourcewc db12102 -patchedwc db12102_psu
```

db12102 is the source working copy of the database being patched

db12102_psu is the working copy of the Oracle Database software with patches applied, based on the image DB12102_PSU.

2. To patch an unmanaged Oracle Database home (source working copy does not exist because the Oracle home is not managed by Fleet Patching and Provisioning), run:

```
rhpctl move database -sourcehome /u01/app/orabase/product/12.1.0.2/  
dbhome_1  
-patchedwc db12102_psu -targetnode node1
```

targetnode specifies the node on which the database to be upgraded is running, because the source Oracle Database is on a 12.1.0.2 cluster.

/u01/app/orabase/product/12.1.0.2/dbhome_1 is the path of the database being patched

db12102_psu is the working copy of the Oracle Database software with patches applied, based on the image DB12102_PSU.

Use the saved gold image for standardized patching of all your databases of release 12c release 1 to the same patch level.

3. If for some reason, you want to rollback the patches applied to a managed Oracle Database home, run:

```
rhpctl move database -sourcewc db12102_psu  
-patchedwc db12102 -ignorewcpatches
```

db12102 is the working copy of the unpatched database to which you want to roll back.

db12102_psu is the working copy of the Oracle Database software with patches applied, based on the image DB12102_PSU.

For all Oracle Databases, you can also specify these additional options with the `move database` command:

- `-keepplacement`: For admin-managed Oracle RAC Databases (not Oracle RAC One Node Database), Fleet Patching and Provisioning retains the services on the same nodes after the move.
- `-disconnect`: Disconnects all sessions before stopping or relocating services.
- `-drain_timeout`: Specify the time, in seconds, allowed for resource draining to be completed for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. This option is available only with Oracle Database 12c release 2 (12.2) or later.
- `-stopoption`: Stops the database.
- `-nodatapatch`: Ensures `datapatch` is not run for databases you are moving.

Watch a video  [Video](#)

Upgrading to Oracle Database 12c Release 2

This procedure describes how to upgrade an Oracle database from Oracle Database 11g release 2 (11.2) to 12c release 2 with a single command, using Fleet Patching and Provisioning, both for managed and unmanaged Oracle homes.

Before you Begin

- To upgrade to Oracle Database 12c release 2 (12.2.0.1), your source database must be either Oracle Database 11g release 2 (11.2.0.3 or 11.2.0.4), or Oracle Database 12c release 1 (12.1.0.2).
- Oracle Grid Infrastructure on which the pre-upgrade database is running must be of the same release or later than the database release to which you are upgrading.
- The source Oracle home to be upgraded can be a managed working copy, that is an Oracle home provisioned using Fleet Patching and Provisioning, or an unmanaged home, that is, an Oracle home not provisioned using Fleet Patching and Provisioning. If you are upgrading an unmanaged Oracle home, provide the complete path of the database for upgrade.

Procedure to Upgrade Oracle Database using Fleet Patching and Provisioning

- From the Fleet Patching and Provisioning Server, run one of the following commands as per your source and destination database:
 1. To upgrade an Oracle home managed by Fleet Patching and Provisioning, and there exist working copies of the source and destination databases, run:

```
$ rhpctl upgrade database -dbname test_database -sourcewc db112 -  
destwc db122  
  {authentication_option}
```

test_database is the name of the database being upgraded.

db112 is the source working copy of the pre-upgrade database.

db122 is the working copy of the upgraded Oracle Database software.

2. To upgrade an unmanaged Oracle home (source working copy does not exist because the Oracle home is not managed by Fleet Patching and Provisioning), run:

```
$ rhpctl move database -sourcehome /u01/app/orabase/product/11.2.0/  
dbhome_1  
  -destwc db122 -targetnode node1 {authentication_option}
```

/u01/app/orabase/product/11.2.0/dbhome_1 is the path of the database being upgraded.

db122 is the working copy of the upgraded Oracle Database software.

targetnode specifies the node on which the database to be upgraded is running, because the source Oracle Database is on a 11.2.0.4 cluster.

The upgraded database is now managed by Fleet Patching and Provisioning. You can ensure that your database is patched to the latest level, using Fleet Patching and Provisioning.

Note:

During upgrade, if an error occurs, the procedure stops and allows you to fix the error. After fixing the error, you can resume the upgrade operation from where it last stopped.

Watch a video  [Video](#)

Related Topics

- [rhpctl upgrade database](#)
Upgrades a database to the version of the destination working copy.
- [rhpctl move database](#)
Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.

Adding a Node to a Cluster and Scaling an Oracle RAC Database to the Node

You can add a node to your two-node cluster by using Fleet Patching and Provisioning to add the node, and then extend an Oracle RAC database to the new node.

Before You Begin

In this procedure, Oracle Grid Infrastructure 12c release 2 (12.2.0.1) is running on the cluster. Working copy `GI_HOME_12202_WCPY` is the active Grid home on this cluster.

The Oracle RAC database home runs on the working copy `DB_HOME_12202_WCPY`.

Ensure that you have storage, network, and operating system requirements configured for the new node as stated in *Oracle Grid Infrastructure Installation Guide*.

Procedure

1. From the Fleet Patching and Provisioning Server, run the following command to add a node to the existing Oracle Grid Infrastructure working copy:

```
rhctl addnode gihome -workingcopy GI_HOME_12202_WCPY -newnodes n3:n3-  
vip {authentication_option}
```

The command extends the cluster by adding `node3`.

2. Add instances to the administrator-managed Oracle RAC database on the new node:

```
rhctl addnode database -workingcopy DB_HOME_12202_WCPY -dbname db321 -  
node n3 {authentication_option}
```

The command extends the database home on the `node3` and creates database `db321` on this node.

Related Topics

- [rhctl addnode gihome](#)
- [rhctl addnode database](#)

Adding Gold Images for Fleet Patching and Provisioning

Create gold images of software home and store them on the Fleet Patching and Provisioning Server, to use later to provision Oracle homes.

Before You Begin

The Oracle home to be used for creating the gold image can be on the Fleet Patching and Provisioning Server, or Fleet Patching and Provisioning Client, or any target machine that the Fleet Patching and Provisioning Server can communicate with.

Procedure

Create gold images of Oracle homes in any of the following ways and store them on the Fleet Patching and Provisioning server:

1. Import an image from an installed Oracle home on the Fleet Patching and Provisioning Server:

```
rhctl import image -image db12201 -path /share/software/122/dbhome -  
imagetype ORACLEDBSOFTWARE
```

The gold image of `imagetype` Oracle Database 12c release 2 software is created and stored on the Fleet Patching and Provisioning Server.

You can also create gold images of Oracle Grid Infrastructure or any other software by specifying `-imagetype` as `ORACLEGISOFTWARE`, `ORACLEGGSOFTWARE`, or `SOFTWARE` respectively.

2. Import an image from an installed Oracle home on a Fleet Patching and Provisioning Client by running the following command from the Fleet Patching and Provisioning Client:

```
rhctl import image -image db12201 -path /u01/app/dbusr/product/12.2.0/
```

The command creates and adds the image `db12201` based on the local Oracle home installed in the specified path.

Note:

You cannot directly use images as software homes. Use images to create working copies of software homes.

User Actions for Common Fleet Patching and Provisioning Tasks

You can use Fleet Patching and Provisioning user actions to perform many tasks, such as installing and configuring any type of software and running scripts.

Deploying a Web Server

The following procedure demonstrates automated deployment of Apache Web Server using Fleet Patching and Provisioning:

1. Create a script to install Apache Web server, as follows:
 - a. On the Fleet Patching and Provisioning Server, download and extract the Apache Web server installation kit.
 - b. Create the script to install, configure, and start the Apache Web server.

2. Register the script as a user action with Fleet Patching and Provisioning by running the following command on the Fleet Patching and Provisioning Server:

```
rhpcctl useraction -useraction apachestart
-actionscript /user1/useractions/apacheinstall.sh
-post -optype ADD_WORKINGCOPY -onerror ABORT
```

The preceding command adds the `apachestart` user action for the action script stored in the specified directory. As per the specified properties, the user action runs after the `ADD_WORKINGCOPY` operation and aborts if there is any error.

3. Create an image type and associate the user action with the image type, as follows:

```
rhpcctl add imagetype -imagetype apachetype -basetype SOFTWARE
-useraction "apachestart"
```

The preceding command creates a new image type called `apachetype`, a derivative of the basic image type, `SOFTWARE`, with an associated user action `apachestart`.

4. Create a gold image of the image type, as follows:

```
rhpcctl import image -image apacheinstall -path /user1/apache2219_kit/
-imagetype apachetype
```

The preceding command creates a gold image, `apacheinstall`, with the script for Apache Web server installation, in the specified path, based on the `imagetype` you created earlier.

To view the properties of this image, run the `rhpcctl query image -image apacheinstall` command.

5. Deploy a working copy of the gold image on the target, as follows:

```
rhpcctl add workingcopy -workingcopy apachecopy -image apacheinstall
-path /user1/apacheinstallloc -sudouser user1
-sudopath /usr/local/bin/sudo -node node1 -user user1
-useractiondata "/user1/apachehome:1080:2.2.19"
```

Fleet Patching and Provisioning provisions the software to the target and runs the `apachestart` script specified in the user action. You can provide the Apache Web server configuration details such as port number with the `useractiondata` option. If the target is a Fleet Patching and Provisioning Client, then you need not specify sudo credentials.

Registering Multiple Scripts Using a Single User Action

Run multiple scripts as part of a user action plug-in by registering a wrapper script and bundled custom scripts. The wrapper script extracts the bundled scripts, which are copied under the directory of the wrapper script, and then runs those extracted scripts as necessary, similar to the following procedure:

1. The following command creates a user action called `ohadd_ua`, and associates a wrapper script, `wc_add.sh`, with a zip file containing other scripts:

```
rhptl add useraction -useraction ohadd_ua -actionscrip  
t /scratch/crsusr/wc_add.sh -actionfile /scratch/crsusr/pack.zip -pre -  
runscope  
ALLNODES -optype ADD_WORKINGCOPY
```

The wrapper script, `wc_add.sh`, extracts the `pack.zip` file into the script path, a temporary path to which the user action scripts are copied. The wrapper script can invoke any scripts contained in the file.

2. The following command creates an image type, `sw_ua`, for the `ohadd_ua` user action:

```
rhptl add imagetype -imagetype sw_ua -useractions ohadd_ua -basetype  
SOFTWARE
```

3. The following command creates an image called `swimgua` from the software specified in the path:

```
rhptl import image -image swimgua -path /tmp/custom_sw -imagetype sw_ua
```

4. The following command adds a working copy called `wcua` and runs the `wc_add.sh` script:

```
rhptl add workingcopy -workingcopy wcua -image swimgua -client  
targetcluster
```

6

Managing Oracle Cluster Registry and Voting Files

Oracle Clusterware includes two important components that manage configuration and node membership: Oracle Cluster Registry (OCR), which also includes the local component Oracle Local Registry (OLR), and voting files.

- OCR stores Oracle Clusterware and Oracle RAC database configuration information
- OLR resides on every node in the cluster and manages Oracle Clusterware configuration information for each particular node
- Voting files store information about node membership. Each voting file must be accessible by all nodes in the cluster for nodes to be members of the cluster

Notes:

- Oracle Clusterware 12c does not support the use of raw or block devices. To upgrade to Oracle Clusterware 12c from a previous Oracle Clusterware release on which you were using raw or block devices, you must migrate OCR and voting files to Oracle Automatic Storage Management (Oracle ASM) or a shared file system before you upgrade.
- You can store OCR and voting files on Oracle ASM or a shared file system on Standalone Clusters.

In Oracle Database 12c Release 2 (12.2), the placement of OCR and voting disk files directly on a shared file system was desupported. Starting with Oracle Database 19c (19.3), that desupport is rescinded for Standalone Clusters. For Oracle Domain Services Clusters, you must continue to place OCR and voting disk files in disk groups managed by Oracle ASM.

Oracle recommends that you configure multiple voting files during Oracle Clusterware installation to improve availability. If you choose to put the voting files into an Oracle ASM disk group, then Oracle ASM ensures the configuration of multiple voting files if you use a normal or high redundancy disk group. If you choose to store the voting files on a shared file system, then select the option to configure multiple voting files, in which case you will have to specify three different file systems based on different disks.

If necessary, you can dynamically add or replace voting files after you complete the Oracle Clusterware installation process without stopping the cluster.

This chapter includes the following topics:

- [Managing Oracle Cluster Registry and Oracle Local Registry](#)

- [Managing Voting Files](#)

Managing Oracle Cluster Registry and Oracle Local Registry

Use OCRCONFIG, OCRDUMP, and OCRCHECK to manage OCR and the Oracle Local Registry (OLR).

OCR contains information about all Oracle resources in the cluster.

OLR is a registry similar to OCR located on each node in a cluster, but contains information specific to each node. It contains manageability information about Oracle Clusterware, including dependencies between various services. Oracle High Availability Services uses this information. OLR is located on local storage on each node in a cluster. Its default location is in the path *Grid_home/cdatahost_name.olr*, where *Grid_home* is the Oracle Grid Infrastructure home, and *host_name* is the host name of the node.

Administering OCR is described in the following topics:

- [Migrating Oracle Cluster Registry to Oracle Automatic Storage Management](#)
- [Adding_ Replacing_ Repairing_ and Removing Oracle Cluster Registry Locations](#)
- [Backing Up Oracle Cluster Registry](#)
- [Restoring Oracle Cluster Registry](#)
- [Diagnosing Oracle Cluster Registry Problems](#)
- [Administering Oracle Cluster Registry with Export and Import Commands](#)
- [Oracle Local Registry](#)
- [Upgrading and Downgrading the Oracle Cluster Registry Configuration](#)

Related Topics

- [About OCRCONFIG](#)
Use the `ocrconfig` command to manage OCR. Using this utility you can import, export, add, delete, restore, overwrite, backup, repair, replace, move, upgrade, or downgrade OCR.
- [Troubleshooting Oracle Cluster Registry](#)

Migrating Oracle Cluster Registry to Oracle Automatic Storage Management

With the Oracle Clusterware storage residing in an Oracle ASM disk group, you can manage both database and clusterware storage using Oracle Enterprise Manager.

To improve Oracle Clusterware storage manageability, OCR is configured, by default, to use Oracle ASM in Oracle Database 12c. However, if you upgrade from a previous version of Oracle Clusterware, you can migrate OCR to reside on Oracle ASM, and take advantage of the improvements in managing Oracle Clusterware storage.

 **Note:**

If you upgrade from a previous version of Oracle Clusterware to Oracle Clusterware 12c and you want to store OCR in an Oracle ASM disk group, then you must set the **ASM Compatibility** compatibility attribute to 11.2.0.2, or later.

To migrate OCR to Oracle ASM using OCRCONFIG:

1. Ensure the upgrade to Oracle Clusterware 12c is complete. Run the following command to verify the current running version:

```
$ crsctl query crs activeversion
```

2. Use the Oracle ASM Configuration Assistant (ASMCA) to configure and start Oracle ASM on all nodes in the cluster.
3. Use ASMCA to create an Oracle ASM disk group that is at least the same size of the existing OCR and has at least normal redundancy.

 **Note:**

- If OCR is stored in an Oracle ASM disk group with external redundancy, then Oracle recommends that you add another OCR location to another disk group to avoid the loss of OCR, if a disk fails in the disk group.

Oracle does not support storing OCR on different storage types simultaneously, such as storing OCR on both Oracle ASM and a shared file system, except during a migration.

- If an Oracle ASM instance fails on any node, then OCR becomes unavailable on that particular node.

If the CRSD process running on the node affected by the Oracle ASM instance failure is the [OCR writer](#), the majority of the OCR locations are stored in Oracle ASM, and you attempt I/O on OCR during the time the Oracle ASM instance is down on this node, then CRSD stops and becomes inoperable. Cluster management is now affected on this particular node.

Under no circumstances will the failure of one Oracle ASM instance on one node affect the whole cluster.

- Ensure that Oracle ASM disk groups that you create are mounted on all of the nodes in the cluster.

4. To add OCR to an Oracle ASM disk group, ensure that the [Oracle Clusterware stack](#) is running and run the following command as root:

```
# ocrconfig -add +new_disk_group
```

You can run this command more than once if you add multiple OCR locations. You can have up to five OCR locations. However, each successive run must point to a different disk group.

5. To remove storage configurations no longer in use, run the following command as root:

```
# ocrconfig -delete old_storage_location
```

Run this command for every configured OCR.

The following example shows how to migrate two OCRs to Oracle ASM using OCRCONFIG.

```
# ocrconfig -add +new_disk_group
# ocrconfig -delete /ocrdata/ocr_1
# ocrconfig -delete /ocrdata/ocr_2
```

Note:

- OCR inherits the redundancy of the disk group. If you want high redundancy for OCR, you must configure the disk group with high redundancy when you create it.
- Oracle recommends that you put the SPFILE for Oracle ASM in this newly-created OCR location.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Grid Infrastructure Installation and Upgrade Guide*

Migrating Oracle Cluster Registry from Oracle ASM to Other Types of Storage

To migrate OCR from Oracle ASM to another storage type:

1. Ensure the upgrade to Oracle Clusterware 12c is complete. Run the following command to verify the current running version:

```
$ crsctl query crs activeversion
```

2. Create a file in a shared or cluster file system with the following permissions: root, oinstall, 640.

Note:

Create at least two mirrors of the primary storage location to eliminate a single point of failure for OCR. OCR supports up to five locations.

3. Ensure there is at least 500 MB of space on the mount partition.
4. Ensure that the file you created is visible from all nodes in the cluster.
5. To add the file as an OCR location, ensure that the Oracle Clusterware stack is running and run the following command as `root`:

```
# ocrconfig -add file_location
```

You can run this command more than once if you add more than one OCR location. Each successive run of this command must point to a different file location.

6. Ensure that the OCR backup location is not configured on an Oracle ASM disk group.

To migrate the OCR backup location to a file system, run the following command as `root`:

```
# ocrconfig -backuploc file_location
```

7. To remove storage configurations no longer in use, run the following command as `root`:

```
# ocrconfig -delete +asm_disk_group
```

You can run this command more than once if there is more than one OCR location configured.

The following example shows how to migrate OCR from Oracle ASM to NAS using `OCRCONFIG`.

```
# ocrconfig -add /nas/ocr
```

For OCRs not stored on Oracle ASM, Oracle recommends that you mirror OCR on different devices.

Adding, Replacing, Repairing, and Removing Oracle Cluster Registry Locations

Manage OCR locations using `OCRCONFIG`.

The Oracle installation process for Oracle Clusterware gives you the option of automatically mirroring OCR. You can manually put the mirrored OCRs on a shared Network File System (NFS), or on any cluster file system that is certified by Oracle. Alternatively, you can place OCR on Oracle ASM and allow it to create mirrors automatically, depending on the redundancy option you select.

You can manually mirror OCR, as described in the *Adding an Oracle Cluster Registry Location* section, if you:

- Upgraded to Oracle Clusterware 12c but did not choose to mirror OCR during the upgrade
- Created only one OCR location during the Oracle Clusterware installation

Oracle recommends that you configure:

- At least three OCR locations, if OCR is configured on non-mirrored or non-redundant storage. Oracle strongly recommends that you mirror OCR if the

underlying storage is not RAID. Mirroring can help prevent OCR from becoming a single point of failure.

- At least two OCR locations if OCR is configured on an Oracle ASM disk group. You should configure OCR in two independent disk groups. Typically this is the work area and the recovery area.
- At least two OCR locations if OCR is configured on mirrored hardware or third-party mirrored volumes.

 **Note:**

- If the original OCR location does not exist, then you must create an empty (0 byte) OCR location with appropriate permissions before you run the `ocrconfig -add` or `ocrconfig -replace` commands.
- Ensure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.
- Ensure that the Oracle ASM disk group that you specify exists and is mounted.
- The new OCR file, device, or disk group must be accessible from all of the active nodes in the cluster.

In addition to mirroring OCR locations, managing OCR locations include:

- [Adding an Oracle Cluster Registry Location](#)
- [Removing an Oracle Cluster Registry Location](#)
- [Replacing an Oracle Cluster Registry Location](#)
- [Repairing an Oracle Cluster Registry Configuration on a Local Node](#)
- [Overriding the Oracle Cluster Registry Data Loss Protection Mechanism](#)

 **Note:**

The operations in this section affect OCR throughout the entire cluster: the operations change the OCR configuration information in the `ocr.loc` file on Linux and UNIX systems and the Registry keys on Windows systems. However, the `ocrconfig` command cannot modify OCR configuration information for nodes that are shut down or for nodes on which Oracle Clusterware is not running.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Automatic Storage Management Administrator's Guide*

Adding an Oracle Cluster Registry Location

Use the procedure in this section to add an OCR location. Oracle Clusterware can manage up to five redundant OCR locations.

Note:

If OCR resides on a cluster file system file or a network file system, create an empty (0 byte) OCR location file before performing the procedures in this section.

As the `root` user, run the following command to add an OCR location to either Oracle ASM or other storage device:

```
# ocrconfig -add +asm_disk_group | file_name
```

Note:

On Linux and UNIX systems, you must be `root` to run `ocrconfig` commands. On Windows systems, the user must be a member of the Administrator's group.

Removing an Oracle Cluster Registry Location

To remove an OCR location or a failed OCR location, at least one other OCR must be online. You can remove an OCR location to reduce OCR-related overhead or to stop mirroring your OCR because you moved OCR to redundant storage such as RAID.

Perform the following procedure as the `root` user to remove an OCR location from your Oracle Clusterware environment:

1. Ensure that *at least one* OCR location other than the OCR location that you are removing is online.

Caution:

Do *not* perform this OCR removal procedure unless there is at least one other active OCR location online.

2. Run the following command on any node in the cluster to remove an OCR location from either Oracle ASM or other location:

```
# ocrconfig -delete +ASM_disk_group | file_name
```

The `file_name` variable can be a device name or a file name. This command updates the OCR configuration on all of the nodes on which Oracle Clusterware is running.

 **Note:**

The only configured Oracle ASM OCR location, which is on a different Oracle ASM disk group than the Oracle ASM OCR backup location, cannot be removed.

Replacing an Oracle Cluster Registry Location

If you must change an existing OCR location, or change a failed OCR location to a working location, then you can use the following procedure, if all remaining OCR locations remain online. The `ocrconfig -replace` command requires that at least two OCR locations are configured.

To change an Oracle Cluster Registry location:

Complete the following procedure:

1. Use the `OCRCHECK` utility to verify that a copy of OCR other than the one you are going to replace is *online*, using the following command:

```
$ ocrcheck
```

`OCRCHECK` displays all OCR locations that are registered and whether they are available (online). If an OCR location suddenly becomes unavailable, then it might take a short period for Oracle Clusterware to show the change in status.

 **Note:**

The OCR location that you are *replacing* can be either online or offline.

2. Use the following command to verify that Oracle Clusterware is running on the node on which you are going to perform the replace operation:

```
$ crsctl check crs
```

3. Run the following command as `root` to replace the current OCR location using either `destination_file` or `+ASM_disk_group` to indicate the current and target OCR locations:

```
# ocrconfig -replace current_OCR_location -replacement new_OCR_location
```

The preceding command fails if you have less than two configured OCR locations that are online.

If you have only one OCR location configured and online, then you must first add a new location and then delete the failed location, as follows:

```
# ocrconfig -add new_OCR_location  
# ocrconfig -delete current_OCR_location
```

 **Note:**

If your cluster configuration changes while the node on which OCR resides is stopped, and the Oracle Clusterware stack is running on the other nodes, then OCR detects configuration changes and self-corrects the configuration by changing the contents of the `ocr.loc` file.

Related Topics

- [Oracle Automatic Storage Management Administrator's Guide](#)

Repairing an Oracle Cluster Registry Configuration on a Local Node

It may be necessary to repair OCR if your cluster configuration changes while that node is stopped and this node is the only member in the cluster.

Repairing an OCR involves either adding, deleting, or replacing an OCR location. For example, if any node that is part of your current Oracle RAC cluster is shut down, then you must update the OCR configuration on the stopped node to let that node rejoin the cluster after the node is restarted. Use the following command syntax as `root` on the restarted node where you use either a `destination_file` or `+ASM_disk_group` to indicate the current and target OCR locations:

```
# ocrconfig -repair -replace current_OCR_location -replacement  
target_OCR_location
```

This operation only changes OCR on the node on which you run this command. For example, if the OCR location is `/dev/sde1`, then use the command syntax `ocrconfig -repair -add /dev/sde1` on this node to repair OCR on that node.

 **Note:**

- You cannot repair the OCR configuration on a node on which the Oracle Cluster Ready Services daemon is running.
- When you repair OCR on a stopped node using `ocrconfig -repair`, you must provide the same OCR file name (which should be case-sensitive) as the OCR file names on other nodes.
- If you run the `ocrconfig -add | -repair | -replace` command, then the device, file, or Oracle ASM disk group that you are adding must be accessible. This means that a device must exist. You must create an empty (0 byte) OCR location, or the Oracle ASM disk group must exist and be mounted.

Related Topics

- [OCRCONFIG Command Reference](#)
- [Oracle Automatic Storage Management Administrator's Guide](#)

Overriding the Oracle Cluster Registry Data Loss Protection Mechanism

OCR has a mechanism that prevents data loss due to accidental overwrites. If you configure a mirrored OCR and if Oracle Clusterware cannot access the mirrored OCR locations and also cannot verify that the available OCR location contains the most recent configuration, then Oracle Clusterware prevents further modification to the available OCR location. In addition, the process prevents overwriting by prohibiting Oracle Clusterware from starting on the node on which only one OCR is available. In such cases, Oracle Database displays an alert message in either Oracle Enterprise Manager, the Oracle Clusterware alert log files, or both. If this problem is local to only one node, you can use other nodes to start your cluster database.

However, if you are unable to start any cluster node in your environment and if you can neither repair OCR nor restore access to all OCR locations, then you can override the protection mechanism. The procedure described in the following list enables you to start the cluster using the available OCR location. However, overriding the protection mechanism can result in the loss of data that was not available when the previous known good state was created.

Caution:

Overriding OCR using the following procedure can result in the loss of OCR updates that were made between the time of the last known good OCR update made to the currently accessible OCR and the time at which you performed the overwrite. In other words, running the `ocrconfig -overwrite` command can result in data loss if the OCR location that you are using to perform the overwrite does not contain the latest configuration updates for your cluster environment.

Perform the following procedure to overwrite OCR if a node cannot start *and* if the alert log contains CLSD-1009 and CLSD-1011 messages.

1. Attempt to resolve the cause of the CLSD-1009 and CLSD-1011 messages.

Compare the node's OCR configuration (`ocr.loc` on Linux and UNIX systems and the Registry on Windows systems) with other nodes on which Oracle Clusterware is running.

- If the configurations do not match, run `ocrconfig -repair`.
- If the configurations match, ensure that the node can access all of the configured OCRs by running an `ls` command on Linux and UNIX systems. On Windows, use a `dir` command if the OCR location is a file and run `GuiOracleObjectManager.exe` to verify that the part of the cluster with the name exists.

2. Ensure that the most recent OCR contains the latest OCR updates.

Look at output from the `ocrdump` command and determine whether it has your latest updates.

3. If you cannot resolve the problem that caused the CLSD message, then run the command `ocrconfig -overwrite` to start the node.

Backing Up Oracle Cluster Registry

This section describes how to back up OCR content and use it for recovery. The first method uses automatically generated OCR copies and the second method enables you to issue a backup command manually:

- **Automatic backups:** Oracle Clusterware automatically creates OCR backups every four hours. At any one time, Oracle Database always retains the last three backup copies of OCR. The CRSD process that creates the backups also creates and retains an OCR backup for each *full day* and *after each week*. You cannot customize the backup frequencies or the number of files that Oracle Database retains.
- **Manual backups:** Run the `ocrconfig -manualbackup` command on a node where the Oracle Clusterware stack is up and running to force Oracle Clusterware to perform a backup of OCR at any time, rather than wait for the automatic backup. You must run the command as a user with administrative privileges. The `-manualbackup` option is especially useful when you want to obtain a binary backup on demand, such as before you make changes to OCR. The OLR only supports manual backups.

When the clusterware stack is down on all nodes in the cluster, the backups that are listed by the `ocrconfig -showbackup` command may differ from node to node.

Note:

After you install or upgrade Oracle Clusterware on a node, or add a node to the cluster, when the `root.sh` script finishes, it backs up OLR.

This section includes the following topics:

- [Listing Backup Files](#)
- [Changing Backup Location](#)

Listing Backup Files

Run the following command to list the backup files:

```
ocrconfig -showbackup
```

The `ocrconfig -showbackup` command displays the backup location, timestamp, and the originating node name of the backup files that Oracle Clusterware creates. By default, the `-showbackup` option displays information for both automatic and manual backups but you can include the `auto` or `manual` flag to display only the automatic backup information or only the manual backup information, respectively.

Run the following command to inspect the contents and verify the integrity of the backup file:

```
ocrdump -backupfile backup_file_name
```

You can use any backup software to copy the automatically generated backup files at least once daily to a different device from where the primary OCR resides.

As of Oracle Clusterware 12c release 2 (12.2), the default location for generating backups is an Oracle ASM disk group, which you can change between Oracle ASM disk groups, but you *cannot* change to a local file system. Oracle recommends that you include the backup file created with the OCRCONFIG utility as part of your operating system backup using standard operating system or third-party tools.

Changing Backup Location

Run the following command to change the location where OCR creates backups:

```
# ocrconfig -backuploc file_name
```

The *file_name* variable in the preceding command can be a full directory path name that is accessible by all nodes, or it can be an Oracle ASM disk group that is mounted on all nodes. You must migrate OCR to Oracle ASM before changing the OCR backup location to an Oracle ASM disk group. You can change the OCR backup location to an Oracle ASM disk group only if there is at least one Oracle ASM OCR location in a separate disk group.

For example, to specify an OCR backup location in a directory:

```
# ocrconfig -backuploc Grid_home/cdata/cluster3
```

To specify an OCR backup location in an Oracle ASM disk group:

```
# ocrconfig -backuploc +bkupdg
```



Note:

On Linux and UNIX systems, you must be `root` user to run most but not all of the `ocrconfig` command options. On Windows systems, the user must be a member of the Administrator's group.

Related Topics

- [Administering Oracle Cluster Registry with Export and Import Commands](#)
- [OCRCONFIG Command Reference](#)

Restoring Oracle Cluster Registry

A procedure for OCR restoration for your platform.

If a resource fails, then before attempting to restore OCR, restart the resource. As a definitive verification that OCR failed, run `ocrcheck` and if the command returns a failure message, then both the primary OCR and the OCR mirror have failed.

 **Note:**

- You *cannot* restore your configuration from an OCR backup file using the `-import` option, which is explained in "Administering Oracle Cluster Registry with Export and Import Commands". You *must instead* use the `-restore` option, as described in the following sections.
- If you store OCR on an Oracle ASM disk group and the disk group is not available, then you must recover and mount the Oracle ASM disk group.

- [Restoring the Oracle Cluster Registry on Linux or UNIX Systems](#)
- [Restoring the Oracle Cluster Registry on Windows Systems](#)
- [Restoring the Oracle Cluster Registry in an Oracle Restart Environment](#)

Related Topics

- [Administering Oracle Cluster Registry with Export and Import Commands](#)
- [Oracle Automatic Storage Management Administrator's Guide](#)

Restoring the Oracle Cluster Registry on Linux or UNIX Systems

Use this procedure to restore OCR on Linux or UNIX systems.

If you are storing OCR on an Oracle ASM disk group, and that disk group is corrupt, then you must restore the Oracle ASM disk group using Oracle ASM utilities, and then mount the disk group again before recovering OCR. Recover OCR by running the `ocrconfig -restore` command, as instructed in the following procedure.

 **Note:**

If the original OCR location does not exist, then you must create an empty (0 byte) OCR location with the same name as the original OCR location before you run the `ocrconfig -restore` command.

Use the following procedure to restore OCR on Linux or UNIX systems:

1. List the nodes in your cluster by running the following command on one node:

```
$ olsnodes
```

2. Depending on whether OCR is located in an Oracle ASM disk group or on network attached storage (NAS), stop Oracle Clusterware by running one of the following commands as root on all of the nodes:

If OCR is located in an Oracle ASM disk group, then stop the Oracle Clusterware daemon:

```
# crsctl stop crs
```

If the preceding command returns any error due to OCR corruption, then force stop Oracle Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl stop crs -f
```

3. If you are restoring OCR to a cluster file system or network file system, then run the following command as `root` to restore OCR with an OCR backup that you can identify in "Listing Backup Files":

```
# ocrconfig -restore file_name
```

 **Note:**

If the OCR backup file is located in an Oracle ASM disk group, then ensure that the disk group exists and is mounted.

After you complete this step, proceed to step 11.

If you are *not* restoring OCR to a cluster file system or network file system, then proceed to the next step.

4. Start the Oracle Clusterware stack on one node in exclusive mode by running the following command as `root`:

```
# crsctl start crs -excl -nocrs
```

The `-nocrs` option ensures that the CRSD process and OCR do not start with the rest of the Oracle Clusterware stack.

Ignore any errors that display.

5. Check whether CRSD is running by running the following command:

```
$ crsctl status resource ora.crsd -init
```

If CRSD is running, then stop it by running the following command as `root`:

```
# crsctl stop resource ora.crsd -init
```

 **Caution:**

Do not use the `-init` flag with any other command unless directed to do so by My Oracle Support.

6. To restore OCR to an Oracle ASM disk group, you must first create a disk group using SQL*Plus that has the same name as the disk group you want to restore and mount it on the local node.

If you cannot mount the disk group locally, then run the following SQL*Plus command:

```
SQL> drop diskgroup disk_group_name force including contents;
```

7. Restore OCR with an OCR backup that you can identify in "Listing Backup Files" by running the following command as `root`:

```
# ocrconfig -restore file_name
```

Note:

- If the original OCR location does not exist, then you must create an empty (0 byte) OCR location before you run the `ocrconfig -restore` command.
- Ensure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.
- If you configured OCR in an Oracle ASM disk group, then ensure that the Oracle ASM disk group exists and is mounted.
- If the OCR backup file is located in an Oracle ASM disk group, then ensure that the disk group exists and is mounted.

See Also:

Oracle Grid Infrastructure Installation and Upgrade Guide for information about creating OCRs

8. Verify the integrity of OCR:

```
# ocrcheck
```

9. Stop Oracle Clusterware on the node where it is running in exclusive mode:

```
# crsctl stop crs -f
```

10. Run the `ocrconfig -repair -replace` command as `root` on all the nodes in the cluster where you *did not* run the `ocrconfig -restore` command. For example, if you ran the `ocrconfig -restore` command on node 1 of a four-node cluster, then you must run the `ocrconfig -repair -replace` command on nodes 2, 3, and 4.

11. Begin to start Oracle Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl start crs
```

12. Verify OCR integrity of all of the cluster nodes that are configured as part of your cluster by running the following CVU command:

```
$ cluvfy comp ocr -n all -verbose
```

Related Topics

- [Oracle Automatic Storage Management Administrator's Guide](#)
- [Listing Backup Files](#)
- [Oracle Automatic Storage Management Administrator's Guide](#)
- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)
- [Cluster Verification Utility Reference](#)

Restoring the Oracle Cluster Registry on Windows Systems

Use this procedure to restore OCR on Windows systems.

If you are storing OCR on an Oracle ASM disk group, and that disk group is corrupt, then you must restore the Oracle ASM disk group using Oracle ASM utilities, and then mount the disk group again before recovering OCR. Recover OCR by running the `ocrconfig -restore` command.

Note:

If the original OCR location does not exist, then you must create an empty (0 byte) OCR location with the same name as the original OCR location before you run the `ocrconfig -restore` command.

Use the following procedure to restore OCR on Windows systems:

1. List the nodes in your cluster by running the following command on one node:

```
C:\>olsnodes
```

2. Stop Oracle Clusterware by running the following command as a member of the Administrators group on all of the nodes:

```
C:\>crsctl stop crs
```

If the preceding command returns any error due to OCR corruption, stop Oracle Clusterware by running the following command as a member of the Administrators group on all of the nodes:

```
C:\>crsctl stop crs -f
```

3. Start the Oracle Clusterware stack on one node in exclusive mode by running the following command as a member of the Administrators group:

```
C:\>crsctl start crs -excl -nocrs
```

The `-nocrs` option ensures that the CRS process and OCR do not start with the rest of the Oracle Clusterware stack.

Ignore any errors that display.

4. Restore OCR with the OCR backup file that you identified in "Listing Backup Files" by running the following command as a member of the Administrators group:

```
C:\>ocrconfig -restore file_name
```

Make sure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.

Note:

- Ensure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.
- Ensure that the Oracle ASM disk group you specify exists and is mounted.

See Also:

Oracle Grid Infrastructure Installation and Upgrade Guide for information about creating OCRs

5. Verify the integrity of OCR:

```
C:\>ocrcheck
```

6. Stop Oracle Clusterware on the node where it is running in exclusive mode:

```
C:\>crsctl stop crs -f
```

7. Begin to start Oracle Clusterware by running the following command as a member of the Administrators group on all of the nodes:

```
C:\>crsctl start crs
```

8. Run the following Cluster Verification Utility (CVU) command to verify OCR integrity of all of the nodes in your cluster database:

```
C:\>cluvfy comp ocr -n all -verbose
```

Related Topics

- [Listing Backup Files](#)
- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Automatic Storage Management Administrator's Guide*

- [Cluster Verification Utility Reference](#)

Restoring the Oracle Cluster Registry in an Oracle Restart Environment

Use this procedure to restore OCR in an Oracle Restart environment.

Note:

- OCR is present for backward compatibility.
- Once an OCR location is created, it does not get updated in the Oracle Restart environment.
- If the Oracle Restart home has been backed up, and if there is a failure, then restoring the Oracle Restart home restores OCR.

1. Stop Oracle High Availability Services by running the following command as `root` on all of the nodes:

```
# crsctl stop has [-f]
```

2. Run the `ocrcheck -config` command to determine the OCR location and then create an empty (0 byte) OCR location with appropriate permissions in that location.

3. Restore OCR by running the following command as `root`:

```
# crsctl pin css -n host_name
```

Note:

Ensure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.

See Also:

Oracle Grid Infrastructure Installation and Upgrade Guide for information about creating OCRs

4. Run the `ocrcheck` command to verify the integrity of OCR:
5. Start Oracle High Availability Services by running the following command on all of the nodes:

```
$ crsctl start has
```

Related Topics

- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)

Diagnosing Oracle Cluster Registry Problems

Use the OCRDUMP and OCRCHECK utilities to diagnose OCR problems.

Related Topics

- [OCRDUMP Utility Syntax and Options](#)
- [Using the OCRCHECK Utility](#)

The OCRCHECK utility displays the version of the OCR's block format, total space available and used space, OCRID, and the OCR locations that you have configured.

Administering Oracle Cluster Registry with Export and Import Commands

In addition to using the automatically created OCR backup files, you should also export OCR contents before and after making significant configuration changes, such as adding or deleting nodes from your environment, modifying Oracle Clusterware resources, and upgrading, downgrading or creating a database. Do this by using the `ocrconfig -export` command, which exports OCR content to a file format.

▲ Caution:

Note the following restrictions for restoring OCR:

- The file format generated by `ocrconfig -restore` is incompatible with the file format generated by `ocrconfig -export`. The `ocrconfig -export` and `ocrconfig -import` commands are compatible. The `ocrconfig -manualbackup` and `ocrconfig -restore` commands are compatible. The two file formats are incompatible and must not be interchangeably used.
- When exporting OCR, Oracle recommends including "ocr", the cluster name, and the timestamp in the name string. For example:

```
ocr_mycluster1_20090521_2130_export
```

Using the `ocrconfig -export` command also enables you to restore OCR using the `-import` option if your configuration changes cause errors. For example, if you have configuration problems that you cannot resolve, or if you are unable to restart Oracle Clusterware after such changes, then restore your configuration using the procedure for your platform.

Oracle recommends that you use either automatic or manual backups, and the `ocrconfig -restore` command instead of the `ocrconfig -export` and `ocrconfig -import` commands to restore OCR for the following reasons:

- A backup is a consistent snapshot of OCR, whereas an export is not.
- Backups are created when the system is online. You must shut down Oracle Clusterware on all nodes in the cluster to get a consistent snapshot using the `ocrconfig -export` command.

- You can inspect a backup using the OCRDUMP utility. You cannot inspect the contents of an export.
- You can list backups with the `ocrconfig -showbackup` command, whereas *you* must keep track of all generated exports.

This section includes the following topics:

- [Importing Oracle Cluster Registry Content on Linux or UNIX Systems](#)
- [Importing Oracle Cluster Registry Content on Windows Systems](#)

 **Note:**

Most configuration changes that you make not only change OCR contents, the configuration changes also cause file and database object creation. Some of these changes are often not restored when you restore OCR. Do not restore OCR as a correction to revert to previous configurations, if some of these configuration changes should fail. This may result in an OCR location that has contents that do not match the state of the rest of your system.

Importing Oracle Cluster Registry Content on Linux or UNIX Systems

Use this procedure to import OCR on Linux or UNIX systems.

 **Note:**

This procedure assumes default installation of Oracle Clusterware on all nodes in the cluster, where Oracle Clusterware autostart is enabled.

1. List the nodes in your cluster by running the following command on one node:

```
$ olsnodes
```

2. Stop Oracle Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl stop crs
```

If the preceding command returns any error due to OCR corruption, stop Oracle Clusterware by running the following command as `root` on all of the nodes:

```
# crsctl stop crs -f
```

3. Start the Oracle Clusterware stack on one node in exclusive mode by running the following command as `root`:

```
# crsctl start crs -excl
```


Ignore any errors that display.

Check whether CRSD is running. If it is, stop it by running the following command as root:

```
# crsctl stop resource ora.crsd -init
```

 **Caution:**

Do not use the `-init` flag with any other command.

4. Import OCR by running the following command as root:

```
# ocrconfig -import file_name
```

If you are importing OCR to a cluster or network file system, then skip to step 7.

 **Note:**

- If the original OCR location does not exist, then you must create an empty (0 byte) OCR location before you run the `ocrconfig -import` command.
- Ensure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.
- If you configured OCR in an Oracle ASM disk group, then ensure that the Oracle ASM disk group exists and is mounted.

 **See Also:**

Oracle Grid Infrastructure Installation and Upgrade Guide for information about creating OCRs

5. Verify the integrity of OCR:

```
# ocrcheck
```

6. Stop Oracle Clusterware on the node where it is running in exclusive mode:

```
# crsctl stop crs -f
```

7. Begin to start Oracle Clusterware by running the following command as root on all of the nodes:

```
# crsctl start crs
```

8. Verify OCR integrity of all of the cluster nodes that are configured as part of your cluster by running the following CVU command:

```
$ cluvfy comp ocr -n all -verbose
```

 **Note:**

You *can only* import an exported OCR. To restore OCR from a backup, you must instead use the `-restore` option, as described in "Backing Up Oracle Cluster Registry".

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Automatic Storage Management Administrator's Guide*
- [Backing Up Oracle Cluster Registry](#)
- [Cluster Verification Utility Reference](#)

Importing Oracle Cluster Registry Content on Windows Systems

Use this procedure to import OCR on Windows systems.

 **Note:**

This procedure assumes default installation of Oracle Clusterware on all nodes in the cluster, where Oracle Clusterware autostart is enabled.

1. List the nodes in your cluster by running the following command on one node:

```
C:\>olsnodes
```

2. Stop Oracle Clusterware by running the following command as a member of the Administrators group on all of the nodes:

```
C:\>crsctl stop crs
```

If the preceding command returns any error due to OCR corruption, stop Oracle Clusterware by running the following command as a member of the Administrators group on all of the nodes:

```
C:\>crsctl stop crs -f
```

3. Start the Oracle Clusterware stack on one node in exclusive mode by running the following command as a member of the Administrators group:

```
C:\>crsctl start crs -excl
```

Ignore any errors that display.

Check whether CRSD is running. If it is, stop it by running the following command as a member of the Administrators group:

```
C:\>crsctl stop resource ora.crsd -init
```

 **Caution:**

Do not use the -init flag in any other command.

4. Import OCR by running the following command as a member of the Administrators group:

```
C:\>ocrconfig -import file_name
```

Make sure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.

 **Note:**

- If the original OCR location does not exist, then you must create an empty (0 byte) OCR location before you run the `ocrconfig -import` command.
- Ensure that the OCR devices that you specify in the OCR configuration exist and that these OCR devices are valid.
- Ensure that the Oracle ASM disk group you specify exists and is mounted.

5. Verify the integrity of OCR:

```
C:\>ocrcheck
```

6. Stop Oracle Clusterware on the node where it is running in exclusive mode:

```
C:\>crsctl stop crs -f
```

7. Begin to start Oracle Clusterware by running the following command as a member of the Administrators group on all of the nodes:

```
C:\>crsctl start crs
```

8. Run the following Cluster Verification Utility (CVU) command to verify OCR integrity of all of the nodes in your cluster database:

```
C:\>cluvfy comp ocr -n all -verbose
```

Related Topics

- [Oracle Automatic Storage Management Administrator's Guide](#)
- [Cluster Verification Utility Reference](#)

Oracle Local Registry

In Oracle Clusterware 12c, each node in a cluster has a local registry for node-specific resources, called an Oracle Local Registry (OLR), that is installed and configured when Oracle Clusterware installs OCR. Multiple processes on each node have simultaneous read and write access to the OLR particular to the node on which they reside, regardless of whether Oracle Clusterware is running or fully functional.

By default, OLR is located at `Grid_home/cdata/host_name.olr` on each node.

Manage OLR using the OCRCHECK, OCRDUMP, and OCRCONFIG utilities as `root` with the `-local` option.

- You can check the status of OLR on the local node using the OCRCHECK utility, as follows:

```
# ocrcheck -local

Status of Oracle Cluster Registry is as follows :
  Version           :           3
  Total space (kbytes) :       262132
  Used space (kbytes)  :           9200
  Available space (kbytes) :       252932
  ID                 :       604793089
  Device/File Name    : /private2/crs/cdata/localhost/dglnx6.olr
                    Device/File integrity check succeeded

Local OCR integrity check succeeded
```

- You can display the content of OLR on the local node to the text terminal that initiated the program using the OCRDUMP utility, as follows:

```
# ocrdump -local -stdout
```

- You can perform administrative tasks on OLR on the local node using the OCRCONFIG utility.

- To export OLR to a file:

```
# ocrconfig -local -export file_name
```

Note:

- * Oracle recommends that you use the `-manualbackup` and `-restore` commands and not the `-import` and `-export` commands.
- * When exporting OLR, Oracle recommends including "olr", the host name, and the timestamp in the name string. For example:

```
olr_myhost1_20090603_0130_export
```

- To import a specified file to OLR:

```
# ocrconfig -local -import file_name
```

- To manually back up OLR:

```
# ocrconfig -local -manualbackup
```

 **Note:**

Oracle Clusterware backs up OLR after installation or upgrade and, by default, periodically backs up OLR, thereafter. At any time after the initial backup, you can manually back up OLR.

Oracle also recommends that you create a new backup when you migrate OCR from Oracle ASM to other storage, or when you migrate OCR from other storage to Oracle ASM.

The default backup location for the OLR is in the path *Grid_home/cdata/host_name*.

- To view the contents of the OLR backup file:

```
ocrdump -local -backupfile olr_backup_file_name
```

- To change the OLR backup location:

```
ocrconfig -local -backuploc new_olr_backup_path
```

- To restore OLR:

```
# crsctl stop crs
# ocrconfig -local -restore file_name
# ocrcheck -local
# crsctl start crs
$ cluvfy comp olr
```

Upgrading and Downgrading the Oracle Cluster Registry Configuration

When you upgrade Oracle Clusterware, it automatically runs the `ocrconfig -upgrade` command. To downgrade, follow the downgrade instructions for each component and also downgrade OCR using the `ocrconfig -downgrade` command. If you are upgrading OCR, then you can use the OCRCHECK utility to verify the integrity of OCR.

Managing Voting Files

This section includes the following topics for managing voting files in your cluster:

- [Storing Voting Files on Oracle ASM](#)
- [Backing Up Voting Files](#)
- [Restoring Voting Files](#)
- [Adding_ Deleting_ or Migrating Voting Files](#)

 **Caution:**

The `dd` commands used to back up and recover voting files in previous versions of Oracle Clusterware are not supported in Oracle Clusterware 12c. Restoring voting files that were copied using `dd` or `cp` commands can prevent the Oracle Clusterware 12c stack from coming up. Use the backup and restore procedures described in this chapter to ensure proper voting file functionality.

 **Note:**

- Voting file management requires a valid and working OCR. Before you add, delete, replace, or restore voting files, run the `ocrcheck` command as `root`. If OCR is not available or it is corrupt, then you must restore OCR as described in "Restoring Oracle Cluster Registry".
- If you upgrade from a previous version of Oracle Clusterware to Oracle Clusterware 12c and you want to store voting files in an Oracle ASM disk group, then you must set the **ASM Compatibility** (`COMPATIBLE.ASM`) compatibility attribute to `12.1.0.0`.

Related Topics

- [Restoring Oracle Cluster Registry](#)
A procedure for OCR restoration for your platform.
- *Oracle Automatic Storage Management Administrator's Guide*

Storing Voting Files on Oracle ASM

If you choose to store your voting files in Oracle ASM, then Oracle ASM stores all the voting files for the cluster in the disk group you choose.

Oracle ASM manages voting files differently from other files that it stores. You cannot use voting files stored in Oracle ASM and voting files not stored in Oracle ASM in the same cluster.

Once you configure voting files on Oracle ASM, you can only make changes to the voting files' configuration using the `crsctl replace votedisk` command. This is true even in cases where there are no working voting files. Despite the fact that `crsctl query css votedisk` reports zero vote disks in use, Oracle Clusterware remembers the fact that Oracle ASM was in use and the `replace` verb is required. Only after you use the `replace` verb to move voting files back to non-Oracle ASM storage are the verbs `add css votedisk` and `delete css votedisk` again usable.

The number of voting files you can store in a particular Oracle ASM disk group depends upon the redundancy of the disk group.

By default, Oracle ASM puts each voting file in its own failure group within the disk group. A failure group is a subset of the disks in a disk group. Failure groups define disks that share components, such that if one fails then other disks sharing the component might also fail. An example of what you might define as a failure group

would be a set of SCSI disks sharing the same SCSI controller. Failure groups are used to determine which Oracle ASM disks to use for storing redundant data. For example, if two-way mirroring is specified for a file, then redundant copies of file extents must be stored in separate failure groups.

The redundancy level that you choose for the Oracle ASM disk group determines how Oracle ASM mirrors files in the disk group, and determines the number of disks and amount of disk space that you require. If the voting files are in a disk group, then the disk groups that contain Oracle Clusterware files (OCR and voting files) have a higher minimum number of failure groups than other disk groups because the voting files are stored in quorum failure groups.

A quorum failure group is a special type of failure group that is used to store the Oracle Clusterware voting files. The quorum failure group is used to ensure that a quorum of the specified failure groups are available. When Oracle ASM mounts a disk group that contains Oracle Clusterware files, the quorum failure group is used to determine if the disk group can be mounted if there is a loss of one or more failure groups. Disks in the quorum failure group do not contain user data, therefore a quorum failure group is not considered when determining redundancy requirements in respect to storing user data.

An Oracle ASM flex disk group is a disk group type that supports Oracle ASM file groups and quota groups. In general, a flex disk group enables users to manage storage at the granularity of the database, in addition to at the disk group level.

Redundancy levels include:

- **External redundancy:** An external redundancy disk group requires a minimum of one disk device. The effective disk space in an external redundancy disk group is the sum of the disk space in all of its devices.

Because Oracle ASM does not mirror data in an external redundancy disk group, Oracle recommends that you use external redundancy with storage devices such as RAID, or other similar devices that provide their own data protection mechanisms.

- **Normal redundancy:** A normal redundancy disk group requires a minimum of two disk devices (or two failure groups). The effective disk space in a normal redundancy disk group is half the sum of the disk space in all of its devices.

For Oracle Clusterware files, a normal redundancy disk group requires a minimum of three disk devices (two of the three disks are used by failure groups and all three disks are used by the quorum failure group) and provides three voting files and one OCR and mirror of the OCR. When using a normal redundancy disk group, the cluster can survive the loss of one failure group.

- **High redundancy:** In a high redundancy disk group, Oracle ASM uses three-way mirroring to increase performance and provide the highest level of reliability. A high redundancy disk group requires a minimum of three disk devices (or three failure groups). The effective disk space in a high redundancy disk group is one-third the sum of the disk space in all of its devices.

For Oracle Clusterware files, a high redundancy disk group requires a minimum of five disk devices (three of the five disks are used by failure groups and all five disks are used by the quorum failure group) and provides five voting files and one OCR and two mirrors of the OCR. With high redundancy, the cluster can survive the loss of two failure groups.

Using the `crsctl replace votedisk` command, you can move a given set of voting files from one Oracle ASM disk group into another, or onto a certified file system. If

you move voting files from one Oracle ASM disk group to another, then you can change the number of voting files by placing them in a disk group of a different redundancy level as the former disk group.

 **Note:**

- You cannot directly influence the number of voting files in one disk group.
- You cannot use the `crsctl add | delete votedisk` commands on voting files stored in Oracle ASM disk groups because Oracle ASM manages the number of voting files according to the redundancy level of the disk group.
- You cannot add a voting file to a cluster file system if the voting files are stored in an Oracle ASM disk group. Oracle does not support having voting files in Oracle ASM and directly on a cluster file system for the same cluster at the same time.

Related Topics

- [Oracle Automatic Storage Management Administrator's Guide](#)
- [Adding, Deleting, or Migrating Voting Files](#)

Backing Up Voting Files

Oracle Clusterware automatically backs up voting file data in OCR as part of any configuration change and automatically restores the data to any voting file you add.

If all voting files are corrupted, however, you can restore them as described in "Restoring Voting Files".

Related Topics

- [Restoring Voting Files](#)

Restoring Voting Files

If all of the voting files are corrupted, then you can restore them, as follows:

1. Restore OCR as described in "Restoring Oracle Cluster Registry", if necessary.
This step is necessary only if OCR is also corrupted or otherwise unavailable, such as if OCR is on Oracle ASM and the disk group is no longer available.
2. Run the following command as `root` from only one node to start the Oracle Clusterware stack in exclusive mode, which does not require voting files to be present or usable:

```
# crsctl start crs -excl
```


3. Run the `crsctl query css votedisk` command to retrieve the list of voting files currently defined, similar to the following:

```
$ crsctl query css votedisk
-- -----
## STATE   File Universal Id                        File Name Disk group
 1. ONLINE 6f57843d89464c46ea747362e8a3fa43 (/dev/sdb1) [DATA]
 2. ONLINE 7c54856e98474f61bf349401e7c9fb95 (/dev/sdc1) [DATA]
 3. ONLINE 9c46232b76234f61fc934673d5c8ec59 (/dev/sdd1) [DATA]
```

This list may be empty if all voting files were corrupted, or may have entries that are marked as status 3 or OFF.

4. Depending on where you store your voting files, do one of the following:
 - If the voting files are stored in Oracle ASM, then run the following command to migrate the voting files to the Oracle ASM disk group you specify:

```
crsctl replace votedisk +asm_disk_group
```

The Oracle ASM disk group to which you migrate the voting files must exist in Oracle ASM. You can use this command whether the voting files were stored in Oracle ASM or some other storage device.

- If you did not store voting files in Oracle ASM, then run the following command using the File Universal Identifier (FUID) obtained in the previous step:

```
$ crsctl delete css votedisk FUID
```

Add a voting file, as follows:

```
$ crsctl add css votedisk path_to_voting_disk
```

5. Stop the Oracle Clusterware stack as `root`:

```
# crsctl stop crs
```

Note:

If the Oracle Clusterware stack is running in exclusive mode, then use the `-f` option to force the shutdown of the stack.

6. Restart the Oracle Clusterware stack in normal mode as `root`:

```
# crsctl start crs
```

Related Topics

- [Restoring Oracle Cluster Registry](#)
A procedure for OCR restoration for your platform.
- *Oracle Automatic Storage Management Administrator's Guide*

Adding, Deleting, or Migrating Voting Files

You can add, remove, and migrate voting files after you install Oracle Clusterware. Note that the commands you use to do this are different, depending on whether your voting files are located in Oracle ASM, or are located in another storage option.

Modifying Voting Files that are Stored in Oracle ASM

- To display the voting file FUID and file path of each current voting file, run the `crsctl query css votedisk` command to display output similar to the following:

```
$ crsctl query css votedisk
--  -----
##  STATE      File Universal Id                File Name Disk group
  1.  ONLINE    6f57843d89464c46ea747362e8a3fa43 (/dev/sdb1) [DATA]
  2.  ONLINE    7c54856e98474f61bf349401e7c9fb95 (/dev/sdc1) [DATA]
  3.  ONLINE    9c46232b76234f61fc934673d5c8ec59 (/dev/sdd1) [DATA]
```

This command returns a disk sequence number, the status of the disk, the FUID, the path of the disk, and the name of the Oracle ASM disk group on which the disk is stored.

- To migrate voting files from Oracle ASM to an alternative storage device, specify the path to the non-Oracle ASM storage device with which you want to replace the Oracle ASM disk group using the following command:

```
$ crsctl replace votedisk path_to_voting_disk
```

You can run this command on any node in the cluster.

- To replace all voting files not stored in Oracle ASM with voting files managed by Oracle ASM in an Oracle ASM disk group, run the following command:

```
$ crsctl replace votedisk +asm_disk_group
```

Modifying Voting Files that are Not Stored on Oracle ASM

- To display the voting file FUID and file path of each current voting file, run the following command:

```
$ crsctl query css votedisk
##  STATE      File Universal Id                File Name Disk group
--  -----
  1.  ONLINE    7c54856e98474f61bf349401e7c9fb95 (/cfs/host09_vd3) []
```

This command returns a disk sequence number, the status of the disk, the FUID, and the path of the disk and no name of an Oracle ASM disk group.

- To add one or more voting files, run the following command, replacing the `path_to_voting_disk` variable with one or more space-delimited, complete paths to the voting files you want to add:

```
$ crsctl add css votedisk path_to_voting_disk [...]
```

- To replace voting file A with voting file B, you must add voting file B, and then delete voting file A. To add a new disk and remove the existing disk, run the following command, replacing the `path_to_voting_diskB` variable with the fully qualified path name of voting file B:

```
$ crsctl add css votedisk path_to_voting_diskB -purge
```

The `-purge` option deletes existing voting files.

- To remove a voting file, run the following command, specifying one or more space-delimited, voting file FUIDs or comma-delimited directory paths to the voting files you want to remove:

```
$ crsctl delete css votedisk {FUID | path_to_voting_disk[...]}
```

Note:

If the cluster is down and cannot restart due to lost voting files, then you must start CSS in exclusive mode by running the following command, as root:

```
# crsctl start crs -excl
```

After you start CSS in exclusive mode, you can replace the voting file, as follows:

```
# crsctl replace votedisk path_to_voting_disk
```

Migrating Voting Files to Oracle ASM

To migrate voting files to Oracle ASM, specify the Oracle ASM disk group name in the following command:

```
$ crsctl replace votedisk +asm_disk_group
```

You can run this command on any node in the cluster.

Verifying the Voting File Location

After modifying the voting file, verify the voting file location, as follows:

```
$ crsctl query css votedisk
```

Related Topics

- [crsctl query css votedisk](#)

7

Adding and Deleting Cluster Nodes

Describes how to add nodes to an existing cluster, and how to delete nodes from clusters.

This chapter provides procedures for these tasks for Linux, UNIX, and Windows systems.

Note:

- Unless otherwise instructed, perform all add and delete node steps as the user that installed Oracle Clusterware.
- You can use Fleet Patching and Provisioning to added and delete cluster nodes.
- Oracle recommends that you use the cloning procedure described in "Cloning Oracle Clusterware" to create clusters.

This chapter includes the following topics:

- [Prerequisite Steps for Adding Cluster Nodes](#)
- [Adding and Deleting Cluster Nodes on Linux and UNIX Systems](#)
- [Adding and Deleting Cluster Nodes on Windows Systems](#)

Related Topics

- [Cloning Oracle Clusterware](#)

Prerequisite Steps for Adding Cluster Nodes

Note:

Ensure that you perform the preinstallation tasks listed in *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux* before adding a node to a cluster.

Do not install Oracle Clusterware. The software is copied from an existing node when you add a node to the cluster.

Complete the following steps to prepare nodes to add to the cluster:

1. Make physical connections.

Connect the nodes' hardware to the network infrastructure of your cluster. This includes establishing electrical connections, configuring network interconnects, configuring shared disk subsystem connections, and so on. See your hardware vendor documentation for details about this step.

2. Install the operating system.

Install a cloned image of the operating system that matches the operating system on the other nodes in your cluster. This includes installing required service patches, updates, and drivers. See your operating system vendor documentation for details about this process.

 **Note:**

Oracle recommends that you use a cloned image. However, if the installation fulfills the installation requirements, then install the operating system according to the vendor documentation.

3. Create Oracle users.

You must create all Oracle users on the new node that exist on the existing nodes. For example, if you are adding a node to a cluster that has two nodes, and those two nodes have different owners for the Oracle Grid Infrastructure home and the Oracle home, then you must create those owners on the new node, even if you do not plan to install an Oracle home on the new node.

 **Note:**

Perform this step only for Linux and UNIX systems.

As `root`, create the Oracle users and groups using the same user ID and group ID as on the existing nodes.

4. Ensure that SSH is configured on the node.

 **Note:**

SSH is configured when you install Oracle Clusterware 12c. If SSH is not configured, then see *Oracle Grid Infrastructure Installation and Upgrade Guide* for information about configuring SSH.

5. Verify the hardware and operating system installations with the Cluster Verification Utility (CVU).

After you configure the hardware and operating systems on the nodes you want to add, you can run the following commands to verify that the nodes you want to add are reachable by other nodes in the cluster. You can also use this command to verify user equivalence to all given nodes from the local node, node connectivity among all of the given nodes, accessibility to shared storage from all of the given nodes, and so on.

- a. From the `Grid_home/bin` directory on an existing node, run the CVU command to obtain a detailed comparison of the properties of the reference node with all of the other nodes that are part of your current cluster environment. Replace `ref_node` with the name of a node in your existing cluster against which you want CVU to compare the nodes to be added. Specify a comma-delimited list of nodes after the `-n` option. In the following example, `orainventory_group` is the name of the Oracle Inventory group, and `osdba_group` is the name of the OSDBA group:

```
$ cluvfy comp peer [-refnode ref_node] -n node_list  
[-orainv orainventory_group] [-osdba osdba_group] [-verbose]
```

- b. Ensure that the Grid Infrastructure Management Repository has at least an additional 500 MB of space for each node added above four, as follows:

```
$ oclumon manage -get represize
```

Add additional space, if required, as follows:

```
$ oclumon manage -repos changerepossize total_in_MB
```

 **See Also:**

Oracle Autonomous Health Framework User's Guide for more information about using OCLUMON

 **Note:**

For the reference node, select a cluster node against which you want CVU to compare, for example, the nodes that you want to add that you specify with the `-n` option.

After completing the procedures in this section, you are ready to add the nodes to the cluster.

 **Note:**

Avoid changing host names after you complete the Oracle Clusterware installation, including adding or deleting domain qualifications. Nodes with changed host names must be deleted from the cluster and added back with the new name.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*
- *Oracle Autonomous Health Framework User's Guide*

Adding and Deleting Cluster Nodes on Linux and UNIX Systems

Add or delete cluster nodes on Linux and UNIX systems.

The procedure in the section for adding nodes assumes that you have performed the steps in the "Prerequisite Steps for Adding Cluster Nodes" section.

The last step of the node addition process includes extending the Oracle Clusterware home from an Oracle Clusterware home on an existing node to the nodes that you want to add.

This section includes the following topics:

- [Adding a Cluster Node on Linux and UNIX Systems](#)
- [Deleting a Cluster Node on Linux and UNIX Systems](#)



Note:

Beginning with Oracle Clusterware 11g release 2 (11.2), Oracle Universal Installer defaults to silent mode when adding nodes.

Related Topics

- [Prerequisite Steps for Adding Cluster Nodes](#)

Adding a Cluster Node on Linux and UNIX Systems

There are two methods that you can use to add a node to your cluster.

Using Fleet Patching and Provisioning to Add a Node

If you have a Fleet Patching and Provisioning Server, then you can use Fleet Patching and Provisioning to add a node to a cluster with one command, as shown in the following example:

```
$ rhpctl addnode gihome -client rhpclient -newnodes  
clientnode2:clientnode2-vip -root
```

The preceding example adds a node named `clientnode2` with VIP `clientnode2-vip` to the Fleet Patching and Provisioning Client named `rhpclient`, using root credentials (login for the node you are adding).

Using Oracle Grid Infrastructure Installer to Add a Node

If you do not want to use Fleet Patching and Provisioning to add a node to the cluster, then you can use the Oracle Grid Infrastructure installer to accomplish the task.

 **Note:**

You can use the `$Oracle_home/install/response/gridSetup.rsp` template to create a response file to add nodes using the Oracle Grid Infrastructure Installer for non-interactive (silent mode) operation.

To add a node to the cluster using the Oracle Grid Infrastructure installer

1. Run `./gridSetup.sh` to start the installer.
2. On the Select Configuration Option page, select **Add more nodes to the cluster**.
3. On the Cluster Node Information page, click **Add...** to provide information for nodes you want to add.
4. When the verification process finishes on the Perform Prerequisite Checks page, check the summary and then click **Install**.
5. If prompted, then run the `oraInstRoot.sh` script as root to populate the `/etc/oraInst.loc` file with the location of the central inventory. For example:

```
# /opt/oracle/oraInventory/oraInstRoot.sh
```

6. If you have an Oracle RAC or Oracle RAC One Node database configured on the cluster and you have a local Oracle home, then do the following to extend the Oracle database home to node3:
 - a. Navigate to the `Oracle_home/addnode` directory on node1 and run the `addnode.sh` script as the user that installed Oracle RAC using the following syntax:

```
$ ./addnode.sh "CLUSTER_NEW_NODES={node3}"
```

- b. Run the `Oracle_home/root.sh` script on node3 as root, where `Oracle_home` is the Oracle RAC home.

If you have a shared Oracle home that is shared using Oracle Automatic Storage Management Cluster File System (Oracle ACFS), then do the following to extend the Oracle database home to node3:

- a. Run the `Grid_home/root.sh` script on node3 as root, where `Grid_home` is the Oracle Grid Infrastructure home.
- b. Run the following command as the user that installed Oracle RAC from the `Oracle_home/oui/bin` directory on the node you are adding to add the Oracle RAC database home:

```
$ ./runInstaller -attachHome ORACLE_HOME="ORACLE_HOME"
"CLUSTER_NODES={node3}"
LOCAL_NODE="node3" ORACLE_HOME_NAME="home_name" -cfs
```

- c. Navigate to the `Oracle_home/addnode` directory on node1 and run the `addnode.sh` script as the user that installed Oracle RAC using the following syntax:

```
$ ./addnode.sh -noCopy "CLUSTER_NEW_NODES={node3}"
```


 **Note:**

Use the `-noCopy` option because the Oracle home on the destination node is already fully populated with software.

If you have a shared Oracle home on a shared file system that *is not* Oracle ACFS, then you must first create a mount point for the Oracle RAC database home on the target node, mount and attach the Oracle RAC database home, and update the Oracle Inventory, as follows:

- a. Run the `srvctl config database -db db_name` command on an existing node in the cluster to obtain the mount point information.
- b. Run the following command as `root` on `node3` to create the mount point:

```
# mkdir -p mount_point_path
```

- c. Mount the file system that hosts the Oracle RAC database home.
- d. Run the following command as the user that installed Oracle RAC from the `Oracle_home/oui/bin` directory on the node you are adding to add the Oracle RAC database home:

```
$ ./runInstaller -attachHome ORACLE_HOME="ORACLE_HOME" "CLUSTER
_NODES={local_node_name}" LOCAL_NODE="node_name"
ORACLE_HOME_NAME="home_name" -cfs
```

Navigate to the `Oracle_home/addnode` directory on `node1` and run the `addnode.sh` script as the user that installed Oracle RAC using the following syntax:

```
$ ./addnode.sh -noCopy "CLUSTER_NEW_NODES={node3}"
```

 **Note:**

Use the `-noCopy` option because the Oracle home on the destination node is already fully populated with software.

 **Note:**

After running `addnode.sh`, ensure the `Grid_home/network/admin/samples` directory has permissions set to 750.

7. Run the `Grid_home/root.sh` script on the `node3` as `root` and run the subsequent script, as instructed.

 **Note:**

- If you ran the `root.sh` script in the previous step, then you do not need to run it again.
- If you have a policy-managed database, then you must ensure that the Oracle home is cloned to the new node before you run the `root.sh` script.
- If you have any administrator-managed database instances configured on the nodes which are going to be added to the cluster, then you must extend the Oracle home to the new node before you run the `root.sh` script.

Alternatively, remove the administrator-managed database instances using the `srvctl remove instance` command.

8. Start the Oracle ACFS resource on the new node by running the following command as `root` from the `Grid_home/bin` directory:

```
# srvctl start filesystem -device volume_device_name -node node3
```

 **Note:**

Ensure the Oracle ACFS resources, including Oracle ACFS registry resource and Oracle ACFS file system resource where the Oracle home is located, are online on the newly added node.

9. Run the following CVU command as the user that installed Oracle Clusterware to check cluster integrity. This command verifies that any number of specified nodes has been successfully added to the cluster at the network, shared storage, and clusterware levels:

```
$ cluvfy stage -post nodeadd -n node3 [-verbose]
```

Related Topics

- [Oracle Fleet Patching and Provisioning](#)
Oracle Fleet Patching and Provisioning is a software lifecycle management method for provisioning and maintaining Oracle homes.
- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)
- [cluvfy stage \[-pre | -post\] nodeadd](#)
- [Oracle Real Application Clusters Administration and Deployment Guide](#)
- [Oracle Real Application Clusters Administration and Deployment Guide](#)

Deleting a Cluster Node on Linux and UNIX Systems

Delete a node from a cluster on Linux and UNIX systems.

Note:

- You can remove the Oracle RAC database instance from the node before removing the node from the cluster but this step is not required. If you do not remove the instance, then the instance is still configured but never runs. Deleting a node from a cluster does not remove a node's configuration information from the cluster. The residual configuration information does not interfere with the operation of the cluster.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for more information about deleting an Oracle RAC database instance

- If you delete the last node of a cluster that is serviced by GNS, then you must delete the entries for that cluster from GNS.
- If you have nodes in the cluster that are unpinned, then Oracle Clusterware ignores those nodes after a time and there is no need for you to remove them.
- If one creates node-specific configuration for a node (such as disabling a service on a specific node, or adding the node to the candidate list for a server pool) that node-specific configuration is not removed when the node is deleted from the cluster. Such node-specific configuration must be removed manually.
- Voting files are automatically backed up in OCR after any changes you make to the cluster.
- When you want to delete a non-Hub Node from an Oracle Flex Cluster, you need only complete steps 1 through 4 of this procedure.

To delete a node from a cluster:

1. Ensure that *Grid_home* correctly specifies the full directory path for the Oracle Clusterware home on each node, where *Grid_home* is the location of the installed Oracle Clusterware software.
2. Run the following command as either *root* or the user that installed Oracle Clusterware to determine whether the node you want to delete is active and whether it is pinned:

```
$ olsnodes -s -t
```

If the node is pinned, then run the `crsctl unpin css` command. Otherwise, proceed to the next step.

3. On the node that you are deleting, depending on whether you have a shared or local Oracle home, complete one of the following procedures as the user that installed Oracle Clusterware:

- For a local home, deinstall the Oracle Clusterware home from the node that you want to delete, as follows, by running the following command, where *Grid_home* is the path defined for the Oracle Clusterware home:

```
$ Grid_home/deinstall/deinstall -local
```

Caution:

- If you do not specify the `-local` flag, then the command removes the Oracle Grid Infrastructure home from every node in the cluster.
- If you cut and paste the preceding command, then paste it into a text editor before pasting it to the command line to remove any formatting this document might contain.

Note:

Alternatively, after you configure Oracle Grid Infrastructure, if you want to delete any node from the grid infrastructure, then you can do so by running `Grid_home/gridSetup.sh`, selecting **Remove nodes from the cluster** and following the prompts.

- If you have a shared home, then run the following commands in the following order on the node you want to delete.

Run the following command to deconfigure Oracle Clusterware:

```
$ Grid_home/crs/install/rootcrs.sh -deconfig -force
```

Run the following command from the `Grid_home/oui/bin` directory to detach the Grid home:

```
$ ./runInstaller -detachHome ORACLE_HOME=Grid_home -silent -local
```

Manually delete any configuration files, as prompted by the installation utility.

4. From any node that you are *not deleting*, run the following command from the `Grid_home/bin` directory as `root` to delete the node from the cluster:

```
# crsctl delete node -n node_to_be_deleted
```

5. Run the following CVU command to verify that the specified nodes have been successfully deleted from the cluster:

```
$ cluvfy stage -post nodedel -n node_list [-verbose]
```

6. If you remove a cluster node on which Oracle Clusterware is down, then determine whether the VIP for the deleted node still exists, as follows:

```
$ srvctl config vip -node deleted_node_name
```

If the VIP still exists, then delete it, as follows:

```
$ srvctl stop vip -node deleted_node_name  
$ srvctl remove vip -vip deleted_vip_name
```

Using Fleet Patching and Provisioning to Delete a Node

Alternatively, you can use Fleet Patching and Provisioning to delete a node from a cluster with one command, as shown in the following example:

```
$ rhpctl deletenode gihome -client rhpcclient -node clientnode2 -root
```

The preceding example deletes a node named `clientnode2` from the Fleet Patching and Provisioning Client named `rhpcclient`, using root credentials (login for the node you are deleting).

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*

- [cluvfy stage -post nodedel](#)

- [Oracle Fleet Patching and Provisioning](#)

Oracle Fleet Patching and Provisioning is a software lifecycle management method for provisioning and maintaining Oracle homes.

Adding and Deleting Cluster Nodes on Windows Systems

This section explains cluster node addition and deletion on Windows systems. This section includes the following topics:

- [Adding a Node to a Cluster on Windows Systems](#)
- [Deleting a Cluster Node on Windows Systems](#)

See Also:

Oracle Grid Infrastructure Installation and Upgrade Guide for Microsoft Windows x64 (64-Bit) for more information about deleting an entire cluster

Adding a Node to a Cluster on Windows Systems

Ensure that you complete the prerequisites listed in "Prerequisite Steps for Adding Cluster Nodes" before adding nodes.

This procedure describes how to add a node to your cluster. This procedure assumes that:

- There is an existing cluster with two nodes named `node1` and `node2`
- You are adding a node named `node3`
- You have successfully installed Oracle Clusterware on `node1` and `node2` in a local home, where `Grid_home` represents the successfully installed home

 **Note:**

Do not use the procedures described in this section to add cluster nodes in configurations where the Oracle database has been upgraded from Oracle Database 10g release 1 (10.1) on Windows systems.

To add a node:

1. Verify the integrity of the cluster and `node3`:

```
C:\>cluvfy stage -pre nodeadd -n node3 [-fixup] [-verbose]
```

You can specify the `-fixup` option and a directory into which CVU prints instructions to fix the cluster or node if the verification fails.

2. On `node1`, go to the `Grid_home\addnode` directory and run the `addnode.bat` script, as follows:

```
C:\>addnode.bat "CLUSTER_NEW_NODES={node3}"  
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={node3-vip}"
```

3. Run the following command on the new node:

```
C:\>Grid_home\crs\config\gridconfig.bat
```

4. The following steps are required *only* if you have database homes configured to use Oracle ACFS:

- a. For each database configured to use Oracle ACFS, run the following command from the Oracle RAC database home:

```
C:\>ORACLE_HOME/bin/srvctl stop database -db database_unique_name
```

 **Note:**

Run the `srvctl config database` command to list all of the databases configured with Oracle Clusterware. Use the `srvctl config database -db database_unique_name` to find the database details. If the `ORACLE_HOME` path leads to the Oracle ACFS mount path, then the database uses Oracle ACFS. Use the command output to find the database instance name configured to run on the newly added node.

- b. Use Windows Server Manager Control to stop and delete services.

- c. For each of the databases and database homes collected in the first part of this step, run the following command:

```
C:\> ORACLE_HOME/bin/srvctl start database -db database_unique_name
```

5. Run the following command to verify the integrity of the Oracle Clusterware components on all of the configured nodes, both the preexisting nodes and the nodes that you have added:

```
C:\>cluvfy stage -post crsinst -n all [-verbose]
```

After you complete the procedure in this section for adding nodes, you can optionally extend Oracle Database with Oracle RAC components to the new nodes, making them members of an existing Oracle RAC database.

See Also:

Oracle Real Application Clusters Administration and Deployment Guide for more information about extending Oracle Database with Oracle RAC to new nodes

Creating the OraMTS Service for Microsoft Transaction Server

Oracle Services for Microsoft Transaction Server (OraMTS) permit Oracle databases to be used as resource managers in Microsoft application-coordinated transactions. OraMTS acts as a proxy for the Oracle database to the Microsoft Distributed Transaction Coordinator (MSDTC). As a result, OraMTS provides client-side connection pooling and allows client components that leverage Oracle to participate in promotable and distributed transactions. In addition, OraMTS can operate with Oracle databases running on any operating system, given that the services themselves are run on Windows.

On releases earlier than Oracle Database 12c, the OraMTS service was created as part of a software-only installation. Starting with Oracle Database 12c, you must use a configuration tool to create this service.

Create the OraMTS service after adding a node or performing a software-only installation for Oracle RAC, as follows:

1. Open a command window.
2. Change directories to %ORACLE_HOME%\bin.
3. Run the `OraMTSctl` utility to create the OraMTS Service, where `host_name` is a list of nodes on which the service should be created:

```
C:\.bin> oramtsctl.exe -new -host host_name
```

Related Topics

- [Prerequisite Steps for Adding Cluster Nodes](#)
- *Oracle Real Application Clusters Administration and Deployment Guide*

- *Oracle Services for Microsoft Transaction Server Developer's Guide for Microsoft Windows*

 **See Also:**

Oracle Services for Microsoft Transaction Server Developer's Guide for Microsoft Windows for more information about OraMTS, which allows Oracle databases to be used as resource managers in distributed transactions

Deleting a Cluster Node on Windows Systems

Delete a cluster node from Windows systems.

This procedure assumes that Oracle Clusterware is installed on `node1`, `node2`, and `node3`, and that you are deleting `node3` from the cluster.

 **Note:**

- Oracle does not support using Oracle Enterprise Manager to delete nodes on Windows systems.
- If you delete the last node of a cluster that is serviced by GNS, then you must delete the entries for that cluster from GNS.
- You can remove the Oracle RAC database instance from the node before removing the node from the cluster but this step is not required. If you do not remove the instance, then the instance is still configured but never runs. Deleting a node from a cluster does not remove a node's configuration information from the cluster. The residual configuration information does not interfere with the operation of the cluster.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for more information about deleting an Oracle RAC database instance

To delete a cluster node on Windows systems:

1. Only if you have a local home, on the node you want to delete, run the following command with `-local` option to update the node list:

```
C:\>Grid_home\oui\bin\setup.exe -updateNodeList ORACLE_HOME=Grid_home
"CLUSTER_NODES=
{node_to_be_deleted}" CRS=TRUE -local
```

 **Note:**

If you are deleting a non-Hub Node from an Oracle Flex Cluster, then you do not have run this command.

2. Run the deinstall tool on the node you want to delete to deinstall and deconfigure the Oracle Clusterware home, as follows:

```
C:\Grid_home\deinstall\>deinstall.bat -local
```

 **Caution:**

- If you do not specify the `-local` flag, then the command removes the Oracle Grid Infrastructure home from every node in the cluster.
- If you cut and paste the preceding command, then paste it into a text editor before pasting it to the command line to remove any formatting this document might contain.

3. On a node that you are not deleting, run the following command:

```
C:\>Grid_home\bin\crsctl delete node -n node_to_be_deleted
```

4. Run the following CVU command to verify that the specified nodes have been successfully deleted from the cluster:

```
C:\>cluvfy stage -post nodedel -n node_list [-verbose]
```

5. If you remove a cluster node on which Oracle Clusterware is down, then determine whether the VIP for the deleted node still exists, as follows:

```
C:\> ORACLE_HOME/bin/srvctl config vip -node deleted_node_name
```

If the VIP still exists, then delete it, as follows:

```
C:\> ORACLE_HOME/bin/srvctl srvctl stop vip -node deleted_node_name  
C:\> ORACLE_HOME/bin/srvctl srvctl remove vip -node deleted_node_name
```

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*
- [cluvfy stage -post nodedel](#)

8

Cloning Oracle Clusterware

This chapter describes how to clone an Oracle Grid Infrastructure home and use the cloned home to create a cluster. You perform the cloning procedures in this chapter by running scripts in silent mode. The cloning procedures are applicable to Linux and UNIX systems. Although the examples in this chapter use Linux and UNIX commands, the cloning concepts and procedures apply generally to all platforms.

Note:

This chapter assumes that you are cloning an Oracle Clusterware 12c installation configured as follows:

- No Grid Naming Service (GNS)
- No Intelligent Platform Management Interface specification (IPMI)
- Voting file and Oracle Cluster Registry (OCR) are stored in Oracle Automatic Storage Management (ASM)
- Single Client Access Name (SCAN) resolves through DNS

This chapter contains the following topics:

- [Introduction to Cloning Oracle Clusterware](#)
- [Preparing the Oracle Grid Infrastructure Home for Cloning](#)
- [Creating a Cluster by Cloning Oracle Clusterware](#)
- [Using Cloning to Add Nodes to a Cluster](#)
- [Locating and Viewing Log Files Generated During Cloning](#)

Introduction to Cloning Oracle Clusterware

Cloning is the process of copying an existing Oracle Clusterware installation to a different location and then updating the copied installation to work in the new environment. Changes made by one-off patches applied on the source Oracle Grid Infrastructure home are also present after cloning. During cloning, you run a script that replays the actions that installed the Oracle Grid Infrastructure home.

Cloning requires that you start with a successfully installed Oracle Grid Infrastructure home. You use this home as the basis for implementing a script that extends the Oracle Grid Infrastructure home to create a cluster based on the original Grid home.

Manually creating the cloning script can be error prone because you prepare the script without interactive checks to validate your input. Despite this, the initial effort is worthwhile for scenarios where you run a single script to configure tens or even hundreds of clusters. If you have only one cluster to install, then you should use the

traditional, automated and interactive installation methods, such as Oracle Universal Installer (OUI) or the Provisioning Pack feature of Oracle Enterprise Manager.

 **Note:**

Cloning is not a replacement for Oracle Enterprise Manager cloning that is a part of the Provisioning Pack. During Oracle Enterprise Manager cloning, the provisioning process simplifies cloning by interactively asking for details about the Oracle home. The interview questions cover such topics as the location to which you want to deploy the cloned environment, the name of the Oracle database home, a list of the nodes in the cluster, and so on.

The Provisioning Pack feature of Oracle Enterprise Manager Grid Control provides a framework that automates the provisioning of nodes and clusters. For data centers with many clusters, the investment in creating a cloning procedure to provision new clusters and new nodes to existing clusters is worth the effort.

The following list describes some situations in which cloning is useful:

- Cloning prepares an Oracle Grid Infrastructure home once and deploys it to many hosts simultaneously. You can complete the installation in silent mode, as a noninteractive process. You do not need to use a graphical user interface (GUI) console, and you can perform cloning from a Secure Shell (SSH) terminal session, if required.
- Cloning enables you to create an installation (copy of a production, test, or development installation) with all patches applied to it in a single step. Once you have performed the base installation and applied all patch sets and patches on the source system, cloning performs all of these individual steps as a single procedure. This is in contrast to going through the installation process to perform the separate steps to install, configure, and patch the installation on each node in the cluster.
- Installing Oracle Clusterware by cloning is a quick process. For example, cloning an Oracle Grid Infrastructure home to a cluster with more than two nodes requires a few minutes to install the Oracle software, plus a few minutes more for each node (approximately the amount of time it takes to run the `root.sh` script).
- Cloning provides a guaranteed method of accurately repeating the same Oracle Clusterware installation on multiple clusters.

A cloned installation acts the same as its source installation. For example, you can remove the cloned Oracle Grid Infrastructure home using OUI or patch it using OPatch. You can also use the cloned Oracle Grid Infrastructure home as the source for another cloning operation. You can create a cloned copy of a test, development, or production installation by using the command-line cloning scripts.

The default cloning procedure is adequate for most cases. However, you can also customize some aspects of cloning, such as specifying custom port assignments or preserving custom settings.

For example, you can specify a custom port for the listener, as follows:

```
$ export ORACLE_HOME=/u01/app/12.1.0/grid
$ $ORACLE_HOME/bin/srvctl modify listener -endpoints tcp:12345
```

The cloning process works by copying all of the files from the source Oracle Grid Infrastructure home to the destination Oracle Grid Infrastructure home. You can clone either a local (non-shared) or shared Oracle Grid Infrastructure home. Thus, any files used by the source instance that are located outside the source Oracle Grid Infrastructure home's directory structure are not copied to the destination location.

The size of the binary files at the source and the destination may differ because these files are relinked as part of the cloning operation, and the operating system patch levels may also differ between these two locations. Additionally, the number of files in the cloned home would increase because several files copied from the source, specifically those being instantiated, are backed up as part of the clone operation.

Preparing the Oracle Grid Infrastructure Home for Cloning

To prepare the source Oracle Grid Infrastructure home to be cloned, create a copy of an installed Oracle Grid Infrastructure home and then use it to perform the cloning procedure on other nodes. Use the following step-by-step procedure to prepare the copy of the Oracle Grid Infrastructure home:

- [Step 1: Install Oracle Clusterware](#)
- [Step 2: Shut Down Running Software](#)
- [Step 3: Create a Copy of the Oracle Grid Infrastructure Home](#)

Step 1: Install Oracle Clusterware

Use the detailed instructions in the *Oracle Grid Infrastructure Installation and Upgrade Guide* to perform the following steps on the source node:

1. Install Oracle Clusterware 12c. This installation puts Oracle Cluster Registry (OCR) and the voting file on Oracle Automatic Storage Management (Oracle ASM).

Note:

Either install and configure the Oracle Grid Infrastructure for a cluster or install just the Oracle Clusterware software, as described in your platform-specific *Oracle Grid Infrastructure Installation and Upgrade Guide*.

If you installed and configured Oracle Grid Infrastructure for a cluster, then you must stop Oracle Clusterware before performing the cloning procedures. If you performed a software-only installation, then you do not have to stop Oracle Clusterware.

2. Install any patches that are required (for example, an Oracle Grid Infrastructure bundle patch), if necessary.
3. Apply one-off patches, if necessary.

Related Topics

- [Oracle Grid Infrastructure Installation and Upgrade Guide](#)

Step 2: Shut Down Running Software

Before copying the source Oracle Grid Infrastructure home, shut down all of the services, databases, listeners, applications, Oracle Clusterware, and Oracle ASM instances that run on the node. Oracle recommends that you use the Server Control (SRVCTL) utility to first shut down the databases, and then the Oracle Clusterware Control (CRSCTL) utility to shut down the rest of the components.

Related Topics

- [Oracle Real Application Clusters Administration and Deployment Guide](#)
- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)

Step 3: Create a Copy of the Oracle Grid Infrastructure Home

To keep the installed Oracle Grid Infrastructure home as a working home, make a full copy of the source Oracle Grid Infrastructure home for cloning.

Tip:

When creating the copy, a best practice is to include the release number in the name of the file.

You can use one of the following methods to create a create a copy of the Oracle Grid Infrastructure home.

- [Method 1: Create a Copy of the Oracle Grid Infrastructure Home Using the Setup Wizard](#)
- [Method 2a: Create a Copy of the Oracle Grid Infrastructure Home by Removing Unnecessary Files from the Copy](#)
- [Method 2b: Create a Copy of the Oracle Grid Infrastructure Home Using the -X Option](#)

Method 1 is the recommended method to create the a gold image of a working Oracle Grid Infrastructure home.

You can use Method 2a or 2b to manually create copy of the Oracle Grid Infrastructure home, where *grid_home* is the original Oracle Grid Infrastructure home on the original node with all files included, and *copy_path* is the directory path to the copied Oracle Grid Infrastructure home with unnecessary files deleted.

Method 1: Create a Copy of the Oracle Grid Infrastructure Home Using the Setup Wizard

You can use the grid setup wizard to create a gold image of a working Oracle Grid Infrastructure home.

1. Run `gridSetup.sh` with the appropriate options to create a gold image.

For example, create a gold image in a specified destination:

```
./gridSetup.sh -createGoldImage -destinationLocation /tmp/my_grid_images
```

2. Run `gridSetup.sh` with the `-help` option for additional information.

See Also:

- *Oracle Grid Infrastructure Installation and Upgrade Guide* for information about using the grid setup wizard installation options for creating images

Method 2a: Create a Copy of the Oracle Grid Infrastructure Home by Removing Unnecessary Files from the Copy

1. On the source node, create a copy of the Oracle Grid Infrastructure home. To keep the installed Oracle Grid Infrastructure home as a working home, make a full copy of the source Oracle Grid Infrastructure home and remove the unnecessary files from the copy. For example, as `root` on Linux systems, run the `cp` command:

```
# cp -prf Grid_home copy_path
```

2. Delete unnecessary files from the copy.

The Oracle Grid Infrastructure home contains files that are relevant only to the source node, so you can remove the unnecessary files from the copy of the Oracle Grid Infrastructure home in the `log`, `crs/init`, `crf`, and `cdata` directories. The following example for Linux and UNIX systems shows the commands to run to remove the unnecessary files from the *copy* of the Oracle Grid Infrastructure home:

```
[root@node1 root]# cd copy_path
[root@node1 grid]# rm -rf log/host_name
[root@node1 grid]# rm -rf gpnnp/host_name
[root@node1 grid]# find gpnnp -type f -exec rm -f {} \;
[root@node1 grid]# rm -rf cfgtoollogs/*
[root@node1 grid]# rm -rf crs/init/*
[root@node1 grid]# rm -rf cdata/*
[root@node1 grid]# rm -rf crf/*
[root@node1 grid]# rm -rf network/admin/*.ora
[root@node1 grid]# rm -rf crs/install/crsconfig_params
[root@node1 grid]# rm -rf crs/install/crsngenconfig_params
[root@node1 grid]# find . -name '*.ouibak' -exec rm {} \;
[root@node1 grid]# find . -name '*.ouibak.1' -exec rm {} \;
[root@node1 grid]# rm -rf root.sh*
[root@node1 grid]# rm -rf bin/clsecho/*
[root@node1 grid]# rm -rf rdbms/audit/*
[root@node1 grid]# rm -rf rdbms/log/*
[root@node1 grid]# rm -rf inventory/backup/*
[root@node1 grid]# rm -rf oraInst.loc
```

3. Create a compressed copy of the previously copied Oracle Grid Infrastructure home using tar or gzip on Linux and UNIX systems. Ensure that the tool you use preserves the permissions and file timestamps. For example:

On Linux and UNIX systems:

```
[root@node1 root]# cd copy_path
[root@node1 grid]# tar -zcvpf /copy_path/gridHome.tgz
```

In the preceding example, the `cd` command changes the location to the copy of the Oracle Grid Infrastructure home with the unnecessary files removed that you created in the first two steps of this procedure, and the `tar` command creates a file named `gridHome.tgz`. In the `tar` command, *copy_path* represents the location of the copy of the Oracle Grid Infrastructure home.

On AIX or HPUX systems:

```
uncompress gridHome.tar.Z
tar xvf gridHome.tar
```

On Windows systems, use WinZip to create a zip file.

Method 2b: Create a Copy of the Oracle Grid Infrastructure Home Using the -X Option

1. Create a file that lists the unnecessary files in the Oracle Grid Infrastructure home. For example, list the following file names, using the asterisk (*) wildcard, in a file called `excludeFileList`:

```
Grid_home/host_name
Grid_home/log/host_name
Grid_home/gpnp/*
Grid_home/bin/clsecho/*
Grid_home/crs/init/*
Grid_home/cdata/*
Grid_home/crf/*
Grid_home/network/admin/*.ora
Grid_home/root.sh*
Grid_home/cfgtoollogs/*
Grid_home/crs/install/crsconfig_params
Grid_home/crs/install/crsngenconfig_params
Grid_home/rdbms/audit/*
Grid_home/rdbms/log/*
Grid_home/inventory/backup/*
Grid_home/oraInst.loc
*.ouibak
*.ouibak1
```

2. Use the `tar` command or Winzip to create a compressed copy of the Oracle Grid Infrastructure home. For example, on Linux and UNIX systems, run the following command to archive and compress the source Oracle Grid Infrastructure home:

```
tar cpfX - excludeFileList Grid_home | compress -fv > temp_dir/
gridHome.tar.Z
```



Note:

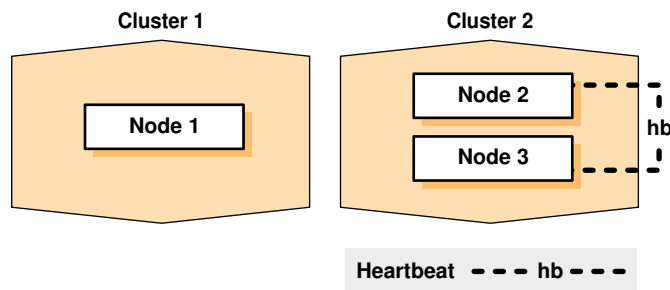
Do not use the jar utility to copy and compress the Oracle Grid Infrastructure home.

Creating a Cluster by Cloning Oracle Clusterware

This section explains how to create a cluster by cloning a successfully installed Oracle Clusterware environment and copying it to the nodes on the destination cluster. The procedures in this section describe how to use cloning for Linux, UNIX, and Windows systems. OCR and voting files are not shared between the two clusters after you successfully create a cluster from a clone.

For example, you can use cloning to quickly duplicate a successfully installed Oracle Clusterware environment to create a cluster. [Figure 8-1](#) shows the result of a cloning procedure in which the Oracle Grid Infrastructure home on Node 1 has been cloned to Node 2 and Node 3 on Cluster 2, making Cluster 2 a new two-node cluster.

Figure 8-1 Cloning to Create an Oracle Clusterware Environment



The steps to create a cluster through cloning are as follows:

- [Step 1: Prepare the New Cluster Nodes](#)
- [Step 2: Deploy the Oracle Grid Infrastructure Home](#)
- [Step 3: Run the gridSetup.sh Utility](#)

Step 1: Prepare the New Cluster Nodes

Before you clone Oracle Clusterware you must complete preinstallation steps.

On each destination node, perform the following preinstallation steps:

1. Specify the kernel parameters
2. Configure block devices for Oracle Clusterware devices
3. Ensure that you have set the block device permissions correctly
4. Use short, nondomain-qualified names for all of the names in the `/etc/hosts` file
5. Test whether the interconnect interfaces are reachable, by using the `ping` command

6. Verify that the virtual IP (VIP) addresses are not active at the start of the cloning process, by using the `ping` command (the `ping` command of the VIP address must fail)
7. (IBM AIX systems and Oracle Solaris x86-64-bit systems running vendor clusterware only) If you add a node to the cluster, then you must run the `rootpre.sh` script (located at the mount point it you install Oracle Clusterware from a DVD or in the directory where you unzip the tar file if you download the software) on the node before you add the node to the cluster

 **Note:**

Starting with Oracle Database 19c, the integration of vendor clusterware with Oracle Clusterware is deprecated, and can be desupported in a future release. For this reason, Oracle recommends that you align your next software or hardware upgrade to transition off of vendor cluster solutions.

8. To verify that your hardware and operating system environment is configured correctly, run the Cluster Verification Utility (CVU)

 **Note:**

Make sure you check your cluster with CVU before you proceed. Unlike traditional methods of installation, the cloning process does not validate your input during the preparation phase. By comparison, during the traditional method of installation using Oracle Universal Installer (OUI), various checks occur during the interview phase. Thus, if you make errors during the hardware setup or in the preparation phase, then the cloned installation fails.

Refer to your platform-specific Oracle Clusterware installation guide for the complete preinstallation checklist.

Step 2: Deploy the Oracle Grid Infrastructure Home

Before you begin the cloning procedure that is described in this section, ensure that you have completed the prerequisite tasks to create a copy of the Oracle Grid Infrastructure home, as described in the section titled "Preparing the Oracle Grid Infrastructure Home for Cloning".

1. On one of the nodes in the cluster, deploy the copy of the Oracle Grid Infrastructure home that you created in "Step 3: Create a Copy of the Oracle Grid Infrastructure Home", as follows:

For example, on Linux or UNIX systems, run commands similar to the following:

```
[root@node1 root]# mkdir -p location_of_the_copy_of_the_Grid_home
[root@node1 root]# cd location_of_the_copy_of_the_Grid_home
[root@node1 crs]# tar -zxvf /gridHome.tgz
```

In this example, *location_of_the_copy_of_the_Grid_home* represents the directory structure in which you want to install the Oracle Grid Infrastructure home, such as `/u01/app/12.1.0/grid`. Note that you can change the Grid home location as part of the process.

2. If you have not already deleted unnecessary files from the Oracle Grid Infrastructure home, then repeat step 2 in "Method 2: Create a Compressed Copy of the Oracle Grid Infrastructure Home Using the -X Option".
3. If necessary, change the ownership of all of the files in the Oracle Grid Infrastructure home to be owned by the Oracle Grid Infrastructure installation owner and by the Oracle Inventory (`oinstall` privilege) group. If the Oracle Grid Infrastructure installation owner is `oracle`, and the Oracle Inventory group is `oinstall`, then the following example shows the commands to do this on a Linux system:

```
[root@node1 crs]# chown -R oracle:oinstall /u01/app
```

When you run the preceding command on the Grid home, it clears `setuid` and `setgid` information from the Oracle binary. As expected, the command also clears `setuid` from the following binaries:

```
Grid_home/bin/extjob  
Grid_home/bin/jssu  
Grid_home/bin/oradism
```

The `setuid` information is properly set in subsequent steps.

4. It is important to remove any Oracle network files from the *Grid_home* directory on both the source and destination nodes before continuing.

Related Topics

- [Step 3: Create a Copy of the Oracle Grid Infrastructure Home](#)
- [Method 2b: Create a Copy of the Oracle Grid Infrastructure Home Using the -X Option](#)
- [Preparing the Oracle Grid Infrastructure Home for Cloning](#)
- [Locating and Viewing Log Files Generated During Cloning](#)

Step 3: Run the `gridSetup.sh` Utility

To set up the new Oracle Clusterware environment, run the `gridSetup.sh` utility in either interactive or silent mode on one node, as you would when installing Oracle Grid Infrastructure for a new cluster.

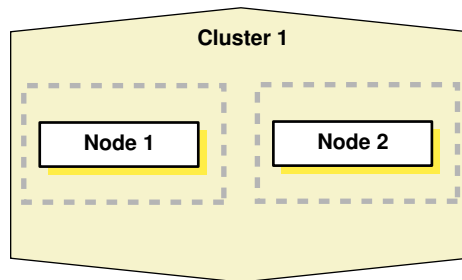
Once you launch the `$ORACLE_home/gridSetup.sh` utility, you can select to run it either in interactive mode, using the interactive interface, or in silent mode, where you provide a response file.

Using Cloning to Add Nodes to a Cluster

You can also use cloning to add nodes to a cluster.

Figure 8-2 shows the result of a cloning procedure in which the Oracle Grid Infrastructure home on Node 1 has been cloned to Node 2 in the same cluster, making it a two-node cluster. Newly added nodes to the cluster share the same OCR and voting files.

Figure 8-2 Cloning to Add Nodes to a Cluster



Using Figure 8-2 as an example, the following procedure explains how to add nodes to a cluster using cloning. In this procedure, you make a copy of the image (a clone) that you used to create Node 1, initially, to Node 2.

Prepare Node 2 as described in "Step 1: Prepare the New Cluster Nodes".

1. Deploy the Oracle Grid Infrastructure home on Node 2, as described in "Step 2: Deploy the Oracle Grid Infrastructure Home on the Destination Nodes".

Use the tar utility to create an archive of the Oracle Grid Infrastructure home on the Node 1 and copy it to Node 2. If the location of the Oracle Grid Infrastructure home on Node 1 is `$ORACLE_HOME`, then you must use this same directory as the destination location on Node 2.

2. Run the `gridSetup.sh` script and select the `Software Only` installation option.
3. This step does not apply to Windows.

In the Central Inventory directory on Node 2, run the `orainstRoot.sh` script as `root`. This script populates the `/etc/oraInst.loc` directory with the location of the central inventory. For example:

```
[root@node2 root]# /opt/oracle/oraInventory/orainstRoot.sh
```

You can run the script on more than one destination node simultaneously.

4. Run the `gridSetup.sh -noCopy` script and select `Add more nodes to cluster`.
5. On Node 2, run the `Grid_home/root.sh` script.

 **Note:**

- Ensure that you extend any database homes before you run the `root.sh` or `gridconfig.bat` scripts.
- The cluster in this example has only two nodes. When you add multiple nodes to a cluster, you can run `root.sh` concurrently on all of the nodes.

The following example is for a Linux or UNIX system. On Node 2, run the following command:

```
[root@node2 root]# Grid_home/root.sh
```

The `root.sh` script automatically configures the virtual IP (VIP) resources in the Oracle Cluster Registry (OCR).

On Windows, run the following command on Node 2:

```
C:\>Grid_home\crs\config\gridconfig.bat
```

6. Run the following cluster verification utility (CVU) command on Node 1:

```
$ cluvfy stage -post nodeadd -n destination_node_name [-verbose]
```

Related Topics

- [cluvfy stage \[-pre | -post\] nodeadd](#)

Locating and Viewing Log Files Generated During Cloning

The cloning script runs multiple tools, each of which can generate log files. After the `gridSetup.sh` utility finishes running, you can view log files to obtain more information about the status of your cloning procedures. [Table 8-1](#) lists the log files that are generated during cloning that are the key log files for diagnostic purposes.

 **Note:**

`Central_inventory` in [Table 8-1](#) refers to the Oracle Inventory directory.

Table 8-1 Cloning Log Files and their Descriptions

Log File Name and Location	Description
<code>Central_inventory/logs/cloneActiontimestamp.log</code>	Contains a detailed log of the actions that occur during the OUI part of the cloning.
<code>Central_inventory/logs/oraInstalltimestamp.err</code>	Contains information about errors that occur when OUI is running.

Table 8-1 (Cont.) Cloning Log Files and their Descriptions

Log File Name and Location	Description
<i>Central_inventory/logs/oraInstalltimestamp.out</i>	Contains other miscellaneous information.

[Table 8-2](#) lists the location of the Oracle Inventory directory for various platforms.

Table 8-2 Finding the Location of the Oracle Inventory Directory

Type of System	Location of the Oracle Inventory Directory
All UNIX computers except Linux and IBM AIX	<code>/var/opt/oracle/oraInst.loc</code>
IBM AIX and Linux	<code>/etc/oraInst.loc</code>
Windows	<code>C:\Program Files\Oracle\Inventory</code>

9

Making Applications Highly Available Using Oracle Clusterware

When an application, process, or server fails in a cluster, you want the disruption to be as short as possible, if not completely unknown to users. For example, when an application fails on a server, that application can be restarted on another server in the cluster, minimizing or negating any disruption in the use of that application. Similarly, if a server in a cluster fails, then all of the applications and processes running on that server must be able to fail over to another server to continue providing service to the users. Using the built-in `generic_application` resource type or customizable scripts and application agent programs, and resource attributes that you assign to applications and processes, Oracle Clusterware can manage all these entities to ensure high availability.

This chapter explains how to use Oracle Clusterware to start, stop, monitor, restart, and relocate applications. Oracle Clusterware is the underlying cluster solution for Oracle Real Application Clusters (Oracle RAC). The same functionality and principles you use to manage Oracle RAC databases are applied to the management of applications.

This chapter includes the following topics:

- [Oracle Clusterware Resources and Agents](#)
- [Overview of Using Oracle Clusterware to Enable High Availability](#)
- [Registering an Application as a Resource](#)
- [Managing Resources](#)
- [Managing Automatic Restart of Oracle Clusterware Resources](#)

Oracle Clusterware Resources and Agents

This section discusses the framework that Oracle Clusterware uses to monitor and manage [resources](#), to ensure high application availability.

This section includes the following topics:

- [Oracle Clusterware Resources](#)
- [Oracle Clusterware Resource Types](#)
- [Agents in Oracle Clusterware](#)
- [Action Scripts](#)
- [Building an Agent](#)
- [Registering a Resource in Oracle Clusterware](#)

Oracle Clusterware Resources

Oracle Clusterware manages applications and processes as resources that you register with Oracle Clusterware.

The number of resources you register with Oracle Clusterware to manage an application depends on the application. Applications that consist of only one process are usually represented by only one resource. For more complex applications that are built on multiple processes or components that may require multiple resources, you can create [resource groups](#).

Each resource is based on a resource type that serves as a template. You can configure how Oracle Clusterware will place an application in the cluster by specifying an explicit list of servers, or by using features such as server pools and policies. Relationships between applications and components are expressed using dependencies. Oracle Clusterware manages the application by performing operations on the resources, and the resource state represents the availability of the application.

When you register an application as a resource in Oracle Clusterware, in addition to actually adding the resource to the system, you define how Oracle Clusterware manages the application using resource attributes you ascribe to the resource. The frequency with which the resource is checked and the number of attempts to restart a resource on the same server after a failure before attempting to start it on another server (failover) are examples of resource attributes. The registration information also includes a path to an action script or application-specific action program that Oracle Clusterware calls to start, stop, check, and clean up the application.

An action script is a shell script (a batch script in Windows) that a generic script agent provided by Oracle Clusterware calls. An application-specific agent is usually a C or C++ program that calls Oracle Clusterware-provided APIs directly.

Critical Resources

Some large enterprise applications modeled as resource groups can comprise multiple resources representing application or infrastructure components. If any resource in the resource group fails, then Oracle Clusterware must fail the entire resource group over to another server in the cluster.

You can mark a resource as critical for its resource group by specifying the name of the resource in the `CRITICAL_RESOURCES` attribute of the resource group.

Related Topics

- [Oracle Clusterware Resource Reference](#)
- [Resource Groups](#)
A resource group is a container for a logically related group of resources.

Virtual Machine Resources

A virtual machine is an environment created for a running operating system, known as a guest operating system. The virtual machine displays as a window on your computer's desktop which can be displayed in full-screen mode or remotely on another computer.

A virtual machine is, essentially, a set of parameters that determines its behavior, analogous to computer system hardware. Parameters include hardware settings (such

as how much memory the virtual machine has) as well as state information (such as whether the virtual machine is currently running).

Black-box virtual machines are virtual machines whose contents are unknown to the management interface. All that is known about black-box virtual machines is the virtual hardware they contain: the number of CPUs, the amount of RAM, attached disks, and attached network interfaces. The contents of the hardware however, are unknown. For example, there may be a number of disks attached, but it is not known which operating system is installed on them, nor is it known whether the network cards are configured.

You can manage black-box Oracle virtual machines on physical hardware using Oracle Clusterware, which provides high availability and ease of management of virtual machines.

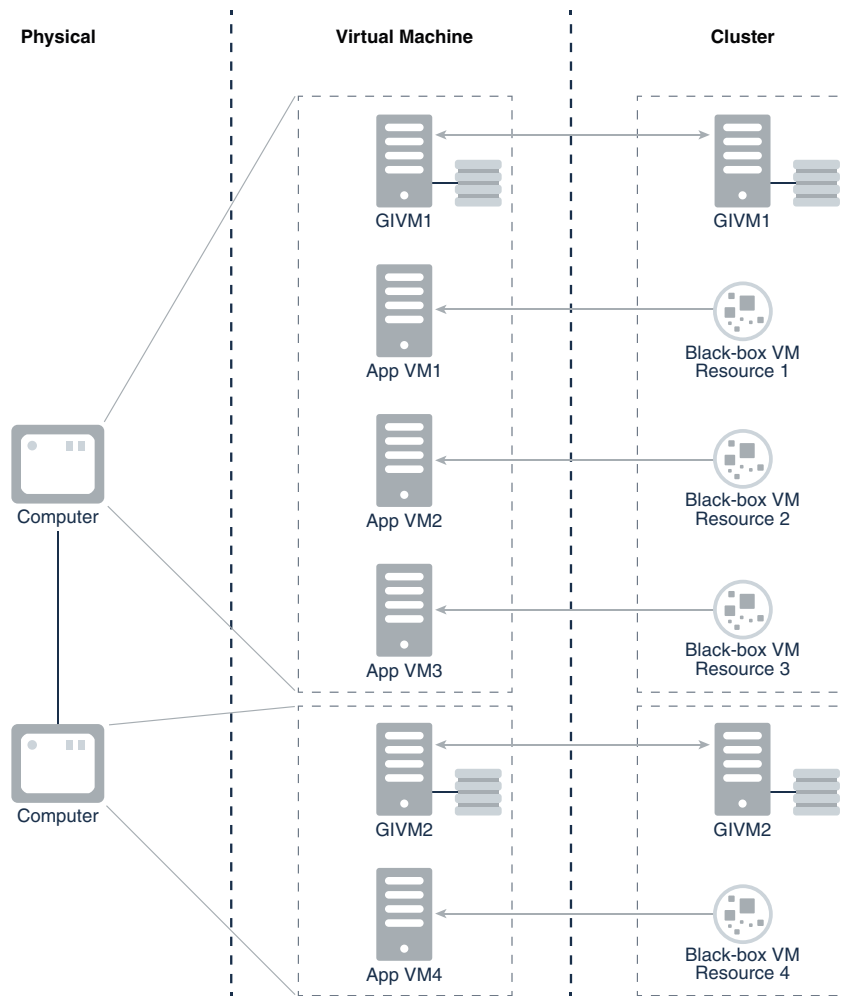
 **Note:**

This is specific to virtual machines, and does not apply to Oracle VM VirtualBox, or any other Oracle VM product.

As an example, in following figure, there are two physical computers, each of which has multiple virtual machines running on it. One of the computers, for each physical host, is an Oracle Grid Infrastructure virtual machine (GIVM).

The GIVMs, themselves, form an Oracle Clusterware cluster, and within this cluster are four black-box virtual machine Oracle Clusterware resources, each monitoring one of the non-GIVM virtual machines. The cluster is not aware of the contents of the virtual machines it is monitoring because they are black-box virtual machines. In this example, if one of the physical hosts goes down, then its GIVM would also go down, causing the GIVM to leave the cluster, which, in turn, causes the resources to fail over to the other GIVM, which starts the black-box virtual machines on the new physical host.

Figure 9-1 Highly Available Virtual Machines in Oracle Database Appliance



Virtual Machine Architecture

Oracle virtual machines consist of two parts: the virtual machine server and the virtual machine manager. The virtual machine server is a minimal operating system installed on bare hardware that uses a Xen hypervisor to manage guests. The server has an agent process, the Oracle virtual machine agent, which acts as an intermediary through which the virtual machine manager manipulates domains on the server.

The virtual machine manager is a web-based management console that is used to manage virtual machine servers and their virtual machines. The virtual machine manager requires a database as well as an Oracle WebLogic Server in order to run, and is necessary for Oracle-supported management of the Oracle virtual machine server. The management domain is supposed to be as small as possible and, therefore, the virtual machine manager may not be installed there. You must install the virtual machine manager on another host, which means either having another physical computer or manually creating a temporary virtual machine using the Xen `xm` commands.

The Oracle virtual machine manager is the sole interface for managing virtual machines. All requests are directed through it, including all APIs and utilities.

The resource type for a virtual machine resource (which is an Oracle Clusterware resource) is similar to the following:

```
ATTRIBUTE=DESCRIPTION
TYPE=string
DEFAULT_VALUE="Resource type for VM Agents"
ATTRIBUTE=AGENT_FILENAME
TYPE=string
DEFAULT_VALUE=%CRS_HOME%/bin/oraagent%CRS_EXE_SUFFIX%
ATTRIBUTE=CHECK_INTERVAL
DEFAULT_VALUE=1
TYPE=int
ATTRIBUTE=OVMM_VM_ID
TYPE=string
DEFAULT_VALUE=' '
ATTRIBUTE=OVMM_VM_NAME
TYPE=string
DEFAULT_VALUE=' '
ATTRIBUTE=VM
TYPE=strings
```

Resource Groups

A resource group is a container for a logically related group of resources.

An application is modeled as a resource group that contains the application resource and related application resources (such as WebServer), and infrastructure resources (such as disk groups and VIPs). A resource group provides a logical and intuitive entity for high availability modeling of all classes of applications.

You create resource groups using CRSCCTL, and then add resources to the resource group. A resource group provides a set of attributes that cover naming, description, and common placement and failover parameter values for the resources that are members of the resource group.

Resource Group Principles

- You create a resource group based on a resource group type.
- A resource can be member of only one resource group. You can specify a resource group for a resource when you create the resource.

If you do not specify a resource group when you create a resource, then the resource becomes a member of an automatic resource group created for that resource. You can later add the resource to a different resource group.
- Resource groups are aware of critical resources, and the state of the resource group is solely determined by the state of its critical resources.

You can remove a non-critical resource from a resource group (subject to dependency checks) and, at a later time, add it to another resource group.
- Resource groups have cardinality to specify the number of instances of the resource group that can simultaneously run in the cluster.
- All member resources of a running resource group instance are located on the same server.

- Oracle Clusterware restarts a resource group in the event of failure and then relocates the resource group to another server in the event of local restart failures.

Automatic Resource Groups

If you create a resource without specifying a resource group, then Oracle Clusterware implicitly and automatically adds the resource to a resource group with the same name as the resource.

An automatic resource group is created for each resource that is not explicitly added to a resource group. You can create resources without using resource groups and work with Oracle Clusterware without disruption. Using resource groups, however, enables you to define relationships to infrastructure and application resources (through automatic resource groups) created by SRVM or other existing utilities.

An automatic resource group is solely described by the resource that it has been created for, and cannot be modified by an administrator. Resources that you create without specifying a resource group can be added to a resource group at a later time. Oracle Clusterware deletes the automatic resource group to which the resource belongs when the resource has been explicitly added to a resource group.

Resource Group Management

- You can add a resource to a resource group when you create the resource.
- You can explicitly add a resource that belongs to its automatic resource group to another resource group. The resource must be OFFLINE when you add the resource to a group that is either ONLINE or OFFLINE.
- You can remove a non-critical resource from a resource group (OFFLINE or ONLINE) as long as no other resource in the group depends on it. The resource you remove then becomes a member of its automatic resource group. At a later time, you can add this resource to another group.
- You can delete a non-critical resource, thereby removing the resource from the resource group and deleting it from Oracle Clusterware. You cannot delete a critical resource of a resource group, unless you first update the critical resources list of the resource group to unmark the resource as critical.
- A resource group is empty when it is initially created and also becomes empty when each resource in the group has been removed. An empty resource group cannot be started and its state will be always be OFFLINE.

Share Resources

In various Oracle Clusterware deployments, there are components, such as file systems, that multiple applications share. A single Oracle ACFS resource, for example, cannot be a member of multiple resource groups that make use of the filesystem because, by definition, a resource can be a member of only one resource group.

For these types of resources, Oracle recommends that you put them in their own individual resource groups, either explicit or automatic, and configure appropriate dependencies from the application resource groups to these shared resource groups. In this manner, multiple applications can share components.

Critical Resources

You can have a large-enterprise type application that is modeled as a resource group that contains multiple resources corresponding to application or infrastructure components. If any of the resources in such a resource group fails, then Oracle Clusterware fails the entire resource group over to another server in the cluster. Some resources in the resource group, however, are not critical to the application and would not necessarily require failing over the entire resource group, which would cause an unnecessary disruption in the running of the instance.

You can define certain resources within a resource group as *critical* (by specifying the name of the resource in the CRITICAL_RESOURCES list attribute of the resource group) and, should any of those resources fail, then Oracle Clusterware will fail the resource group over to another server in the cluster.

Further, the state of a resource group is determined by the state of its critical resources. Non-critical resources do not affect the state of the resource group nor can they trigger failover of the resource group. A resource group must have at least one critical resource before the resource group can be started and brought online.

You can specify individual resources in a resource group as critical or you can specify a resource type as critical, which would make all resources of that particular type critical. For example:

```
CRITICAL_RESOURCES="r1 r2 r3"
```

The preceding example lists three, space-delimited resources marked as critical.

```
CRITICAL_RESOURCES="appvip type:ora.export.type"
```

The preceding example lists a particular resource type as critical, thereby making any resource of this type a critical resource.

When you create a resource group or remove its members, it is empty and, consequently, there are no critical resources. The first resource that you add to the resource group is automatically marked as critical by Oracle Clusterware, provided that you have not already specified a resource type in the CRITICAL_RESOURCES attribute of the resource group. Oracle Clusterware always checks for the presence of a critical resource before attempting to start a resource group.

Resource Group Privileges

You can create resource groups and resource group types, and then create and add resources to those groups. You can also define privileges for modifying and executing operations on a resource group using the ACL attribute of the resource group. The resource group owner can assign privileges to other operating system users and groups by appropriately setting the ACL attribute of the resource group. A resource within a resource group can maintain its own privilege specification within its ACL attribute. Specifically:

- A user with write privilege on a resource group and write privilege on a resource can add the resource to the group.
- The owner of the resource group must at all times have execute privileges on all resources in the group. Any user or group granted execute privileges on the group must have execute privileges on all resources in the group.

For example, in cases where certain infrastructure resources in a resource group must be managed by `root`, the owner of the resource must be specified as `root` and execute permissions on the resource granted to the group owner. This must be done explicitly by `root` user.

- The local administrative user (`root` on Unix or Administrators group user on Windows) can modify, delete, start, and stop any resource group.

Resource Group Dependencies

You can set dependencies among resource groups, providing a means to express relationships between applications and components. Oracle Clusterware provides modifiers to specify different ordering, location, and enforcement level of dependencies amongst resource groups. Some things to consider about resource group dependencies:

- A resource group can have a dependency relationship to another resource group and not to individual resources.
- An explicitly created resource group can have a dependency relationship to an automatic resource group.
- A resource in a group can have a dependency relationship to another resource in the same group.
- Resources created without specifying a resource group (thus belonging to an automatic resource group) can have a dependency relationship to another resource group.
- A resource cannot have a dependency relationship to a resource group nor to a resource in a different resource group.

All available Oracle Clusterware resource dependencies are also available to use with resource groups. You configure the `START_DEPENDENCIES` and `STOP_DEPENDENCIES` attributes of a resource group to specify dependencies for resource groups.

Table 9-1 Resource Group Dependency Types and Modifiers

Dependency Type	Description
<i>hard start</i>	<p>Specifies the requirement that specific other resource groups must be online (anywhere in the cluster) before this resource group can be started.</p> <p>For example:</p> <pre>START_DEPENDENCIES=hard([global: intermediate: uniform:] other_resource_group)</pre> <p>If the start of any dependent resource group fails, then Oracle Clusterware aborts the start of this resource group.</p>

Table 9-1 (Cont.) Resource Group Dependency Types and Modifiers

Dependency Type	Description
<i>weak start</i>	<p>Specifies the requirement that an attempt must be made to start specific other resource groups before starting this resource group. If the attempt fails to start the specific other resource groups, then Oracle Clusterware starts this resource group, regardless.</p> <p>For example:</p> <pre>START_DEPENDENCIES=weak([global: concurrent: uniform:] <i>other_resource_group</i>)</pre>
<i>pullup</i>	<p>Use This dependency when this resource group must be automatically started when a dependent resource group starts.</p> <p>For example:</p> <pre>START_DEPENDENCIES=pullup([intermediate: always:] <i>other_resource_group</i>)</pre>
<i>hard stop</i>	<p>Oracle recommends that you use this dependency when a stop dependency exists between the resource groups.</p> <p>This dependency specifies the mandatory requirement of stopping this resource group when another specific resource group stops running.</p> <p>For example:</p> <pre>STOP_DEPENDENCIES=hard([intermediate: global: shutdown:] <i>other_resource_group</i>)</pre>
<i>attraction</i>	<p>Specifies a co-location preference with specific other resource groups.</p> <p>For example:</p> <pre>START_DEPENDENCIES=attraction([intermediate:] <i>other_r esource_group</i>)</pre>
<i>dispersion</i>	<p>Oracle Clusterware will attempt to start this resource group on the same server where a specific other resource group is already online.</p> <p>Specifies preference to not be co-located with specific other resource groups. Oracle Clusterware will attempt to start this resource group on a server with the least number of online resource groups with dispersion dependency.</p> <p>For example:</p> <pre>START_DEPENDENCIES=dispersion([intermediate: active: pool:]: <i>other_resource_group</i>)</pre>

Table 9-1 (Cont.) Resource Group Dependency Types and Modifiers

Dependency Type	Description
<i>exclusion</i>	<p>Specifies a mandatory requirement that this resource group not run on the same server as specific other resource groups. Oracle Clusterware will either reject the start of this resource group or stop the dependent resource groups and restart them on another server.</p> <p>For example:</p> <pre>START_DEPENDENCIES=exclusion[(preempt_pre: preempt_post)] other_resource_group)</pre>

Resource Group Failure and Recovery

As previously discussed, critical resources determine resource group state and failover.

Failure and Recovery of Critical Resources

- When a critical resource of a resource group fails, the resource group immediately transitions to the OFFLINE state.
- Oracle Clusterware attempts local restart of the failed critical resource according to the RESTART_ATTEMPTS and UPTIME_THRESHOLD resource attributes.
- Oracle Clusterware initiates immediate check actions on other resources in the same group that have a stop dependency on the failed resource.
- Oracle Clusterware initiates immediate check actions on other resource groups dependent on this resource group.
- If the resource restarts successfully, then the resource group transitions to ONLINE state and Oracle Clusterware performs pullup dependency evaluation within and across resource groups.
- If Oracle Clusterware exhausts all local restart attempts of the resource, then Oracle Clusterware stops the entire resource group. Oracle Clusterware also immediately stops other resource groups with a stop dependency on the resource group. Oracle Clusterware attempts local restart of the resource group, if configured to do so. On exhausting all restart attempts, the resource group will fail over to another server in the cluster.

Failure and Recovery of Non-Critical Resources

- When a non-critical resource in a resource group fails, Oracle Clusterware attempts local restart of the failed resource according to the values of the RESTART_ATTEMPTS and UPTIME_THRESHOLD resource attributes. There is no impact on the state of the resource group when a non-critical resource fails.
- Oracle Clusterware initiates immediate check actions on other resources in the same group that have a stop dependency on the failed resource.
- If the resource restarts successfully, Oracle Clusterware performs pullup dependency evaluation and corresponding startup actions.

- If Oracle Clusterware exhausts all local restart attempts of the resource, then there is no impact on the state of the resource group. You must then explicitly start the non-critical resource after fixing the cause of the failure.

Related Topics

- [Resource Group Types](#)
In Oracle Clusterware, a resource type is a template for a class of resources.
- [Using Resource Groups](#)
Use CRCTL to create resource groups, resource group types, and to add resources to resource groups.

Resource Group Types

In Oracle Clusterware, a resource type is a template for a class of resources.

Resource group types provide a commonly applicable set of attributes to all resource groups. When you create a resource group, you must specify a resource group type. Oracle Clusterware provides two base resource group types: `local_resourcegroup` and `cluster_resourcegroup`. The base resource types have attributes similar to resources, some of which you can configure.

Local Resource Group Type

Use the `local_resourcegroup` type to create a resource group that contains only local resources. Instances of a resource group of this type can run on each node in the cluster. Local resource group type attributes include:

NAME
DESCRIPTION
ACL
AUTO_START
CRITICAL_RESOURCES
DEBUG
ENABLED
INTERNAL_STATE
RESOURCE_LIST
RESTART_ATTEMPTS
SERVER_CATEGORY
START_DEPENDENCIES
STOP_DEPENDENCIES
STATE
STATE_DETAILS
UPTIME_THRESHOLD

Cluster Resource Group Type

A resource group of type `cluster_resourcegroup` can have one or more instances running on a static or dynamic set of servers in the cluster. Such a resource group can fail over to another server in the cluster according to the placement policy of the group. Cluster resource group type attributes include:

NAME
DESCRIPTION
ACL

ACTIVE_PLACEMENT
AUTO_START
CARDINALITY
CRITICAL_RESOURCES
DEBUG
ENABLED
FAILURE_INTERVAL
FAILURE_THRESHOLD
HOSTING_MEMBERS
INTERNAL_STATE
PLACEMENT
RESOURCE_LIST
RESTART_ATTEMPTS
SERVER_CATEGORY
SERVER_POOLS
START_DEPENDENCIES
STOP_DEPENDENCIES
STATE
STATE_DETAILS
UPTIME_THRESHOLD

Related Topics

- [Using Resource Groups](#)
Use CRSCTL to create resource groups, resource group types, and to add resources to resource groups.
- [Oracle Clusterware Resource Reference](#)

Using Resource Groups

Use CRSCTL to create resource groups, resource group types, and to add resources to resource groups.

To use resource groups, you must first create the resource group based on either a built-in resource group type or a resource group type that you create. Once you have create a resource group, you can add resources to it.

1. Use the following command to create a resource group:

```
$ crsctl add resourcegroup group_name -type group_type
```

The preceding command creates an empty resource group into which you can add resources. You must provide a name for the resource group and a resource group type. If you choose to base your resource group on a custom resource group type, then you must first create a resource group type, as described in the next step.

2. If you want to create a resource group based on a custom resource group type, then you must create the resource group type, as follows:

```
$ crsctl add resourcegroupstype group_type_name -basetype  
base_group_type {-file file_path | -attr  
"attribute_name=attribute_value"}
```

The preceding command creates a resource group type that provides a singular set of attributes for any resource group you created based on this resource group type. You must provide an existing resource group type as a base resource group type, and either a path to a file that contains a line-delimited list of attribute/attribute value pairs or, alternatively, you can provide a comma-delimited list of attribute/attribute value pairs on the command line.

3. After you create a resource group, you can begin to add resources to the resource group, as follows:

```
$ crsctl add resource resource_name -group group_name
```

The resource group to which you add a resource must exist and the resource you are adding must be in an offline state. A resource can be a member of only one resource group. If you have a resource that is shared by multiple applications, such as a file system, then Oracle recommends that you put those resources into their own individual resource groups.

Oracle Clusterware Resource Types

Generally, all resources are unique but some resources may have common attributes. Oracle Clusterware uses resource types to organize these similar resources. Benefits that resource types provide are:

- Manage only necessary resource attributes
- Manage all resources based on the resource type

Every resource that you register in Oracle Clusterware must have a certain resource type. In addition to the resource types included in Oracle Clusterware, you can define custom resource types using the Oracle Clusterware Control (CRSCTL) utility. The included resource types are:

- **Local resource:** Instances of local resources—type name is `local_resource`—run on each server of the cluster (the default) or you can limit them to run on servers belonging to a particular server category. When a server joins the cluster, Oracle Clusterware automatically extends local resources to have instances tied to the new server. When a server leaves the cluster, Oracle Clusterware automatically sheds the instances of local resources that ran on the departing server. Instances of local resources are pinned to their servers; they do not fail over from one server to another.
- **Cluster resource:** Cluster-aware resource types—type name is `cluster_resource`—are aware of the cluster environment and are subject to [cardinality](#) and cross-server switchover and failover.
- **Generic application:** You can use this resource type—type name is `generic_application`—to protect any generic applications without requiring additional scripts. High availability for an application is achieved by defining a resource with the `generic_application` resource type and providing the values for key attributes of the resource. The `generic_application` resource type is derived from the `cluster_resource` resource type and, therefore, all resources of the `generic_application` resource type are cluster-aware resources. Attributes include:
 - `START_PROGRAM`: A complete path to the executable that starts the application, with all appropriate arguments. The executable must exist on every server

where Oracle Grid Infrastructure is configured to run the application. This attribute is required. For example:

```
/opt/my_app -start
```

The executable must also ensure that the application starts and return an exit status value of zero (0) to indicate that the application started successfully and is online. If the executable fails to start the application, then the executable exits with a non-zero status code.

- **STOP_PROGRAM:** A complete path to the executable that stops the application, with all appropriate arguments. The executable must exist on every server where Oracle Grid Infrastructure is configured to run the application. If you do not specify this attribute value, then Oracle Clusterware uses an operating system-equivalent of the `kill` command. For example:

```
/opt/my_app -stop
```

The executable must also ensure that the application stops and return an exit status value of zero (0) to indicate that the application stopped successfully. If the executable fails to stop the application, then the executable exits with a non-zero status code and Oracle Clusterware initiates a clean of the resource.

- **CLEAN_PROGRAM:** A complete path to the executable that cleans the program, with all appropriate arguments. The executable must exist on every server where Oracle Grid Infrastructure is configured to run the application. If you do not specify a value for this attribute, then Oracle Clusterware uses an operating system-equivalent of the `kill -9` command. For example:

```
/opt/my_app -clean
```

 **Note:**

The difference between `STOP_PROGRAM` and `CLEAN_PROGRAM` is that `CLEAN_PROGRAM` is a forced stop that stops an application ungracefully, and must always be able to stop an application or the application becomes unmanageable.

- **PID_FILES:** A comma-delimited list of complete paths to files that will be written by the application and contain a process ID (PID) to monitor. Failure of a single process is treated as a complete resource failure. For example:

```
/opt/app.pid
```

 **Note:**

The files that you specify in the `PID_FILES` attribute are read immediately after the `START` action completes and monitoring commences for the PIDs listed in the files.

- EXECUTABLE_NAMES: A comma-delimited list of names of executables that is created when the application starts and the state of these executables is subsequently monitored. Failure of a single executable is treated as a complete resource failure. For example:

```
my_app
```

 **Note:**

You need specify only the complete name of the executables. This attribute does not accept the path of the executable or wild cards. The PIDs matching the executable names are cached immediately after the START action completes.

- CHECK_PROGRAMS: A list of complete paths to the executable that determines the state of the application. Reporting a non-running state by any of the applications is treated as a failure of the entire resource. For example:

```
/opt/my_app -check
```

- ENVIRONMENT_FILE: A complete path to the file containing environment variables to source when starting the application. The file must be a text file containing *name=value* pairs, one per line. For example:

```
/opt/my_app.env
```

- ENVIRONMENT_VARS: A comma-delimited list of *name=name* pairs to be included into the environment when starting an application. For example:

```
USE_FILES=No, AUTO_START=Yes
```

- SEND_OUTPUT_ALWAYS: This attribute is responsible for sending the application output that is sent to STDOUT, which is then displayed. A value of 0 does not display any application output unless an action fails. When an action fails, whatever application output that has been saved by the agent is displayed. Any value greater than 0 displays every application output. The default value is 0. For example:

```
SEND_OUTPUT_ALWAYS=1
```

 **Note:**

If you do not specify the STOP_PROGRAM, CHECK_PROGRAMS, and CLEAN_PROGRAM attributes, then you must specify either PID_FILES or EXECUTABLE_NAMES, or Oracle Clusterware will not allow you to register a resource of this type.

If you specify all the attributes, then the following rules apply:

1. When stopping a resource, if you specified `STOP_PROGRAM`, then Oracle Clusterware calls `STOP_PROGRAM`. Otherwise, Oracle Clusterware uses an operating system-equivalent of the `kill -9` command on the PID obtained from either the `PID_FILES` or the `EXECUTABLE_NAMES` attribute.
2. When you need to establish the current state of an application, if you specified `CHECK_PROGRAMS`, then Oracle Clusterware calls `CHECK_PROGRAMS`. Otherwise, Oracle Clusterware uses an operating system-equivalent of the `ps -p` command with the PID obtained from either the `PID_FILES` or `EXECUTABLE_NAMES` attribute.
3. When cleaning a resource, if you specified `CLEAN_PROGRAM`, then Oracle Clusterware calls `CLEAN_PROGRAM`. Otherwise, Oracle Clusterware uses an operating system-equivalent of the `kill -9` command on the PID obtained from either the `PID_FILES` or the `EXECUTABLE_NAMES` attribute.

Related Topics

- [crsctl add resource](#)
Registers a resource to be managed by Oracle Clusterware.
- [Resource States](#)
- [Creating Resources that Use the generic_application Resource Type](#)
Use the `crsctl add resource` command to create resources using the `generic_application` resource type to model any type of application requiring high availability without having to create any action scripts.

Agents in Oracle Clusterware

Oracle Clusterware runs all resource-specific commands through an entity called an *agent*.

Oracle Clusterware manages applications when they are registered as resources with Oracle Clusterware. Oracle Clusterware has access to application-specific primitives that have the ability to start, stop, and monitor a specific resource.

Note:

To increase security and further separate administrative duties, Oracle Clusterware agents run with the `SYSRAC` administrative privilege, and no longer require the `SYSDBA` administrative privilege. The `SYSRAC` administrative privilege is the default mode of connecting to the database by the Oracle Clusterware agent on behalf of Oracle RAC utilities, such as `SRVCTL`, so that no `SYSDBA` connections to the database are necessary for everyday administration of Oracle RAC database clusters.

An agent is a process that contains the agent framework and user code to manage resources. The *agent framework* is a library that enables you to plug in your application-specific code to manage customized applications. You program all of the actual application management functions, such as starting, stopping and checking the health of an application, into the agent. These functions are referred to as *entry points*.

The agent framework is responsible for invoking these entry point functions on behalf of Oracle Clusterware. Agent developers can use these entry points to plug in the

required functionality for a specific resource regarding how to start, stop, and monitor a resource. Agents are capable of managing multiple resources.

Agent developers can set the following entry points as callbacks to their code:

- **ABORT:** If any of the other entry points hang, the agent framework calls the ABORT entry point to stop the ongoing action. If the agent developer does not supply a stop function, then the agent framework exits the agent program.
- **ACTION:** The ACTION entry point is invoked when a custom action is invoked using the `clscrs_request_action` API of the `crsctl request action` command.
- **CHECK:** The CHECK (monitor) entry point acts to monitor the health of a resource. The agent framework periodically calls this entry point. If it notices any state change during this action, then the agent framework notifies Oracle Clusterware about the change in the state of the specific resource.
- **CLEAN:** The CLEAN entry point acts whenever there is a need to clean up a resource. It is a non-graceful operation that is invoked when users must forcefully terminate a resource. This command cleans up the resource-specific environment so that the resource can be restarted.
- **DELETE:** The DELETE entry point is invoked on every node where a resource can run when the resource is unregistered.
- **MODIFY:** The MODIFY entry point is invoked on every node where a resource can run when the resource profile is modified.
- **START:** The START entry point acts to bring a resource online. The agent framework calls this entry point whenever it receives the start command from Oracle Clusterware.
- **STOP:** The STOP entry points acts to gracefully bring down a resource. The agent framework calls this entry point whenever it receives the stop command from Oracle Clusterware.

START, STOP, CHECK, and CLEAN are mandatory entry points and the agent developer must provide these entry points when building an agent. Agent developers have several options to implement these entry points, including using C, C++, or scripts. It is also possible to develop agents that use both C or C++ and script-type entry points. When initializing the agent framework, if any of the mandatory entry points are not provided, then the agent framework invokes a script pointed to by the `ACTION_SCRIPT` resource attribute.

At any given time, the agent framework invokes only one entry point per application. If that entry point hangs, then the agent framework calls the ABORT entry point to end the current operation. The agent framework periodically invokes the CHECK entry point to determine the state of the resource. This entry point must return one of the following states as the resource state:

- `CLSAGFW_ONLINE:` The CHECK entry point returns ONLINE if the resource was brought up successfully and is currently in a functioning state. The agent framework continues to monitor the resource when it is in this state. This state has a numeric value of 0 for the `scriptagent`.
- `CLSAGFW_UNPLANNED_OFFLINE` and `CLSAGFW_PLANNED_OFFLINE:` The OFFLINE state indicates that the resource is not currently running. These two states have numeric values of 1 and 2, respectively, for the `scriptagent`.

Two distinct categories exist to describe an resource's offline state: *planned* and *unplanned*.

When the state of the resource transitions to OFFLINE through Oracle Clusterware, then it is assumed that the *intent* for this resource is to be offline (TARGET=OFFLINE), regardless of which value is returned from the CHECK entry point. However, when an agent detects that the state of a resource has changed independent of Oracle Clusterware (such as somebody stopping the resource through a non-Oracle interface), then the intent must be carried over from the agent to the Cluster Ready Services daemon (CRSD). The intent then becomes the determining factor for the following:

- Whether to keep or to change the value of the resource's TARGET resource attribute. PLANNED_OFFLINE indicates that the TARGET resource attribute must be changed to OFFLINE *only if* the resource was running before. If the resource was not running (STATE=OFFLINE, TARGET=OFFLINE) and a request comes in to start it, then the value of the TARGET resource attribute changes to ONLINE. The start request then goes to the agent and the agent reports back to Oracle Clusterware a PLANNED_OFFLINE resource state, and the value of the TARGET resource attribute remains ONLINE. UNPLANNED_OFFLINE does not change the TARGET attribute.
- Whether to leave the resource's state as UNPLANNED_OFFLINE or attempt to recover the resource by restarting it locally or failing it over to a another server in the cluster. The PLANNED_OFFLINE state makes CRSD leave the resource as is, whereas the UNPLANNED_OFFLINE state prompts resource recovery.
- CLSAGFW_UNKNOWN: The CHECK entry point returns UNKNOWN if the current state of the resource cannot be determined. In response to this state, Oracle Clusterware does not attempt to failover or to restart the resource. The agent framework continues to monitor the resource if the previous state of the resource was either ONLINE or PARTIAL. This state has a numeric value of 3 for the scriptagent.
- CLSAGFW_PARTIAL: The CHECK entry point returns PARTIAL when it knows that a resource is partially ONLINE and some of its services are available. Oracle Clusterware considers this state as partially ONLINE and does not attempt to failover or to restart the resource. The agent framework continues to monitor the resource in this state. This state has a numeric value of 4 for the scriptagent.
- CLSAGFW_FAILED: The CHECK entry point returns FAILED whenever it detects that a resource is not in a functioning state and some of its components have failed and some clean up is required to restart the resource. In response to this state, Oracle Clusterware calls the CLEAN action to clean up the resource. After the CLEAN action finishes, the state of the resource is expected to be OFFLINE. Next, depending on the policy of the resource, Oracle Clusterware may attempt to failover or restart the resource. Under no circumstances does the agent framework monitor failed resources. This state has a numeric value of 5 for the scriptagent.

The agent framework implicitly monitors resources in the states listed in [Table 9-2](#) at regular intervals, as specified by the CHECK_INTERVAL or OFFLINE_CHECK_INTERVAL resource attributes.

Table 9-2 Agent Framework Monitoring Characteristics

State	Condition	Frequency
ONLINE	Always	\ CHECK_INTERVAL
PARTIAL	Always	CHECK_INTERVAL

Table 9-2 (Cont.) Agent Framework Monitoring Characteristics

State	Condition	Frequency
OFFLINE	Only if the value of the <code>OFFLINE_CHECK_INTERVAL</code> resource attribute is greater than 0.	<code>OFFLINE_CHECK_INTERVAL</code>
UNKNOWN	Only monitored if the resource was previously being monitored because of any one of the previously mentioned conditions.	If the state becomes UNKNOWN after being ONLINE, then the value of <code>CHECK_INTERVAL</code> is used. Otherwise, there is no monitoring.

Whenever an agent starts, the state of all the resources it monitors is set to UNKNOWN. After receiving an initial probe request from Oracle Clusterware, the agent framework executes the CHECK entry point for all of the resources to determine their current states.

Once the CHECK action successfully completes for a resource, the state of the resource transitions to one of the previously mentioned states. The agent framework then starts resources based on commands issued from Oracle Clusterware. After the completion of every action, the agent framework invokes the CHECK action to determine the current resource state. If the resource is in one of the monitored states listed in [Table 9-2](#), then the agent framework periodically executes the CHECK entry point to check for changes in resource state.

By default, the agent framework does not monitor resources that are offline. However, if the value of the `OFFLINE_CHECK_INTERVAL` attribute is greater than 0, then the agent framework monitors offline resources.

Related Topics

- [ACTION_SCRIPT](#)
- [CHECK_INTERVAL](#)
- [OFFLINE_CHECK_INTERVAL](#)

Oracle Clusterware Built-in Agents

Oracle Clusterware uses agent programs (agents) to manage resources and includes the following built-in agents to protect applications:

- **appagent**: This agent (`appagent.exe` in Windows) automatically protects resources of the `generic_application` resource type and any resources in previous versions of Oracle Clusterware of the `application` resource type.

Note:

Oracle recommends that you not use the deprecated `application` resource type, which is only provided to support pre-Oracle Clusterware 11g release 2 (11.2) resources.

- **scriptagent**: Use this agent (`scriptagent.exe` in Windows) when using shell or batch scripts to protect an application. Both the `cluster_resource` and

`local_resource` resource types are configured to use this agent, and any resources of these types automatically take advantage of this agent.

Additionally, you can create your own agents to manage your resources in any manner you want.

Related Topics

- [Oracle Clusterware Resource Types](#)
- [Building an Agent](#)

Action Scripts

An action script defines one or more actions to start, stop, check, or clean resources.

The agent framework invokes these actions without the C/C++ actions. Using action scripts, you can build an agent that contains the C/C++ entry points and the script entry points. If all of the actions are defined in the action script, then you can use the script agent to invoke the actions defined in any action scripts.

Before invoking the action defined in the action script, the agent framework exports all the necessary attributes from the resource profile to the environment. Action scripts can log messages to the `stdout/stderr`, and the agent framework prints those messages in the agent logs. However, action scripts can use special tags to send the progress, warning, or error messages to the `crs*` client tools by prefixing one of the following tags to the messages printed to `stdout/stderr`:

```
CRS_WARNING:  
CRS_ERROR:  
CRS_PROGRESS:
```

The agent framework strips out the prefixed tag when it sends the final message to the `crs*` clients.

Resource attributes can be accessed from within an action script as environment variables prefixed with `_CRS_`. For example, the `START_TIMEOUT` attribute becomes an environment variable named `_CRS_START_TIMEOUT`.

Related Topics

- [crsctl start resource](#)
- [crsctl stop resource](#)

Building an Agent

Building an [agent](#) for a specific application involves the following steps:

1. Implement the [agent framework](#) entry points either in scripts, C, or C++.
2. Build the agent executable (for C and C++ agents).
3. Collect all the parameters needed by the entry points and define a new resource type. Set the `AGENT_FILENAME` attribute to the absolute path of the newly built executable.

Building and Deploying C and C++ Agents

Example C and C++ agents are included with Oracle Clusterware that demonstrate using the agent framework to implement high availability agents for applications. Appendix F describes an example of an agent called `demoagent1.cpp`. This agent manages a simple resource that represents a file on disk and performs the following tasks:

- On start: Creates the file
- On stop: Gracefully deletes the file
- On check: Detects whether the file is present
- On clean: Forcefully deletes the file

To describe this particular resource to Oracle Clusterware, you must first create a resource type that contains all the characteristic attributes for this resource class. In this case, the only attribute to be described is the name of the file to be managed. The following steps demonstrate how to set up the resource and its agent and test the functionality of the resource:

1. Compile the C++ agent using the `demoagent1.cpp` source file provided and a *makefile*. Modify the makefile based on the local compiler and linker paths and installation locations. The output is an executable named `demoagent1`. This example assumes that the executable is located in a directory named `/path/to/` on every node of the cluster.
2. Use `CRSCTL` to add a new resource type, as follows:

```
$ crsctl add type hotfile_type -basetype cluster_resource -attr
  "ATTRIBUTE=PATH_NAME,TYPE=string,DEFAULT_VALUE=default.txt,
  ATTRIBUTE=AGENT_FILENAME,TYPE=string,DEFAULT_VALUE=/path/to/
demoagent1"
```

In the preceding command example, `PATH_NAME` is the directory path for every resource of this type. Modify the value of `PATH_NAME` to the appropriate directory location on the disk.

The `AGENT_FILENAME` attribute specifies the location of the agent binary that implements the resource management commands for this resource type. This step adds a new resource type to Oracle Clusterware.

3. Create a new resource based on the type that is defined in the previous step, as follows:

```
$ crsctl add res file1 -type hotfile_type -attr "PATH_NAME=/var/log/
file1.txt"
$ crsctl add res file2 -type hotfile_type -attr "PATH_NAME=/var/log/
file2.txt"
```

The preceding commands add resources named `file1` and `file2` to be managed and monitored by Oracle Clusterware.

4. Start and stop the resources using CRSCTL, as follows:

```
$ crsctl start res file1
$ crsctl start res file2
$ crsctl relocate res file1
$ crsctl stop res file2
```

Oracle Clusterware creates and deletes the disk files as the resources are started and stopped.

Related Topics

- [Oracle Clusterware Agent Framework C Application Program Interfaces](#)

Registering a Resource in Oracle Clusterware

Register resources in Oracle Clusterware 12c using the `crsctl add resource` command.

To register an application as a resource:

```
$ crsctl add resource resource_name -type [-group group_name] resource_type
  [-file file_path] | [-attr "attribute_name='attribute_value',
attribute_name='
attribute_value', ..."]
```

Choose a name for the resource based on the application for which it is being created. For example, if you create a resource for an Apache Web server, then you might name the resource `myApache`. Specify the name of an existing resource type after the `-type` option. Optionally, you can add the resource to an existing resource group.

You can specify resource attributes in either a text file specified with the `-file` option or in a comma-delimited list of resource attribute-value pairs enclosed in double quotation marks (") following the `-attr` option. You must enclose space- or comma-delimited attribute names and values enclosed in parentheses in single quotation marks (').

The following is an example of an attribute file:

```
PLACEMENT=favored
HOSTING_MEMBERS=node1 node2 node3
RESTART_ATTEMPTS@CARDINALITYID(1)=0
RESTART_ATTEMPTS@CARDINALITYID(2)=0
FAILURE_THRESHOLD@CARDINALITYID(1)=2
FAILURE_THRESHOLD@CARDINALITYID(2)=4
FAILURE_INTERVAL@CARDINALITYID(1)=300
FAILURE_INTERVAL@CARDINALITYID(2)=500
CHECK_INTERVAL=2
CARDINALITY=2
```

The following is an example of using the `-attr` option:

```
$ crsctl add resource resource_name -type resource_type [-attr "PLACEMENT='
favored', HOSTING_MEMBERS='node1 node2 node3', ..."]
```

Related Topics

- [Adding User-Defined Resources](#)
You can add resources to Oracle Clusterware at any time.
- [crsctl add resource](#)
Registers a resource to be managed by Oracle Clusterware.
- [Resource Attributes](#)
- [Resource Groups](#)
A resource group is a container for a logically related group of resources.

Overview of Using Oracle Clusterware to Enable High Availability

Oracle Clusterware manages resources and resource groups based on how you configure them to increase their availability.

You can configure your resources and resource groups so that Oracle Clusterware:

- Starts resources and resource groups during cluster or server start
- Restarts resources and resource groups when failures occur
- Relocates resources and resource groups to other servers, if the servers are available

To manage your applications with Oracle Clusterware:

1. Use the `generic_application` resource type, write a custom script for the script agent, or develop a new agent.
2. Register your applications as resources with Oracle Clusterware.

If a single application requires that you register multiple resources, then you can create a resource group that Oracle Clusterware manages like a single resource. You may be required to define relevant dependencies between the resources within the resource group.
3. Assign the appropriate privileges to the resource or resource group.
4. Start or stop your resources and resource groups.

When a resource fails, Oracle Clusterware attempts to restart the resource based on attribute values that you provide when you register an application or process as a resource. If the failed resource is a non-critical resource member of a resource group, then the resource group remains in an ONLINE state. If a server in a cluster fails, then you can configure your resources and resource groups so that processes that were assigned to run on the failed server restart on another server. Based on various resource attributes, Oracle Clusterware supports a variety of configurable scenarios.

When you register a resource or create a resource group in Oracle Clusterware, the relevant information about the application and the resource-relevant information, is stored in the Oracle Cluster Registry (OCR). This information includes:

- **Path to the action script or application-specific agent:** This is the absolute path to the script or application-specific agent that defines the start, stop, check, and clean actions that Oracle Clusterware performs on the application.

 **See Also:**

"[Agents in Oracle Clusterware](#)" for more information about these actions

- **Privileges:** Oracle Clusterware has the necessary privileges to control all of the components of your application for high availability operations, including the right to start processes that are owned by other user identities. Oracle Clusterware must run as a privileged user to control applications with the correct start and stop processes.
- **Resource Dependencies:** You can create relationships among resources and resource groups that imply an operational ordering or that affect the placement of resources on servers in the cluster. For example, Oracle Clusterware can only start a resource that has a hard start dependency on another resource if the other resource is running. Oracle Clusterware prevents stopping a resource if other resources that depend on it are running. However, you can force a resource to stop using the `crsctl stop resource -f` command, which first stops all resources that depend on the resource being stopped.

Related Topics

- [Resource Groups](#)
A resource group is a container for a logically related group of resources.
- [Resource Attributes](#)
- [Resource States](#)
- [Resource Dependencies](#)
- [Resource Placement](#)
- [Agents in Oracle Clusterware](#)
Oracle Clusterware runs all resource-specific commands through an entity called an *agent*.

Resource Attributes

Resource attributes define how Oracle Clusterware manages resources of a specific resource type. Each resource type has a unique set of attributes. Some resource attributes are specified when you register resources, while others are internally managed by Oracle Clusterware.

 **Note:**

Where you can define new resource attributes, you can only use US-7 ASCII characters.

Related Topics

- [Oracle Clusterware Resource Reference](#)

Resource States

Every resource in a cluster is in a particular state at any time. Certain actions or events can cause that state to change.

Table 9-3 lists and describes the possible [resource states](#).

Table 9-3 Possible Resource States

State	Description
ONLINE	The resource is running.
OFFLINE	The resource is not running.
UNKNOWN	An attempt to stop the resource has failed. Oracle Clusterware does not actively monitor resources that are in this state. You must perform an application-specific action to ensure that the resource is offline, such as stop a process, and then run the <code>crsctl stop resource</code> command to reset the state of the resource to OFFLINE.
INTERMEDIATE	<p>A resource can be in the INTERMEDIATE state because of one of two events:</p> <ol style="list-style-type: none"> 1. Oracle Clusterware cannot determine the state of the resource but the resource was either attempting to go online or was online the last time its state was precisely known. Usually, the resource transitions out of this state on its own over time, as the conditions that impeded the check action no longer apply. 2. A resource is partially online. For example, the Oracle Database VIP resource fails over to another server when its home server leaves the cluster. However, applications cannot use this VIP to access the database while it is on a non-home server. Similarly, when an Oracle Database instance is started and not open, the resource is partially online: it is running but is not available to provide services. <p>Oracle Clusterware actively monitors resources that are in the INTERMEDIATE state and, typically, you are not required to intervene. If the resource is in the INTERMEDIATE state due to the preceding reason 1, then as soon as the state of the resource is established, Oracle Clusterware transitions the resource out of the INTERMEDIATE state.</p> <p>If the resource is in the INTERMEDIATE state due to the preceding reason 2, then it stays in this state if it remains partially online. For example, the home server of the VIP must rejoin the cluster so the VIP can switch over to it. A database administrator must issue a command to open the database instance.</p> <p>In either case, however, Oracle Clusterware transitions the resource out of the INTERMEDIATE state automatically as soon as it is appropriate. Use the <code>STATE_DETAILS</code> resource attribute to explain the reason for a resource being in the INTERMEDIATE state and provide a solution to transition the resource out of this state.</p>

Resource Dependencies

You can configure resources to be dependent on other resources, so that the dependent resources can only start or stop when certain conditions of the resources on which they depend are met. For example, when Oracle Clusterware attempts to start a resource, it is necessary for any resources on which the initial resource depends to be running and in the same location. If Oracle Clusterware cannot bring the resources online, then the initial (dependent) resource cannot be brought online, either. If Oracle Clusterware stops a resource or a resource fails, then any dependent resource is also stopped.

Some resources require more time to start than others. Some resources must start whenever a server starts, while other resources require a manual start action. These and many other examples of resource-specific behavior imply that each resource must be described in terms of how it is expected to behave and how it relates to other resources (resource dependencies).

You can configure resources so that they depend on Oracle resources. When creating resources, however, do not use an *ora* prefix in the resource name. This prefix is reserved for Oracle use only.

Previous versions of Oracle Clusterware included only two dependency specifications: the `REQUIRED_RESOURCES` resource attribute and the `OPTIONAL_RESOURCES` resource attribute. The `REQUIRED_RESOURCES` resource attribute applied to both start and stop resource dependencies.



Note:

The `REQUIRED_RESOURCES` and `OPTIONAL_RESOURCES` resource attributes are still available only for resources of `application` type. Their use to define resource dependencies in Oracle Clusterware 12c is deprecated.

Resource dependencies are separated into start and stop categories. This separation improves and expands the start and stop dependencies between resources and resource types.

This section includes the following topics:

- [Start Dependencies](#)
- [Stop Dependencies](#)

Start Dependencies

Oracle Clusterware considers start dependencies contained in the profile of a resource when the [start effort evaluation](#) for that resource begins. You specify start dependencies for resources using the `START_DEPENDENCIES` resource attribute. You can use modifiers on each dependency to further configure the dependency.

This section includes descriptions of the following START dependencies:

- [attraction](#)
- [dispersion](#)

- [exclusion](#)
- [hard](#)
- [pullup](#)
- [weak](#)

Related Topics

- [START_DEPENDENCIES](#)

attraction

If resource A has an `attraction` dependency on resource B, then Oracle Clusterware prefers to place resource A on servers hosting resource B. Dependent resources, such as resource A in this case, are more likely to run on servers on which resources to which they have `attraction` dependencies are running. Oracle Clusterware places dependent resources on servers with resources to which they are attracted.

You can configure the `attraction` start dependency with the following constraints:

- `START_DEPENDENCIES=attraction(intermediate:resourceB)`
Use the `intermediate` modifier to specify whether the resource is attracted to resources that are in the `INTERMEDIATE` state.
- `START_DEPENDENCIES=attraction(type:resourceB.type)`
Use the `type` modifier to specify whether the dependency acts on a particular resource type. The dependent resource is attracted to the server hosting the greatest number of resources of a particular type.

Note:

Previous versions of Oracle Clusterware used the now deprecated `OPTIONAL_RESOURCES` attribute to express attraction dependency.

dispersion

If you specify the `dispersion` start dependency for a resource, then Oracle Clusterware starts this resource on a server that has the fewest number of resources to which this resource has dispersion. Resources with dispersion may still end up running on the same server if there are not enough servers to which to disperse them.

You can configure the `dispersion` start dependency with the following modifiers:

- `START_DEPENDENCIES=dispersion(intermediate:resourceB)`
Use the `intermediate` modifier to specify that Oracle Clusterware disperses resource A whether resource B is either in the `ONLINE` or `INTERMEDIATE` state.
- `START_DEPENDENCIES=dispersion:active(resourceB)`
Typically, dispersion is only applied when starting resources. If at the time of starting, resources that disperse each other start on the same server (because there are not enough servers at the time the resources start), then Oracle Clusterware leaves the resources alone once they are running, even when more

servers join the cluster. If you specify the `active` modifier, then Oracle Clusterware reapplies dispersion on resources later when new servers join the cluster.

- `START_DEPENDENCIES=dispersion(pool:resourceB)`

Use the `pool` modifier to specify that Oracle Clusterware disperses the resource to a different server pool rather than to a different server.

exclusion

The `exclusion` start dependency contains a clause that defines the exclusive relationship between resources while starting. Resources that have the `exclusion` start dependency cannot run on the same node. For example, if resource A has an `exclusion` start dependency on resource B, then the CRSD policy provides the following options when resource B is already running on the server where resource A needs to start:

- Deny the start of resource A if resource B is already running.
- Start resource A by preempting resource B. There are two variations to the preempt operation:
 - Resource B is stopped and, if possible, restarted on another node. Resource A is subsequently started.
 - Resource A is started first. Subsequently, resource B is stopped and, if possible, restarted on another node.

You can configure the `exclusion` start dependency with the following modifiers:

- `START_DEPENDENCIES=exclusion([[preempt_pre: | preempt_post:]] target_resource_name | type:target_resource_type]*)`

All modifiers specified are per resource or resource type. Oracle Clusterware permits only one `exclusion` dependency per resource dependency tree. Without any `preempt` modifier, CRSD will only attempt to start the resource if all of its target resources are offline.

- `preempt_pre`: If you choose this preempt modifier, then CRSD stops the specified target resource or resources defined by a specific resource type before starting the source resource. If restarting the stopped resources is possible, then CRSD can do this concurrently while starting the preempting resource.
- `preempt_post`: If you choose this preempt modifier, then, after starting the source resource, CRSD stops and relocates, if possible, the specified target resource or resources defined by a specific resource type.

If CRSD cannot stop the target resources successfully, or cannot start the source resource, then the entire operation fails. Oracle Clusterware then attempts to return the affected resources to their original state, if possible.

hard

Define a `hard` start dependency for a resource if another resource must be running before the dependent resource can start. For example, if resource A has a `hard` start dependency on resource B, then resource B must be running before resource A can start. Similarly, if both resources (A and B) are initially offline, then resource B is started first to satisfy resource A's dependency.

 **Note:**

Oracle recommends that resources with `hard` start dependencies also have `pullup` start dependencies.

You can configure the `hard` start dependency with the following constraints:

- `START_DEPENDENCIES=hard(global:resourceB)`

By default, resources A and B must be located on the same server (collocated). Use the `global` modifier to specify that resources need not be collocated. For example, if resource A has a `hard(global:resourceB)` start dependency on resource B, then, if resource B is running on any node in the cluster, resource A can start.

- `START_DEPENDENCIES=hard(intermediate:resourceB)`

Use the `intermediate` modifier to specify that the dependent resource can start if a resource on which it depends is in either the `ONLINE` or `INTERMEDIATE` state.

- `START_DEPENDENCIES=hard(type:resourceB.type)`

Use the `type` modifier to specify whether the `hard` start dependency acts on a particular resource or a resource type. For example, if you specify that resource A has a `hard` start dependency on the `resourceB.type` type, then if any resource of the `resourceB.type` type is running, resource A can start.

- `START_DEPENDENCIES=hard(uniform:resourceB)`

Use the `uniform` modifier to attempt to start all instances of resource B, but only one instance, at least must start to satisfy the dependency.

- `START_DEPENDENCIES=hard(resourceB, intermediate:resourceC, intermediate:global:type:resourceC.type)`

You can combine modifiers and specify multiple resources in the `START_DEPENDENCIES` resource attribute.

 **Note:**

Separate modifier clauses with commas. The `type` modifier clause must always be the last modifier clause in the list and the `type` modifier must always directly precede the type.

pullup

Use the `pullup` start dependency if resource A must automatically start whenever resource B starts. This dependency only affects resource A if it is not running. As is the case for other dependencies, `pullup` may cause the dependent resource to start on any server. Use the `pullup` dependency whenever there is a hard stop dependency, so that if resource A depends on resource B and resource B fails and then recovers, then resource A is restarted.

 **Note:**

Oracle recommends that resources with hard start dependencies also have pullup start dependencies.

You can configure the pullup start dependency with the following constraints:

- `START_DEPENDENCIES=pullup(intermediate:resourceB)`
Use the `intermediate` modifier to specify whether resource B can be either in the `ONLINE` or `INTERMEDIATE` state to start resource A.
If resource A has a pullup dependency on multiple resources, then resource A starts only when all resources upon which it depends, start.
- `START_DEPENDENCIES=pullup:always(resourceB)`
Use the `always` modifier to specify whether Oracle Clusterware starts resource A despite the value of its `TARGET` attribute, whether it is `ONLINE` or `OFFLINE`. By default, without using the `always` modifier, pullup only starts resources if the value of the `TARGET` attribute of the dependent resource is `ONLINE`.
- `START_DEPENDENCIES=pullup(type:resourceB.type)`
Use the `type` modifier to specify that the dependency acts on a particular resource type.

weak

If resource A has a weak start dependency on resource B, then an attempt to start resource A attempts to start resource B, if resource B is not running. The result of the attempt to start resource B is, however, of no consequence to the result of starting resource A.

You can configure the weak start dependency with the following constraints:

- `START_DEPENDENCIES=weak(global:resourceB)`
By default, resources A and B must be collocated. Use the `global` modifier to specify that resources need not be collocated. For example, if resource A has a `weak(global:resourceB)` start dependency on resource B, then, if resource B is running on any node in the cluster, resource A can start.
- `START_DEPENDENCIES=weak(concurrent:resourceB)`
Use the `concurrent` modifier to specify that resource A and resource B can start concurrently.
- `START_DEPENDENCIES=weak(type:resourceB.type)`
Use the `type` modifier to specify that the dependency acts on a resource of a particular resource type, such as `resourceB.type`.
- `START_DEPENDENCIES=weak(uniform:resourceB)`
Use the `uniform` modifier to attempt to start all instances of resource B.

Stop Dependencies

Oracle Clusterware considers stop dependencies between resources whenever a resource is stopped (the resource state changes from `ONLINE` to any other state).

hard

If resource A has a `hard` stop dependency on resource B, then resource A must be stopped when B stops running. The two resources may attempt to start or relocate to another server, depending upon how they are configured. Oracle recommends that resources with `hard` stop dependencies also have `hard` start dependencies.

You can configure the `hard` stop dependency with the following modifiers:

- `STOP_DEPENDENCIES=hard(intermediate:resourceB)`
Use the `intermediate` modifier to specify whether resource B must be in either the `ONLINE` or `INTERMEDIATE` state for resource A to stay online.
- `STOP_DEPENDENCIES=hard(global:resourceB)`
Use the `global` modifier to specify whether resource A requires that resource B be present on the same server or on any server in the cluster to remain online. If this constraint is not specified, then resources A and B must be running on the same server. Oracle Clusterware stops resource A when that condition is no longer met.
- `STOP_DEPENDENCIES=hard(shutdown:resourceB)`
Use the `shutdown` modifier to stop the resource only when you shut down the Oracle Clusterware stack using either the `crsctl stop crs` or `crsctl stop cluster` commands.

Related Topics

- [STOP_DEPENDENCIES](#)

Effect of Resource Dependencies on Resource State Recovery

When a resource goes from a running to a non-running state, while the intent to have it running remains unchanged, this transition is called a *resource failure*. At this point, Oracle Clusterware applies a resource state recovery procedure that may try to restart the resource locally, relocate it to another server, or just stop the dependent resources, depending on the high availability policy for resources and the state of entities at the time.

When two or more resources depend on each other, a failure of one of them may end up causing the other to fail, as well. In most cases, it is difficult to control or even predict the order in which these failures are detected. For example, even if resource A depends on resource B, Oracle Clusterware may detect the failure of resource B after the failure of resource A.

This lack of failure order predictability can cause Oracle Clusterware to attempt to restart dependent resources in parallel, which, ultimately, leads to the failure to restart some resources, because the resources upon which they depend are being restarted out of order.

In this case, Oracle Clusterware reattempts to restart the dependent resources locally if either or both the `hard` stop and `pullup` dependencies are used. For example, if

resource A has either a `hard stop` dependency or `pullup` dependency, or both, on resource B, and resource A fails because resource B failed, then Oracle Clusterware may end up trying to restart both resources at the same time. If the attempt to restart resource A fails, then as soon as resource B successfully restarts, Oracle Clusterware reattempts to restart resource A.

Resource Placement

As part of the start effort evaluation, the first decision that Oracle Clusterware must make is where to start (or place) the resource. Making such a decision is easy when the caller specifies the target server by name. If a target server is not specified, however, then Oracle Clusterware attempts to locate the best possible server for placement given the resource's configuration and the current state of the cluster.

Oracle Clusterware considers a resource's placement policy first and filters out servers that do not fit with that policy. Oracle Clusterware sorts the remaining servers in a particular order depending on the value of the `PLACEMENT` resource attribute of the resource.

The result of this consideration is a maximum of two lists of candidate servers on which Oracle Clusterware can start the resource. One list contains preferred servers and the other contains possible servers. The list of preferred servers will be empty if the value of the `PLACEMENT` resource attribute for the resource is set to `balanced` or `restricted`. The placement policy of the resource determines on which server the resource wants to run. Oracle Clusterware considers preferred servers over possible servers, if there are servers in the preferred list.

Oracle Clusterware then considers the resource's dependencies to determine where to place the resource, if any exist. The `attraction` and `dispersion` start dependencies affect the resource placement decision, as do some dependency modifiers. Oracle Clusterware applies these placement hints to further order the servers in the two previously mentioned lists. Note that Oracle Clusterware processes each list of servers independently, so that the effect of the resource's placement policy is not confused by that of dependencies.

Finally, Oracle Clusterware chooses the first server from the list of preferred servers, if any servers are listed. If there are no servers on the list of preferred servers, then Oracle Clusterware chooses the first server from the list of possible servers, if any servers are listed. When no servers exist in either list, Oracle Clusterware generates a resource placement error.

Note:

Neither the placement policies nor the dependencies of the resources related to the resource Oracle Clusterware is attempting to start affect the placement decision.

Related Topics

- [Application Placement Policies](#)
A resource can be started on any server, subject to the placement policies, the resource start dependencies, and the availability of the action script on that server.

Registering an Application as a Resource

This section presents examples of the procedures for registering an application as a resource in Oracle Clusterware.

The procedures instruct you how to add an Apache Web server (as an example) as a resource to Oracle Clusterware. The examples in this section assume that the Oracle Clusterware administrator has full administrative privileges over Oracle Clusterware and the user or group that owns the application that Oracle Clusterware is going to manage. Once the registration process is complete, Oracle Clusterware can start any application on behalf of any operating system user.

Oracle Clusterware distinguishes between an owner of a registered resource and a user. The owner of a resource is the operating system user under which the agent runs. The `ACL` resource attribute of the resource defines permissions for the users and the owner. Only `root` can modify any resource.

Note:

- Oracle Clusterware commands prefixed with `crs_` are desupported with this release and can no longer be used. `CRSCTL` commands replace those commands. See "Oracle Clusterware Control (CRSCTL) Utility Reference" for a list of `CRSCTL` commands and their corresponding `crs_` commands.
- Do not use `CRSCTL` commands on any resources that have names prefixed with `ora` (because these are Oracle resources), unless My Oracle Support directs you to do so.

To configure Oracle resources, use the server control utility, `SRVCTL`, which provides you with all configurable options.

Related Topics

- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)
- [Role-Separated Management](#)
Role-separated management is an approach to managing cluster resources and workloads in a coordinated fashion in order to reduce the risks of resource conflicts and shortages.

Creating an Application VIP Managed by Oracle Clusterware

An application VIP is a cluster resource that Oracle Clusterware manages (Oracle Clusterware provides a standard VIP agent for application VIPs).

If clients of an application access the application through a network, and the placement policy for the application allows it to fail over to another node, then you must register a virtual internet protocol address (VIP) on which the application depends. You should base any new application VIPs on this VIP type to ensure that your system experiences consistent behavior among all of the VIPs that you deploy in your cluster.

While you can add a VIP in the same way that you can add any other resource that Oracle Clusterware manages, Oracle recommends using the `Grid_home/bin/appvipcfg` command-line utility to create or delete an application VIP on the default network for which the `ora.net1.network` resource is created by default.

To create an application VIP, use the following syntax:

```
appvipcfg create -network=network_number -ip=ip_address -vipname=vip_name
-user=user_name [-group=group_name] [-failback=0 | 1]
```

 **Note:**

You can modify the VIP name while the resource remains online, without restarting the resource.

When you create an application VIP on a default network, set `-network=1`.

To create an application VIP on a non-default network, you may have to first create the network using the `srvctl add network` command. Then you can create the application VIP, setting `-network=non-default_network_number`.

In an Oracle Flex Cluster, you can also add a non-Hub Node network resource for application VIPs, so that applications can run on non-Hub Nodes using the `srvctl add network` command, as follows:

```
srvctl add network -netnum=network_number -subnet subnet/netmask[if1[if2|...]]
```

To delete an application VIP, use the following syntax:

```
appvipcfg delete -vipname=vip_name
```

In the preceding syntax examples, `network_number` is the number of the network, `ip_address` is the IP address, `vip_name` is the name of the VIP, `user_name` is the name of the user who installed Oracle Database, and `group_name` is the name of the group. The default value of the `-failback` option is 0. If you set the option to 1, then the VIP (and therefore any resources that depend on VIP) fails back to the original node when it becomes available again.

 **Note:**

The `-ip=ip_address` parameter is required, but if Grid Plug and Play and GNS with DHCP have been configured, the parameter always takes the IP address from the DHCP server and ignores the IP address specified in the command. The value for the `-vipname=vip_name` parameter is also ignored with DHCP.

For example, as `root`, run the following command:

```
# Grid_home/bin/appvipcfg create -network=1 -ip=148.87.58.196 -  
vipname=appsVIP -user=root
```

The script only requires a network number, the IP address, and a name for the VIP resource, in addition to the user that owns the application VIP resource. A VIP resource is typically owned by `root` because VIP related operations require root privileges.

To delete an application VIP, use the same script with the `delete` option. This option accepts the VIP name as a parameter. For example:

```
# Grid_home/bin/appvipcfg delete -vipname=appsVIP
```

After you have created the application VIP using this configuration script, you can view the VIP profile using the following command:

```
$ Grid_home/bin/crsctl status res appsVIP -p
```

Verify and, if required, modify the following parameters using the `Grid_home/bin/crsctl modify res` command.

The `appvipcfg` script requires that you specify the `-network` option, even if `-network=1`.

As the Oracle Database installation owner, start the VIP resource:

```
$ crsctl start resource appsVIP
```

Related Topics

- [Oracle Real Application Clusters Administration and Deployment Guide](#)
- [Oracle Flex Clusters](#)
An Oracle Flex Cluster scales Oracle Clusterware to large numbers of nodes.
- [Oracle Clusterware Resource Reference](#)

Adding an Application VIP with Oracle Enterprise Manager

To add an application VIP with Oracle Enterprise Manager:

1. Log into Oracle Enterprise Manager Cloud Control.
2. Select the cluster target that you want to modify.
3. From the cluster target menu, select **Administration > Resources > Manage**.
4. Enter a cluster administrator user name and password to display the Manage Resources page.
5. Click **Add Application VIP**.
6. Enter a name for the VIP in the **Name** field.
7. Enter a network number in the **Network Number** field.

8. Enter an IP address for the VIP in the **Internet Protocol Address** field.
9. Enter **root** in the **Primary User** field. Oracle Enterprise Manager defaults to whatever user name you are logged in as.
10. Select **Start the resource after creation** if you want the VIP to start immediately.
11. Click **Continue** to display the Confirmation: Add VIP Resource page.
12. Enter **root** and the root password as the cluster credentials.
13. Click **Continue** to create the application VIP.

Adding User-Defined Resources

You can add resources to Oracle Clusterware at any time.

However, if you add a resource that depends on another resource, then you must first add the resource upon which it is dependent.

In the examples in this section, assume that an action script, `myApache.scr`, resides in the `/opt/cluster/scripts` directory on each node to facilitate adding the resource to the cluster. Assume also that a server pool has been created to host an application. This server pool is not a sub-pool of Generic, but instead it is used to host the application in a top-level server pool.

Note:

Oracle recommends that you use shared storage, such as Oracle Automatic Storage Management Cluster File System (Oracle ACFS), to store action scripts to decrease script maintenance.

This section includes the following topics:

- [Deciding on a Deployment Scheme](#)
- [Adding a Resource to a Specified Server Pool](#)
- [Adding a Resource Using a Server-Specific Deployment](#)
- [Creating Resources that Use the generic_application Resource Type](#)

Related Topics

- [Examples of Action Scripts for Third-party Applications](#)

Deciding on a Deployment Scheme

You must decide whether to use administrator or policy management for the application. Use administrator management for smaller, two-node configurations, where your cluster configuration is not likely to change. Use policy management for more dynamic configurations when your cluster consists of more than two nodes. For example, if a resource only runs on node 1 and node 2 because only those nodes have the necessary files, then administrator management is probably more appropriate.

Oracle Clusterware supports the deployment of applications in access-controlled server pools made up of anonymous servers and strictly based on the desired pool

size. Cluster policies defined by the administrator can and must be used in this case to govern the server assignment with desired sizes and levels of importance. Alternatively, a strict or preferred server assignment can be used, in which resources run on specifically named servers. This represents the pre-existing model available in earlier releases of Oracle Clusterware now known as administrator management.

Conceptually, a cluster hosting applications developed and deployed in both of the deployment schemes can be viewed as two logically separated groups of servers. One server group is used for server pools, enabling role separation and server capacity control. The other server group assumes a fixed assignment based on named servers in the cluster.

To manage an application using either deployment scheme, you must create a server pool before adding the resource to the cluster. A built-in server pool named *Generic* always owns the servers used by applications of administrator-based management. The Generic server pool is a logical division and can be used to separate the two parts of the cluster using different management schemes.

For third party developers to use the model to deploy applications, server pools must be used. To take advantage of the pre-existing application development and deployment model based on named servers, sub-pools of Generic (server pools that have Generic as their parent pool, defined by the server pool attribute `PARENT_POOLS`) must be used. By creating sub-pools that use Generic as their parent and enumerating servers by name in the sub-pool definitions, applications ensure that named servers are in Generic and are used exclusively for applications using the named servers model.

Adding a Resource to a Specified Server Pool

Use the `crsctl add resource` command to add a resource to a server pool.

To add the Apache Web server to a specific server pool as a resource using the policy-based deployment scheme, run the following command as the user that is supposed to run the Apache Server (for an Apache Server this is typically the `root` user):

```
$ crsctl add resource myApache -type cluster_resource -attr
  "ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,
  PLACEMENT=restricted,
  SERVER_POOLS=server_pool_list,
  CHECK_INTERVAL=30,
  RESTART_ATTEMPTS=2,
  START_DEPENDENCIES=hard(appsvip),
  STOP_DEPENDENCIES=hard(appsvip) "
```

In the preceding example, `myApache` is the name of the resource added to the cluster.

 **Note:**

- You must enclose comma or space-delimited attribute values in single quotation marks (' ') to avoid errors. If you enclose single attributes values in single quotation marks, they are ignored and no errors ensue.
- A resource name cannot begin with a period nor with the character string *ora*.

The resource is configured, as follows:

- The resource is a `cluster_resource` type.
- `ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr`: The path to the required action script.
- `PLACEMENT=restricted`
- `SERVER_POOLS=server_pool_list`: This resource can only run in the server pools specified in a space-delimited list.
- `CHECK_INTERVAL=30`: Oracle Clusterware checks this resource every 30 seconds to determine its status.
- `RESTART_ATTEMPTS=2`: Oracle Clusterware attempts to restart this resource twice before failing it over to another node.
- `START_DEPENDENCIES=hard(appsvip)`: This resource has a hard START dependency on the `appsvip` resource. The `appsvip` resource must be online in order for `myApache` to start.
- `STOP_DEPENDENCIES=hard(appsvip)`: This resource has a hard STOP dependency on the `appsvip` resource. The `myApache` resource stops if the `appsvip` resource goes offline.

Related Topics

- [Examples of Action Scripts for Third-party Applications](#)
- [Application Placement Policies](#)
A resource can be started on any server, subject to the placement policies, the resource start dependencies, and the availability of the action script on that server.

Adding a Resource Using a Server-Specific Deployment

To add the Apache Web server as a resource that uses a named server deployment, assume that you add the resource to a server pool that is, by definition, a sub-pool of the Generic server pool. You create server pools that are sub-pools of Generic using the `crsctl add serverpool` command. These server pools define the Generic server pool as their parent in the server pool attribute `PARENT_POOLS`. In addition, they include a list of server names in the `SERVER_NAMES` parameter to specify the servers that should be assigned to the respective pool. For example:

```
$ crsctl add serverpool myApache_sp -attr  
  "PARENT_POOLS=Generic, SERVER_NAMES=host36 host37"
```

After you create the sub-pool, add the Apache Web server resource, as follows:

```
$ crsctl add resource myApache -type cluster_resource -attr
"ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,
  PLACEMENT='restricted',
  SERVER_POOLS=myApache_sp,
  CHECK_INTERVAL='30',
  RESTART_ATTEMPTS='2',
  START_DEPENDENCIES='hard(appsvip)',
  STOP_DEPENDENCIES='hard(appsvip)'"
```

 **Note:**

A resource name cannot begin with a period nor with the character string *ora*.

In addition, note that when adding a resource using a server-specific deployment, the server pools listed in the `SERVER_POOLS` resource parameter must be sub-pools under Generic.

Related Topics

- [Policy-Based Cluster and Capacity Management](#)

Creating Resources that Use the `generic_application` Resource Type

Use the `crsctl add resource` command to create resources using the `generic_application` resource type to model any type of application requiring high availability without having to create any action scripts.

This section includes two examples for Linux/UNIX platforms of creating resources that use the `generic_application` resource type.

In the following command example, a Samba server resource is created for high availability:

```
$ crsctl add resource sambal -type generic_application -attr
"START_PROGRAM='/etc/init.d/smb start',
  STOP_PROGRAM='/etc/init.d/smb stop',
  CLEAN_PROGRAM='/etc/init.d/smb stop',
  PID_FILES='/var/run/smbd.pid,/var/run/nmbd.pid'"
```

In the preceding example, the attributes that define the resource are configured, as follows:

- `START_PROGRAM='/etc/init.d/smb start'`: This attribute contains the complete path and arguments to the script that starts the Samba server
- `STOP_PROGRAM='/etc/init.d/smb stop'`: This attribute contains the complete path and arguments to the script that stops the Samba server

- `CLEAN_PROGRAM='/etc/init.d/smb stop'`: This attribute contains the complete path and arguments to the script that forcefully terminates and cleans up the Samba server in case there is any failure in starting or stopping the server
- `PID_FILES='/var/run/smbd.pid,/var/run/nmbd.pid'`: This attribute contains the paths to the text files listing the process IDs (PIDs) that must be monitored to ensure that the Samba server and all its components are running

 **Note:**

- If script-based monitoring is required for this Samba server configuration, then you can use the `CHECK_PROGRAMS` attribute instead of the `PID_FILES` attribute, as follows:

```
CHECK_PROGRAMS='/etc/init.d/smb status'
```

- You can specify standard Oracle Clusterware placement and cardinality properties by configuring the `HOSTING_MEMBERS`, `SERVER_POOLS`, `PLACEMENT`, and `CARDINALITY` attributes of the Samba server resource.

In the second command example, a database file server (DBFS) resource is created for high availability. The DBFS provides a Filesystem in Userspace (FUSE) file system to access data stored in an Oracle Database.

You can use the `generic_application` resource type to define a resource that corresponds to the DBFS file system. You can use this DBFS resource to start, stop, monitor, and failover the DBFS file system mount point. The command syntax to create this resource is as follows:

```
$ crsctl add resource dbfs1 -type generic_application -attr
"START_PROGRAM='/app/oracle/12.2/bin/dbfs_client -o wallet
/@inst1 /scratch/mjk/data/dbfs_mount',
STOP_PROGRAM='/bin/fusermount -u /scratch/mjk/data/dbfs_mount',
CHECK_PROGRAMS='ls /scratch/mjk/data/dbfs_mount/dbfsdata1',
ENVIRONMENT_VARS='ORACLE_HOME=/app/oracle/12.2,
LD_LIBRARY_PATH=/app/oracle/12.2/lib:/app/oracle/12.2/rdbms/lib,
TNS_ADMIN=/app/oracle/12.2/network/admin',
CLEAN_PROGRAM='/bin/fusermount -u -z /scratch/mjk/data/dbfs_mount',
START_DEPENDENCIES='hard(ora.inst1_srv.svc)',
STOP_DEPENDENCIES='hard(ora.inst1_srv.svc)'"
```

In addition to the mandatory `START_PROGRAM`, `STOP_PROGRAM`, `CHECK_PROGRAMS`, and `CLEAN_PROGRAM` attributes, the above example also includes the following:

- The `ENVIRONMENT_VARS` attribute specifies custom environment variables that are passed when starting or stopping the program
- The `START_DEPENDENCIES` and `STOP_DEPENDENCIES` dependency attributes create a start and stop dependency on the database service that is the underlying database store of the DBFS file system

You can create dependencies on to the DBFS resource for higher-level application resources based on the application requirements of the DBFS file system.

 **Note:**

- The `ORACLE_HOME` directory shown in the preceding syntax is an example.
- You can specify standard Oracle Clusterware placement and cardinality properties by configuring the `HOSTING_MEMBERS`, `SERVER_POOLS`, `PLACEMENT`, and `CARDINALITY` attributes of the DBFS file system resource.

Related Topics

- [Oracle Clusterware Resource Types](#)
- *Oracle Database SecureFiles and Large Objects Developer's Guide*

Adding Resources Using Oracle Enterprise Manager

Use Enterprise Manager to add resources.

To add resources to Oracle Clusterware using Oracle Enterprise Manager:

1. Log into Oracle Enterprise Manager Cloud Control.
2. Select the cluster target that you want to modify.
3. From the cluster target menu, select **Administration > Resources > Manage**.
4. Enter a cluster administrator user name and password to display the Add Resource page.
5. Enter a name for the resource in the **Name** field.

 **Note:**

A resource name cannot begin with a period nor with the character string *ora*.

6. Choose either **cluster_resource** or **local_resource** from the **Resource Type** drop down.
7. Optionally, enter a description of the resource in the **Description** field.
8. Select **Start the resource after creation** if you want the resource to start immediately.
9. The optional parameters in the **Placement** section define where in a cluster Oracle Clusterware places the resource.

The attributes in this section correspond to the attributes described in [Oracle Clusterware Resource Reference](#).
10. In the **Action Program** section, choose from the **Action Program** drop down whether Oracle Clusterware calls an action script, an agent file, or both to manage the resource.

You must also specify a path to the script, file, or both, depending on what you select from the drop down.

If you choose **Action Script**, then you can click **Create New Action Script** to use the Oracle Enterprise Manager action script template to create an action script for your resource, if you have not yet done so.

11. To further configure the resource, click **Attributes**. On this page, you can configure start, stop, and status attributes, and offline monitoring and any attributes that you define.
12. Click **Advanced Settings** to enable more detailed resource attribute configurations.
13. Click **Dependencies** to configure start and stop dependencies between resources.
14. Click **Submit** when you finish configuring the resource.

Related Topics

- [Application Placement Policies](#)
A resource can be started on any server, subject to the placement policies, the resource start dependencies, and the availability of the action script on that server.
- [Resource Dependencies](#)

Changing Resource Permissions

Oracle Clusterware manages resources based on the permissions of the user who added the resource. The user who first added the resource owns the resource and the resource runs as the resource owner. Certain resources must be managed as `root`. If a user other than `root` adds a resource that must be run as `root`, then the permissions must be changed as `root` so that `root` manages the resource, as follows:

1. Change the permission of the named resource to `root` by running the following command as `root`:

```
# crsctl setperm resource resource_name -o root
```

2. As the user who installed Oracle Clusterware, enable the Oracle Database installation owner (`oracle`, in the following example) to run the script, as follows:

```
$ crsctl setperm resource resource_name -u user:oracle:r-x
```

3. Start the resource:

```
$ crsctl start resource resource_name
```

Application Placement Policies

A resource can be started on any server, subject to the placement policies, the resource start dependencies, and the availability of the action script on that server.

The `PLACEMENT` resource attribute determines how Oracle Clusterware selects a server on which to start a resource and where to relocate the resource after a server failure. The `HOSTING_MEMBERS` and `SERVER_POOLS` attributes determine eligible servers to host a resource and the `PLACEMENT` attribute further refines the placement of resources.

The value of the `PLACEMENT` resource attribute determines how Oracle Clusterware places resources when they are added to the cluster or when a server fails. Together with either the `HOSTING_MEMBERS` or `SERVER_POOLS` attributes, you can configure how Oracle Clusterware places the resources in a cluster. When the value of the `PLACEMENT` attribute is:

- **balanced:** Oracle Clusterware uses any online server pool for placement. Less loaded servers are preferred to servers with greater loads. To measure how loaded a server is, Oracle Clusterware uses the `LOAD` resource attribute of the resources that are in an `ONLINE` state on the server. Oracle Clusterware uses the sum total of the `LOAD` values to measure the current server load.
- **avored:** If a value is assigned to either of the `HOSTING_MEMBERS`, `SERVER_POOLS`, or `SERVER_CATEGORY` resource attributes, then that value expresses a preference. If `HOSTING_MEMBERS` is populated and either `SERVER_POOLS` or `SERVER_CATEGORY` is set, then `HOSTING_MEMBERS` indicates placement preference and `SERVER_POOLS` or `SERVER_CATEGORY` indicates a restriction. For example, the `ora.cluster.vip` resource has a policy that sets the value of `PLACEMENT` to `avored`, `SERVER_CATEGORY` is set to `Hub`, and `HOSTING_MEMBERS` is set to `server_name1`. In this case, Oracle Clusterware restricts the placement of `ora.cluster.vip` to the servers in the `Hub` category and then it prefers the server known as `server_name1`.
- **restricted:** Oracle Clusterware only considers servers that belong to server pools listed in the `SEVER_POOLS` resource attribute, servers of a particular category as configured in the `SERVER_CATEGORY` resource attribute, or servers listed in the `HOSTING_MEMBERS` resource attribute for resource placement. Only one of these resource attributes can have a value, otherwise it results in an error.

Related Topics

- [Oracle Clusterware Resource Reference](#)
 - [HOSTING_MEMBERS](#)
 - [SERVER_CATEGORY](#)
 - [SERVER_POOLS](#)
- The `SERVER_POOLS` resource attribute is described.

Unregistering Applications and Application Resources

To unregister a resource, use the `crsctl delete resource` command. You cannot unregister an application or resource that is `ONLINE` or required by another resource, unless you use the `-force` option. The following example unregisters the Apache Web server application:

```
$ crsctl delete resource myApache
```

Run the `crsctl delete resource` command as a clean-up step when a resource is no longer managed by Oracle Clusterware. Oracle recommends that you unregister any unnecessary resources.

Managing Resources

This section includes the following topics:

- [Registering Application Resources](#)
- [Starting Application Resources](#)
- [Relocating Applications and Application Resources](#)
- [Stopping Applications and Application Resources](#)
- [Displaying Clusterware Application and Application Resource Status Information](#)

Registering Application Resources

Each application that you manage with Oracle Clusterware is registered and stored as a resource in OCR.

Use the `crsctl add resource` command to register applications in OCR. For example, enter the following command to register the Apache Web server application from the previous example:

```
$ crsctl add resource myApache -type cluster_resource
-group group_name -attr "ACTION_SCRIPT=/opt/cluster/scripts/myapache.scr,
PLACEMENT=restricted,
SERVER_POOLS=server_pool_list,CHECK_INTERVAL=30,RESTART_ATTEMPTS=2,
START_DEPENDENCIES=hard(appsvip),STOP_DEPENDENCIES=hard(appsvip)"
```

In the preceding example, you can assign the resource to a resource group by specifying the `-group` parameter.

If you modify a resource, then update OCR by running the `crsctl modify resource` command.

Related Topics

- [crsctl add resource](#)
Registers a resource to be managed by Oracle Clusterware.
- [crsctl modify resource](#)
Modifies the attributes of a particular resource in Oracle Clusterware.

Starting Application Resources

Start resources with the `crsctl start resource` command.

Manually starting or stopping resources outside of Oracle Clusterware can invalidate the resource status. In addition, Oracle Clusterware may attempt to restart a resource on which you perform a manual stop operation.

To start an application resource that is registered with Oracle Clusterware, use the `crsctl start resource` command. For example:

```
$ crsctl start resource myApache
```

The command waits to receive a notification of success or failure from the action program each time the action program is called. Oracle Clusterware can start application resources if they have stopped due to exceeding their failure threshold values. You must register a resource using `crsctl add resource` before you can start it.

Running the `crsctl start resource` command on a resource sets the resource `TARGET` value to `ONLINE`. Oracle Clusterware attempts to change the state to match the `TARGET` by running the action program with the `start` action.

If a cluster server fails while you are starting a resource on that server, then check the state of the resource on the cluster by using the `crsctl status resource` command.

Related Topics

- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)

Relocating Applications and Application Resources

Use the `crsctl relocate resource` command to relocate applications and application resources.

For example, to relocate the Apache Web server application to a server named `rac2`, run the following command:

```
# crsctl relocate resource myApache -n rac2
```

Each time that the action program is called, the `crsctl relocate resource` command waits for the duration specified by the value of the `SCRIPT_TIMEOUT` resource attribute to receive notification of success or failure from the action program. A relocation attempt fails if:

- The application has required resources that run on the initial server
- Applications that require the specified resource run on the initial server

To relocate an application and its required resources, use the `-f` option with the `crsctl relocate resource` command. Oracle Clusterware relocates or starts all resources that are required by the application regardless of their state.

You can also relocate a resource group using the `crsctl relocate resourcegroup` command, which first stops the resources in the resource group before relocating the resource group on the destination server.

Online Relocation

Some resources or resource groups must remain highly available, even during a relocation operation. You can set the `RELOCATE_KIND` resource attribute (which you can also use with resource groups) to `online` (`RELOCATE_KIND=online`), which will start a new resource instance (or several instances for resources belonging to a resource group) on the destination server before stopping it on the original server when you run either the `crsctl relocate resource` or `crsctl relocate resourcegroup` command.

 **Note:**

Before using online relocation, ensure that the resource can manage the extra resource instances that are started during online relocation.

Stopping Applications and Application Resources

Stop application resources with the `crsctl stop resource` command.

The command sets the resource `TARGET` value to `OFFLINE`. Because Oracle Clusterware always attempts to match the state of a resource to its target, the Oracle Clusterware subsystem stops the application. The following example stops the Apache Web server:

```
# crsctl stop resource myApache
```

You cannot stop a resource if another resource has a hard stop dependency on it, unless you use the force (-f) option. If you use the `crsctl stop resource resource_name -f` command on a resource upon which other resources depend, and if those resources are running, then Oracle Clusterware stops the resource *and* all of the resources that depend on the resource that you are stopping.

Displaying Clusterware Application and Application Resource Status Information

Use the `crsctl status resource` command to display status information about applications and resources that are on cluster servers.

The following example displays the status information for the Apache Web server application:

```
# crsctl status resource myApache

NAME=myApache
TYPE=cluster_resource
TARGET=ONLINE
STATE=ONLINE on server010
```

Other information this command returns includes the following:

- How many times the resource has been restarted
- How many times the resource has failed within the failure interval
- The maximum number of times that a resource can restart or fail
- The target state of the resource and the normal status information

Use the -f option with the `crsctl status resource resource_name` command to view full information of a specific resource.

Enter the following command to view information about *all* applications and resources in tabular format:

```
# crsctl status resource -t
```

Related Topics

- [Oracle Clusterware Control \(CRSCTL\) Utility Reference](#)

Managing Automatic Restart of Oracle Clusterware Resources

You can prevent Oracle Clusterware from automatically restarting a resource by setting several resource attributes. You can also control how Oracle Clusterware manages the restart counters for your resources. In addition, you can customize the timeout values for the `start`, `stop`, and `check` actions that Oracle Clusterware performs on resources.

This section includes the following topics:

- [Preventing Automatic Restarts of Oracle Clusterware Resources](#)
- [Automatically Manage Restart Attempts Counter for Oracle Clusterware Resources](#)

Preventing Automatic Restarts of Oracle Clusterware Resources

To manage automatic restarts, use the `AUTO_START` resource attribute to specify whether Oracle Clusterware should automatically start a resource when a server restarts.

When a server restarts, Oracle Clusterware attempts to start the resources that run on the server as soon as the server starts. Resource startup might fail, however, if system components on which a resource depends, such as a volume manager or a file system, are not running. This is especially true if Oracle Clusterware does not manage the system components on which a resource depends.

 **Note:**

Regardless of the value of the `AUTO_START` resource attribute for a resource, the resource can start if another resource has a hard or weak start dependency on it or if the resource has a pullup start dependency on another resource.

Related Topics

- [Start Dependencies](#)
- [Oracle Clusterware Resource Reference](#)

Automatically Manage Restart Attempts Counter for Oracle Clusterware Resources

When a resource fails, Oracle Clusterware attempts to restart the resource the number of times specified in the `RESTART_ATTEMPTS` resource attribute. Note that this attribute does not specify the number of attempts to restart a failed resource (always one attempt), but rather the number of times the resource fails locally, before Oracle Clusterware attempts to fail it over. The CRSD process maintains an internal counter to track how often Oracle Clusterware restarts a resource. The number of times Oracle Clusterware has attempted to restart a resource is reflected in the `RESTART_COUNT` resource attribute. Oracle Clusterware can automatically manage the restart attempts counter based on the stability of a resource. The `UPTIME_THRESHOLD` resource attribute determines the time period that a resource must remain online, after which the `RESTART_COUNT` attribute gets reset to 0. In addition, the `RESTART_COUNT` resource attribute gets reset to 0 if the resource is relocated or restarted by the user, or the resource fails over to another server.

A

Cluster Verification Utility Reference

Cluster Verification Utility (CVU) performs system checks in preparation for installation, patch updates, or other system changes. Using CVU ensures that you have completed the required system configuration and preinstallation steps so that your Oracle Grid Infrastructure or Oracle Real Application Clusters (Oracle RAC) installation, update, or patch operation, completes successfully.

Oracle Universal Installer is fully integrated with CVU, automating many CVU prerequisite checks. Oracle Universal Installer runs all prerequisite checks and associated fixup scripts when you run the installer.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for information about using the Server Control utility (SRVCTL) to manage CVU
- *Oracle Grid Infrastructure Installation Guide* and *Oracle Real Application Clusters Installation Guide* for information about how to manually install CVU

Note:

Check for and download updated versions of CVU on Oracle Technology Network at

<http://www.oracle.com/technetwork/index.html>

This appendix describes CVU under the following topics:

- [About Cluster Verification Utility](#)
 - [Overview of CVU](#)
 - [CVU Operational Notes](#)
 - [Special CVU Topics](#)
- [Cluster Verification Utility Command Reference](#)
- [Troubleshooting and Diagnostic Output for CVU](#)

About Cluster Verification Utility

This section includes topics which relate to using CVU.

- [Overview of CVU](#)
- [CVU Operational Notes](#)
- [Special CVU Topics](#)

Overview of CVU

CVU can verify the primary cluster components during an operational phase or stage.

A component can be basic, such as free disk space, or it can be complex, such as checking Oracle Clusterware integrity. For example, CVU can verify multiple Oracle Clusterware subcomponents across Oracle Clusterware layers. Additionally, CVU can check disk space, memory, processes, and other important cluster components. A stage could be, for example, database installation, for which CVU can verify whether your system meets the criteria for an Oracle Real Application Clusters (Oracle RAC) installation. Other stages include the initial hardware setup and the establishing of system requirements through the fully operational cluster setup.

[Table A-1](#) lists verifications you can perform using CVU.

Table A-1 Performing Various CVU Verifications

Verification to Perform	CVU Commands to Use
System requirements verification	<code>cluvfy comp sys</code>
Oracle Cluster File System verification	<code>cluvfy stage [-pre -post] cfs</code>
Storage verifications	<ul style="list-style-type: none"> • <code>cluvfy comp freespace</code> • <code>cluvfy comp space</code> • <code>cluvfy comp ssa</code> • <code>cluvfy stage [-pre -post] acfscfg</code>
Network verification	<code>cluvfy stage -post hwos</code>
Connectivity verifications	<ul style="list-style-type: none"> • <code>cluvfy comp nodecon</code> • <code>cluvfy comp nodereach</code>
Cluster Time Synchronization Services verification	<code>cluvfy comp clocksync</code>
User and Permissions verification	<code>cluvfy comp admprv</code>
Node comparison and verification	<code>cluvfy comp peer</code>
Installation verification	<ul style="list-style-type: none"> • <code>cluvfy stage -post appcluster</code> • <code>cluvfy stage -post hwos</code> • <code>cluvfy stage -pre dbcfg</code> • <code>cluvfy stage -pre dbinst</code> • <code>cluvfy stage [-pre -post] crsinst</code> • <code>cluvfy stage [-pre -post] hacfg</code> • <code>cluvfy stage [-pre -post] nodeadd</code>
Deletion verification	<code>cluvfy stage -post nodedel</code>

Table A-1 (Cont.) Performing Various CVU Verifications

Verification to Perform	CVU Commands to Use
Oracle Clusterware and Oracle ASM Component verifications	<ul style="list-style-type: none"> • cluvfy comp acfs • cluvfy comp asm • cluvfy comp clumgr • cluvfy comp crs • cluvfy comp dhcp • cluvfy comp dns • cluvfy comp gns • cluvfy comp gnp • cluvfy comp ha • cluvfy comp nodeapp • cluvfy comp ocr • cluvfy comp ohasd • cluvfy comp olr • cluvfy comp scan • cluvfy comp software • cluvfy comp vdisk

CVU Operational Notes

This section includes the following topics:

- [CVU Installation Requirements](#)
- [CVU Usage Information](#)
- [CVU Configuration File](#)
- [Privileges and Security](#)
- [Using CVU Help](#)
- [Deprecated and Desupported CLUVFY Commands](#)

CVU Installation Requirements

CVU installation requirements are:

- At least 30 MB free space for the CVU software on the node from which you run CVU
- A work directory with at least 25 MB free space on each node. The default location of the work directory is `/tmp` on Linux and UNIX systems, and the value specified in the `TEMP` environment variable on Windows systems. You can specify a different location by setting the `CV_DESTLOC` environment variable.

When using CVU, the utility attempts to copy any needed information to the CVU work directory. It checks for the existence of the work directory on each node. If it does not find one, then it attempts to create one. Make sure that the CVU work directory either exists on all nodes in your cluster or proper permissions are established on each node for the user running CVU to create that directory.

CVU Usage Information

CVU includes two scripts: `runcluvfy.sh` (`runcluvfy.bat` on Windows), which you use before installing Oracle software, and `cluvfy` (`cluvfy.bat` on Windows), located in the `Grid_home` directory and `Grid_home/bin`, respectively. The `runcluvfy.sh` script contains temporary variable definitions which enable it to run before you install Oracle Grid Infrastructure or Oracle Database. After you install Oracle Grid Infrastructure, use the `cluvfy` command to check prerequisites and perform other system readiness checks.

 **Note:**

Oracle Universal Installer runs `cluvfy` to check all prerequisites during Oracle software installation.

Before installing Oracle software, run `runcluvfy.sh` from the directory where you want your Grid home to be located, as follows:

```
cd /u01/app/12.2.0/grid
./runcluvfy.sh options
```

In the preceding example, the `options` variable represents CVU command options that you select. For example:

```
$ ./runcluvfy.sh comp nodereach -n node1,node2 -verbose
```

CVU command options include:

- `-html`: Displays CVU output in HTML format
- `-verbose`: Displays explanations about all CVU checks, both failed and passed, in the detailed summary section of the output
- `-file file_location`: Save the output as an HTML file to a specific location

When you enter a CVU command, it provides a summary of the test. During preinstallation, Oracle recommends that you obtain detailed output by using the `-verbose` argument with the CVU command. The `-verbose` argument produces detailed output of individual checks. Where applicable, it shows results for each node in a tabular layout.

Run the CVU command-line tool using the `cluvfy` command. Using `cluvfy` does not adversely affect your cluster environment or your installed software. You can run `cluvfy` commands at any time, even before the Oracle Clusterware installation. In fact, CVU is designed to assist you as soon as your hardware and operating system are operational. If you run a command that requires Oracle Clusterware on a node, then CVU reports an error if Oracle Clusterware is not yet installed on that node.

The node list that you use with CVU commands should be a comma-delimited list of host names without a domain. CVU ignores domains while processing node lists. If a CVU command entry has duplicate node entries after removing domain information, then CVU eliminates the duplicate node entries.

For network connectivity verification, CVU discovers all of the available network interfaces if you do not specify an interface on the CVU command line. For storage accessibility verification, CVU discovers shared storage for all of the supported storage types if you do not specify a particular storage identification on the command line. CVU also discovers the Oracle Clusterware home if one is available.



See Also:

"Privileges and Security" for usage security information

CVU Output

CVU output consists of five distinct sections, including:

- **Header:** A single line containing information about what checks the stage or component is performing.
- **Configuration:** CVU evaluates whether the operation can be performed on all nodes. If the nodes you specify are either down or do not satisfy the necessary prerequisite conditions (such as no SSH setup), then error messages about these nodes are displayed here. This section may not always be included in the output. When the check being performed can be run on all nodes involved in the operation, then this section will not display in the output.
- **Progress Message:** CVU displays progress messages in this section as it cycles through various checks, which helps you determine if CVU hangs.
- **Detailed Summary:** By default, CVU only displays failed tasks or subtasks in this section. If you choose the `-verbose` option, then CVU displays detailed information for all tasks and subtasks.
- **Executive Summary:** Finally, CVU displays a concise summary of the entire checking process, similar to the following examples:

```
$ cluvfy stage -pre crsinst -n sales65
```

```
CVU operation performed: stage -pre crsinst  
Date: Oct 13, 2016 9:43:39 PM  
CVU home: /ade/scott_abc4/oracle/  
User: scott
```

```
$ cluvfy comp baseline -collect all -n sales65
```

```
CVU operation performed: baseline  
Date: Oct 13, 2016 9:48:19 PM  
Operating system: Linux2.6.39-400.211.1.el6uek.x86_64
```

CVU Configuration File

You can use the CVU configuration file to define specific inputs for the execution of CVU. The path for the configuration file is `Grid_home/cv/admin/cvu_config` (or `Staging_area\clusterware\stage\cvu\cv\admin` on Windows platforms). You can modify this file using a text editor. The inputs to CVU are defined in the form of key entries. You must follow these rules when modifying the CVU configuration file:

- Key entries have the syntax `name=value`
- Each key entry and the value assigned to the key only defines one property
- Lines beginning with the number sign (#) are comment lines and are ignored
- Lines that do not follow the syntax `name=value` are ignored

The following is the list of keys supported by CVU:

- `CV_NODE_ALL`: If set, it specifies the list of nodes that should be picked up when Oracle Clusterware is not installed. By default, this entry is commented out.
- `CV_ORACLE_RELEASE`: If set, it specifies the specific Oracle release (10.1, 10.2, 11.1, 11.2, 12.1, or 12.2) for which the verifications have to be performed. If set, you do not have to use the `-r release` option wherever it is applicable. The default value is 12.2.
- `CV_RAW_CHECK_ENABLED`: If set to `TRUE`, it enables the check for accessibility of shared disks on Linux and UNIX systems. This shared disk accessibility check requires that you install the `cvuqdisk` RPM Package Manager (`rpm`) on all of the nodes. By default, this key is set to `TRUE` and shared disk check is enabled.
- `CV_ASSUME_DISTID`: This property is used in cases where CVU cannot detect or support a particular platform or a distribution. Oracle does not recommend that you change this property as this might render CVU non-functional.
- `CV_XCHK_FOR_SSH_ENABLED`: If set to `TRUE`, it enables the X-Windows check for verifying user equivalence with `ssh`. By default, this entry is commented out and X-Windows check is disabled.
- `ORACLE_SRVM_REMOTECOPY`: If set, it specifies the location for the `scp` or `rcp` command to override the CVU default value. By default, this entry is commented out and CVU uses `/usr/bin/scp` and `/usr/sbin/rcp`.
- `ORACLE_SRVM_REMOTESHELL`: If set, it specifies the location for `ssh` command to override the CVU default value. By default, this entry is commented out and the tool uses `/usr/sbin/ssh`.
- `CV_ASSUME_CL_VERSION`: By default, the command line parser uses `crs activeversion` for the display of command line syntax usage and syntax validation. Use this property to pass a version other than `crs activeversion` for command line syntax display and validation. By default, this entry is commented out.

If CVU does not find a key entry defined in the configuration file, then CVU searches for the environment variable that matches the name of the key. If the environment variable is set, then CVU uses its value, otherwise CVU uses a default value for that entity.

Privileges and Security

Because of a lack of user equivalence for the `root` user, most CVU commands cannot be run as `root` to perform any remote node operations, except for the following:

- `cluvfy comp dhcp`
- `cluvfy comp dns`

However, using privilege delegation, you can specify the `-method` parameter and choose one of two methods (`sudo` or `root`) to enable the checks and run the `fixup` scripts that require root privileges to be performed on remote nodes. You will be

prompted for a password but the password is used dynamically while the CVU commands run, rather than being stored on a storage device.

Specifying the `-method` parameter is advantageous in the context of fixup scripts. If you choose privilege delegation, then all the fixup scripts can be run at one time from the local node. If you do not choose privilege delegation, then you must log onto each relevant node as `root` and run the fixup script.

Using CVU Help

The `cluvfy` commands have context sensitive help that shows their usage based on the command-line arguments that you enter. For example, if you enter `cluvfy`, then CVU displays high-level generic usage text describing the stage and component syntax. The following is a list of context help commands:

- `cluvfy -help`: CVU displays detailed CVU command information.
- `cluvfy -version`: CVU displays the version of Oracle Clusterware.
- `cluvfy comp -list`: CVU displays a list of components that can be checked, and brief descriptions of how the utility checks each component.
- `cluvfy comp -help`: CVU displays detailed syntax for each of the valid component checks.
- `cluvfy stage -list`: CVU displays a list of valid stages.
- `cluvfy stage -help`: CVU displays detailed syntax for each of the valid stage checks.

You can also use the `-help` option with any CVU command. For example, `cluvfy stage -pre nodeadd -help` returns detailed information for that particular command.

If you enter an invalid CVU command, then CVU shows the correct usage for that command. For example, if you type `cluvfy stage -pre dbinst`, then CVU shows the correct syntax for the precheck commands for the `dbinst` stage. Enter the `cluvfy -help` command to see detailed CVU command information.

Deprecated and Desupported CLUVFY Commands

The following table includes deprecated and desupported CLUVFY commands:

Table A-2 Deprecated and Desupported Cluvfy Commands

Command	Deprecated	Desupported
<code>cluvfy comp cfs</code>	Oracle Database 12c release 1 (12.1)	No

Special CVU Topics

This section includes the following topics:

- [Generating Fixup Scripts](#)
- [Using CVU to Determine if Installation Prerequisites are Complete](#)

- [Using CVU with Oracle Database 10g Release 1 or 2](#)
- [Entry and Exit Criteria](#)
- [Verbose Mode and UNKNOWN Output](#)
- [CVU Node List Shortcuts](#)

Generating Fixup Scripts

You can use the `-fixup` flag with certain CVU commands to generate [fixup scripts](#) before installation. Oracle Universal Installer can also generate fixup scripts during installation. The installer then prompts you to run the script as `root` in a separate terminal session. If you generate a fixup script from the command line, then you can run it as `root` after it is generated. When you run the script, it raises kernel values to required minimums, if necessary, and completes other operating system configuration.

Alternatively, you can specify the `-method` parameter with certain CVU commands to enable privilege delegation and enable you to run fixup scripts as `root` on remote nodes from the local node.



See Also:

["Privileges and Security"](#)

Using CVU to Determine if Installation Prerequisites are Complete

You can use CVU to determine which system prerequisites for installation are completed. Use this option if you are installing Oracle Database 12c software on a system with a pre-existing Oracle software installation. In using this option, note the following:

- You must run CVU as the user account you plan to use to run the installation. You cannot run CVU as `root`, and running CVU as another user other than the user that is performing the installation does not ensure the accuracy of user and group configuration for installation or other configuration checks.
- Before you can complete a clusterwide status check, SSH must be configured for all cluster nodes. You can use the installer to complete SSH configuration, or you can complete SSH configuration yourself between all nodes in the cluster. You can also use CVU to generate a fixup script to configure SSH connectivity.
- CVU can assist you by finding preinstallation steps that must be completed, but it cannot perform preinstallation tasks.

Use the following syntax to determine what preinstallation steps are completed, and what preinstallation steps you must perform; running the command with the `-fixup` flag generates a fixup script to complete kernel configuration tasks as needed:

```
$ ./runcluvfy.sh stage -pre crsinst -fixup -n node_list
```

In the preceding syntax example, replace the `node_list` variable with the names of the nodes in your cluster, separated by commas. On Windows, you must enclose the comma-delimited node list in double quotation marks (" ").

For example, for a cluster with mountpoint `/mnt/dvdrom/`, and with nodes `node1`, `node2`, and `node3`, enter the following command:

```
$ cd /mnt/dvdrom/  
$ ./runcluvfy.sh stage -pre crsinst -fixup -n node1,node2,node3
```

Review the CVU report, and complete additional steps as needed.

 **See Also:**

Your platform-specific installation guide for more information about installing your product

Using CVU with Oracle Database 10g Release 1 or 2

You can use CVU included on the Oracle Database 12c media to check system requirements for Oracle Database 10g release 1 (10.1) and later installations. To use CVU to check Oracle Clusterware installations, append the command `-r release_code` flag to the standard CVU system check commands.

For example, to perform a verification check before installing Oracle Clusterware version 10. 2 on a system where the media mountpoint is `/mnt/dvdrom` and the cluster nodes are `node1`, `node2`, and `node3`, enter the following command:

```
$ cd /mnt/dvdrom  
$ ./runcluvfy.sh stage -pre crsinst -n node1,node2,node3 -r 10.2
```

 **Note:**

If you do not specify a release version to check, then CVU checks for 12c release 1 (12.1) requirements.

Entry and Exit Criteria

When verifying stages, CVU uses entry and exit criteria. Each stage has entry criteria that define a specific set of verification tasks to be performed before initiating that stage. This check prevents you from beginning a stage, such as installing Oracle Clusterware, unless you meet the Oracle Clusterware prerequisites for that stage.

The exit criteria for a stage define another set of verification tasks that you must perform after the completion of the stage. Post-checks ensure that the activities for that stage have been completed. Post-checks identify stage-specific problems before they propagate to subsequent stages.

Verbose Mode and UNKNOWN Output

Although by default CVU reports in nonverbose mode by only reporting the summary of a test, you can obtain detailed output by using the `-verbose` argument. The `-verbose` argument produces detailed output of individual checks and where applicable shows results for each node in a tabular layout.

If a `cluvfy` command responds with `UNKNOWN` for a particular node, then this is because CVU cannot determine whether a check passed or failed. The cause could be a loss of reachability or the failure of user equivalence to that node. The cause could also be any system problem that was occurring on that node when CVU was performing a check.

The following is a list of possible causes for an `UNKNOWN` response:

- The node is down
- Executables that CVU requires are missing in `Grid_home/bin` or the `Oracle home` directory
- The user account that ran CVU does not have privileges to run common operating system executables on the node
- The node is missing an operating system patch or a required package
- The node has exceeded the maximum number of processes or maximum number of open files, or there is a problem with IPC segments, such as shared memory or semaphores

CVU Node List Shortcuts

To obtain the CVU node list, there are several options you can use with the `cluvfy` command.

To provide CVU a list of all of the nodes of a cluster, enter `-n all`. CVU attempts to obtain the node list in the following order:

1. If vendor clusterware is available, then CVU selects all of the configured nodes from the vendor clusterware using the `lsnodes` utility.

Note:

Starting with Oracle Database 19c, the integration of vendor clusterware with Oracle Clusterware is deprecated, and can be desupported in a future release. For this reason, Oracle recommends that you align your next software or hardware upgrade to transition off of vendor cluster solutions.

2. If Oracle Clusterware is installed, then CVU selects all of the configured nodes from Oracle Clusterware using the `olsnodes` utility.
3. If neither the vendor clusterware nor Oracle Clusterware is installed, then CVU searches for a value for the `CV_NODE_ALL` key in the configuration file.
4. If vendor clusterware and Oracle Clusterware are not installed and no key named `CV_NODE_ALL` exists in the configuration file, then CVU searches for a value for the `CV_NODE_ALL` environmental variable. If you have not set this variable, then CVU reports an error.

To provide a partial node list, you can set an environmental variable and use it in the CVU command. For example, on Linux or Unix systems you can enter:

```
setenv MYNODES node1,node3,node5
cluvfy comp nodecon -n $MYNODES [-verbose]
```

Cluster Verification Utility Command Reference

This section lists and describes CVU commands.

cluvfy comp acfs

Checks the integrity of Oracle Automatic Storage Management Cluster File System (Oracle ACFS) on all nodes in a cluster.

Syntax

```
cluvfy comp acfs [-n node_list] [-f file_system] [-verbose]
```

Parameters

Table A-3 cluvfy comp acfs Command Parameters

Parameter	Description
-n <i>node_list</i>	The comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
-f <i>file_system</i>	The name of the file system to check.
-verbose	CVU prints detailed output.

cluvfy comp admprv

Checks the required administrative privileges for the operation specified by -o parameter on all the nodes that you specify in the node list.

Syntax

On Linux and UNIX platforms:

```
cluvfy comp admprv [-n node_list] -o user_equiv [-sshonly] | -o crs_inst
  [-asmgrp asmadmin_group] [-asmdbagrp asmdba_group] [-orainv
  orainventory_group]
  [-fixup] [-fixupnoexec] [-method {sudo -user user_name [-location
  directory_path] | root}]
  | -o db_inst [-osdba osdba_group] [-osoper osoper_group] [-fixup] [-
  fixupnoexec]
  [-method {sudo -user user_name [-location dir_path] | root}] |
  -o db_config -d oracle_home [-fixup] [-fixupnoexec]
  [-method {sudo -user user_name [-location directory_path] | root}] [-
  verbose]
```

On Windows platforms:

```
cluvfy comp admprv [-n node_list] | -o user_equiv | -o crs_inst [-fixup] [-fixupnoexec]
  | -o db_inst [-fixup] [-fixupnoexec] | -o db_config -d oracle_home [-fixup] [-fixupnoexec]
  [-verbose]
```

Parameters

Table A-4 cluvfy comp admprv Command Parameters

Parameter	Description
<code>-n node_list</code>	The comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
<code>-o user_equiv [-sshonly]</code>	Checks user equivalence between the nodes. On Linux and UNIX platforms, you can optionally verify user equivalence using <code>ssh</code> by adding the <code>-sshonly</code> parameter.
<code>-o crs_inst [option]</code>	Checks administrative privileges for installing Oracle Clusterware. Optionally, you can specify the following: <ul style="list-style-type: none"> <code>-asmgrp</code>: Specify the name of the OSASM group. The default is <code>asmadmin</code>. <code>-asmdbagr</code>: Specify the name of the ASMDBA group. The default is <code>asmdba</code>. <code>-orainv</code>: Specify the name of the Oracle Inventory group. The default is <code>oinstall</code>.
<code>-o db_inst option]</code>	Checks administrative privileges for installing an Oracle RAC database. Optionally, you can specify the following: <ul style="list-style-type: none"> <code>-osdba</code>: The name of the OSDBA group. The default is <code>dba</code>. <code>-osoper</code>: The name of the OSOPER group.
<code>-o db_config -d oracle_home</code>	Checks administrative privileges for creating or configuring an Oracle RAC database. Specify the location of the Oracle home for the Oracle RAC database.
<code>-fixup</code>	Specifies that if the verification fails, then CVU generates fixup instructions, if feasible.
<code>-fixupnoexec</code>	Specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
<code>-method {sudo -user user_name [-location dir_path] root}</code>	Specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
<code>-verbose</code>	CVU prints detailed output.

Usage Notes

- The operations following the `-o` parameter are mutually exclusive and you can specify only one operation at a time.

- By default, the equivalence check does not verify X-Windows configurations, such as whether you have disabled X-forwarding, whether you have the proper setting for the `DISPLAY` environment variable, and so on.

To verify X-Windows aspects during user equivalence checks, set the `CV_XCHK_FOR_SSH_ENABLED` key to `TRUE` in the configuration file that resides in the `CV_HOME/cv/admin/cvu_config` directory before you run the `cluvfy comp admprv -o user_equiv` command.

Examples

You can verify that the permissions required for installing Oracle Clusterware have been configured on the nodes `racnode1` and `racnode2` by running the following command:

```
$ cluvfy comp admprv -n racnode1,racnode2 -o crs_inst -verbose
```

You can verify that the permissions required for creating or modifying an Oracle RAC database using the `C:\app\oracle\product\12.2.0\dbhome_1` Oracle home directory, and generate a script to configure the permissions by running the following command:

```
cluvfy comp admprv -n racnode1,racnode2 -o db_config -d C:\app\oracle  
\product\12.2.0\dbhome_1 -fixup -verbose
```

cluvfy comp asm

Checks the integrity of Oracle Automatic Storage Management (Oracle ASM) on specific nodes in the cluster.

This check ensures that the Oracle ASM instances on the specified nodes are running from the same Oracle home and that `asmlib`, if it exists, has a valid version and ownership.

Syntax

```
cluvfy comp asm [-n node_list] [-verbose]
```

Usage Notes

This command takes only a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node. You can also specify `-verbose` to print detailed output.

Examples

This command produces output similar to the following:

```
$ cluvfy comp asm -n all
```

```
Verifying ASM Integrity
```

```
Task ASM Integrity check started...
```

```
Starting check to see if ASM is running on all cluster nodes...
```

```
ASM Running check passed. ASM is running on all specified nodes

Starting Disk Groups check to see if at least one Disk Group configured...
Disk Group Check passed. At least one Disk Group configured

Task ASM Integrity check passed...

Verification of ASM Integrity was successful.
```

cluvfy comp baseline

Captures system and cluster configuration information to create a baseline. You can use this baseline for comparison with the state of the system. You can collect baselines at strategic times, such as after Oracle Clusterware installation, before and after upgrading Oracle Clusterware, or automatically as part of periodic execution of CVU running as an Oracle Clusterware resource. You can also compare several baselines.

Syntax

```
cluvfy comp baseline -collect {all | cluster | database | asm} [-n
node_list]
    [-d Oracle_home] [-db db_unique_name] [-bestpractice | -mandatory] [-
binlibfilesonly
    [-reportname report_name] [-savedir save_dir]
    [-method {sudo -user user_name [-location directory_path] | root}]
cluvfy comp baseline -compare baseline1,baseline2,... [-cross_compare] [-
deviations] [-savedir save_dir]
```

Parameters

Table A-5 cluvfy comp baseline Command Parameters

Parameter	Description
-collect {all cluster database asm}	The -collect parameter instructs CVU to create a baseline and save it in the <i>Grid_home/cv/report/xml</i> directory. You can collect a baseline related to Oracle Clusterware (cluster), the database (database), Oracle ASM, or all three (all).
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which the test should be conducted.
-d <i>Oracle_home</i>	When collecting a database baseline, if you specify an Oracle home, then CVU collects baselines for all the databases running from the Oracle home. Use the -db parameter to collect a baseline for a specific database.

Table A-5 (Cont.) cluvfy comp baseline Command Parameters

Parameter	Description
<code>-db db_unique_name</code>	The name of the database for which you want to collect information. When collecting a database baseline, if you specify the <code>-db</code> parameter, then CVU only collects the baseline for the specified database. If you do not specify <code>-db</code> , then CVU discovers all of the cluster databases configured in the system and the collects baselines for each of those.
<code>-bestpractice</code> <code>-mandatory</code>	Specify <code>-bestpractice</code> to collect a baseline for only best practice recommendations. Specify <code>-mandatory</code> to collect a baseline for only mandatory requirements.
<code>-binlibfilesonly</code>	Specify <code>-binlibfilesonly</code> to collect only files in the <code>bin/</code> , <code>lib/</code> , and <code>jlib/</code> subdirectories of the software home.
<code>-report report_name</code>	Use this optional parameter to specify a name for the report.
<code>-savedir save_dir</code>	Use this optional parameter to specify a location in which CVU saves the reports. If you do not specify the <code>-savedir</code> option, then CVU saves the reports in the <code>Grid_home/cv/report</code> directory.
<code>-method {sudo -user user_name [-location dir_path] root}</code>	Specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
<code>-compare baseline1,baseline2, ...</code>	Specify <code>-compare</code> to compare baselines. If you specify only one baseline, then CVU displays the results of the collections. If you specify multiple baselines in a comma-delimited list, then CVU compares the values from the baselines against each other in an HTML document.
<code>-cross_compare</code>	Specify <code>-cross_compare</code> to compare baselines across clusters or across cluster nodes and databases.
<code>-deviations</code>	Optionally, you can specify this parameter to display only the deviations from best practice recommendations or mandatory requirements, or both, (depending on whether you specified the <code>-bestpractice</code> and <code>-mandatory</code> parameters).

Usage Notes

- You must specify either the `-collect` or `-compare` parameter.
- Items that CVU collects when running this command include:
 - Physical memory
 - Available memory
 - Swap space
 - Free space
 - Required packages
 - Recommended kernel parameters
 - `/etc/inittab` permissions
 - Domain sockets under `/var/tmp/.oracle`
 - Oracle Clusterware software file attributes
 - Network MTU size

OCR permissions, group, and owner (if OCR is stored on a shared file system)
 OCR disk group (if OCR is stored on Oracle ASM)
 System requirement pluggable tasks (Zeroconf settings, /boot mount, Huge Pages existence, 8888 port availability, Ethernet jumbo frames)
 Oracle Clusterware post-check pluggable tasks (css miscount, reboottime, disktimeout)¹
 Database best practices

Examples

The following examples illustrate usage for both `-collect` and `-compare` command parameters:

```
$ cluvfy comp baseline -collect all -n all -db orcl -bestpractice -report
b11
  -savedir /tmp

$ cluvfy comp baseline -compare b11,b12
```

cluvfy comp clocksync

Checks clock synchronization across all the nodes in the node list. CVU verifies a time synchronization service is running (Oracle Cluster Time Synchronization Service (CTSS) or Network Time Protocol (NTP)), that each node is using the same reference server for clock synchronization, and that the time offset for each node is within permissible limits.

Syntax

```
cluvfy comp clocksync [-noctss] [-n node_list] [-verbose]
```

Parameters

Table A-6 cluvfy comp clocksync Command Parameters

Parameter	Description
<code>-noctss</code>	If you specify this parameter, then CVU does not perform a check on CTSS. Instead, CVU checks the platform's native time synchronization service, such as NTP.
<code>-n <i>node_list</i></code>	The comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
<code>-verbose</code>	CVU prints detailed output.

cluvfy comp clumgr

Checks the integrity of the cluster manager subcomponent, or Oracle Cluster Synchronization Services (CSS), on the nodes in the node list.

¹ Applicable only on UNIX platforms

Syntax

```
cluvfy comp clumgr [-n node_list] [-verbose]
```

Usage Notes

You can specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.

You can also choose to print detailed output.

cluvfy comp crs

Checks the integrity of the Cluster Ready Services (CRS) daemon on the specified nodes.

Syntax

```
cluvfy comp crs [-n node_list] [-verbose]
```

Usage Notes

Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this parameter, then CVU checks only the local node.

You can also choose to print detailed output.

cluvfy comp dhcp

Verifies that the DHCP server exists on the network, and that it can provide a required number of IP addresses.

The required number of IP addresses is calculated, as follows:

- Regardless of the size of the cluster, there must be three scan VIPs
- One node VIP for each node you specify with the `-n` option
- One application VIP for each application VIP resource you specify with the `-vipresname` option

This verification also verifies the response time for the DHCP server.

Syntax

```
cluvfy comp dhcp -clustername cluster_name [-vipresname  
application_vip_resource_name]  
[-port dhcp_port] [-n node_list] [-method {sudo -user user_name [-  
location directory_path] | root}]  
[-networks network_list] [-verbose]
```

Parameters

Table A-7 clufy comp dhcp Command Parameters

Parameter	Description
-clustername <i>cluster_name</i>	You must specify the name of the cluster of which you want to check the integrity of DHCP.
-vipresname <i>application_vip_resource_name</i>	Optionally, you can specify a comma-delimited list of the names of the application VIP resource.
-port <i>dhcp_port</i>	Optionally, you can specify the port to which DHCP packages are sent. The default port is 67.
-n <i>node_list</i>	Optionally, you can specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this parameter, then CVU checks only the local node.
-method {sudo -user <i>user_name</i> [-location <i>directory_path</i>] root}	Optionally, you can specify whether the privilege delegation method is sudo or root, for root user access. If you specify sudo, then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the sudo executable.
-networks <i>network_list</i>	Optionally, you can specify a list of network classifications for the cluster separated by forward slashes (/) that you want CVU to check, where each network is in the form of " <i>if_name</i> "[: <i>subnet_id</i> [: <i>if_type</i> [, <i>if_type</i> ...]]]. In the preceding format, you must enclose <i>if_name</i> in double quotation marks (" "), and you can use regular expressions, such as ".*", as in "eth*", to match interfaces like eth1 and eth02. The <i>subnet_id</i> is the subnet number of the network interface. The <i>if_type</i> is a comma-separated list of interface types: {CLUSTER_INTERCONNECT PUBLIC ASM}.
-verbose	CVU prints detailed output.

Usage Notes

- You must run this command as `root`.
- Do not run this check while the default network Oracle Clusterware resource, configured to use a DHCP-provided IP address, is online (because the VIPs get released and, since the cluster is online, DHCP has provided IP, so there is no need to double the load on the DHCP server).
- Run this check on the local node. This is unlike other CVU commands, which run on all nodes specified in node list. As a result, even though the local node is not included in the node list you specify with the `-n` option, the error messages get reported to the local node.
- Before running this command, ensure that the network resource is offline. Use the `srvctl stop nodeapps` command to bring the network resource offline, if necessary.

 **See Also:**

Oracle Real Application Clusters Administration and Deployment Guide
for more information about the `srvctl stop nodeapps` command

cluvfy comp dns

Verifies that the Grid Naming Service (GNS) subdomain delegation has been properly set up in the Domain Name Service (DNS) server.

Syntax

```
cluvfy comp dns -server -domain gns_sub_domain -vipaddress gns_vip_address
[-port dns_port]
[-method {sudo -user user_name [-location directory_path] | root}] [-
verbose]
```

```
cluvfy comp dns -client -domain gns_sub_domain -vip gns_vip [-port
dns_port]
[-last] [-method {sudo -user user_name [-location directory_path] |
root}] [-verbose]
```

Parameters

Table A-8 cluvfy comp dns Command Parameters

Parameter	Description
-server	Start a test DNS server for the GNS subdomain that listens on the domain specified by the -domain option.
-client	Validate connectivity to a test DNS server started on a specific address. You must specify the same information you specified when you started the DNS server.
-domain <i>gns_sub_domain</i>	Specify the name of the GNS subdomain.
-vipaddress <i>gns_vip_address</i>	Specify the GNS virtual IP address in the form { <i>ip_name</i> <i>ip_address</i> }/ <i>net_mask</i> / <i>interface_name</i> . You can specify either <i>ip_name</i> , which is a name that resolves to an IP address, or <i>IP_address</i> , which is an IP address. Either name or address is followed by <i>net_mask</i> , which is the subnet mask for the IP address, and <i>interface_name</i> , which is the interface on which to start the IP address.
-vip <i>gns_vip</i>	Specify a GNS virtual IP address, which is either a name that resolves to an IP address or a dotted decimal numeric IP address.
-port <i>dns_port</i>	Specify the port on which the test DNS server listens. The default port is 53.
-last	Optionally, you can use this parameter to send a termination request to the test DNS server after all the validations are complete.

Table A-8 (Cont.) cluvfy comp dns Command Parameters

Parameter	Description
<code>-method {sudo -user user_name [-location directory_path] root}</code>	Optionally, you can specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
<code>-verbose</code>	CVU prints detailed output.

Usage Notes

- You must run this command as `root`.
- Run `cluvfy comp dns -server` on one node of the cluster.
- Run `cluvfy comp dns -client` on each node of the cluster to verify DNS server setup for the cluster.
- On the last node, specify the `-last` option to terminate the `cluvfy comp dns -server` instance.
- Do not run this command while the GNS resource is online.
- Oracle does not support this command on Windows.

cluvfy comp freespace

Checks the free space available in the Oracle Clusterware home storage and ensure that there is at least 5% of the total space available. For example, if the total storage is 10GB, then the check ensures that at least 500MB of it is free.

Syntax

```
cluvfy comp freespace [-n node_list]
```

If you choose to include the `-n` option, then enter a comma-delimited list of node names on which to run the command.

cluvfy comp gns

Verifies the integrity of the Grid Naming Service (GNS) on the cluster.

Syntax

```
cluvfy comp gns -precrsinst {-vip gns_vip [-domain gns_domain] | -
clientdata file_name}
[-networks network_list] [-n node_list] [-verbose]
cluvfy comp gns -postcrsinst [-verbose]
```


Parameters

Table A-9 cluvfy comp gns Command Parameters

Parameter	Description
-precrsinst	Use this parameter to perform checks on the GNS domain name and VIP address before Oracle Clusterware is installed.
-vip <i>gns_vip</i>	Specify the GNS virtual IP address. When you specify <code>-vip</code> and <code>-domain</code> together, CVU validates that this cluster can become a GNS server (local GNS).
-domain <i>gns_domain</i>	Optionally, you can specify the GNS subdomain name.
-clientdata <i>file_name</i>	Specify the name of the file that contains the GNS credentials. CVU validates that this cluster can use the specified client data to become a client GNS cluster of another GNS server cluster (shared GNS).
-networks <i>network_list</i>	Specify a list of network classifications for the cluster, including public networks for GNS, separated by forward slashes (/) that you want CVU to check, where each network is in the form of " <i>if_name</i> "[: <i>subnet_id</i> [: <i>if_type</i> [, <i>if_type</i> ...]]]. In the preceding format, you must enclose <i>if_name</i> in double quotation marks (" "), and you can use regular expressions, such as ".*", as in "eth*", to match interfaces like eth1 and eth02. The <i>subnet_id</i> is the subnet number of the network interface. The <i>if_type</i> is a comma-separated list of interface types: {CLUSTER_INTERCONNECT PUBLIC ASM}.
-n <i>node_list</i>	The comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
-postcrsinst	Use this parameter to check the integrity of GNS on all nodes in the cluster. If you use this parameter, then you can use no other parameters with the exception of <code>-verbose</code> .
-verbose	CVU prints detailed output.

cluvfy comp gnp

Checks the integrity of Grid Plug and Play on a list of nodes in a cluster.

Syntax

```
cluvfy comp gnp [-n node_list] [-verbose]
```

Optionally, you can specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.

You can also choose verbose output from CVU.

cluvfy comp ha

Checks the integrity of Oracle Restart on the local node.

Syntax

```
cluvfy comp ha [-verbose]
```

If you include the `-verbose` option, then CVU prints detailed output.

cluvfy comp healthcheck

Checks your Oracle Clusterware and Oracle Database installations for their compliance with mandatory requirements and best practices guidelines, and to ensure that they are functioning properly.

Syntax

```
cluvfy comp healthcheck [-collect {cluster|database|asm}] [-db
db_unique_name]
[-bestpractice|-mandatory] [-deviations] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-10 cluvfy comp healthcheck Command Parameters

Parameter	Description
<code>-collect {cluster database asm}</code>	Use <code>-collect</code> to specify that you want to perform checks for Oracle Clusterware (<code>cluster</code>), Oracle Database (<code>database</code>), or Oracle ASM. If you do not use the <code>-collect</code> flag with the <code>healthcheck</code> option, then CVU performs checks for all three.

Table A-10 (Cont.) cluvfy comp healthcheck Command Parameters

Parameter	Description
-db <i>db_unique_name</i>	<p>Use <code>-db</code> to specify checks on the specific database that you enter after the <code>-db</code> flag.</p> <p>CVU uses JDBC to connect to the database as the user <code>dbsnmp</code> to verify various database parameters. For this reason, if you want CVU to perform checks for the database you specify with the <code>-db</code> flag, then you must first create the <code>dbsnmp</code> user on that database, and grant that user the CVU-specific role, <code>cvusapp</code>. You must also grant members of the <code>cvusapp</code> role <code>select</code> permissions on system tables.</p> <p>There is a SQL script included in the <code>CVU_home/cv/admin/cvsys.sql</code> directory to facilitate the creation of this user. Use this SQL script to create the <code>dbsnmp</code> user on all the databases that you want to verify using CVU.</p> <p>If you use the <code>-db</code> flag but do not provide a database unique name, then CVU discovers all the Oracle Databases on the cluster. To perform best practices checks on these databases, you must create the <code>dbsnmp</code> user on each database, and grant that user the <code>cvusapp</code> role with the <code>select</code> privileges needed to perform the best practice checks.</p>
[-bestpractice -mandatory] [-deviations]	<p>Use the <code>-bestpractice</code> flag to specify best practice checks, and the <code>-mandatory</code> flag to specify mandatory checks. Add the <code>-deviations</code> flag to specify that you want to see only the deviations from either the best practice recommendations or the mandatory requirements. You can specify either the <code>-bestpractice</code> or <code>-mandatory</code> flag, but not both flags. If you specify neither <code>-bestpractice</code> nor <code>-mandatory</code>, then CVU displays both best practices and mandatory requirements.</p>
-html	<p>Use the <code>-html</code> flag to generate a detailed report in HTML format.</p> <p>If you specify the <code>-html</code> flag, and a browser CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete.</p> <p>If you do not specify the <code>-html</code> flag, then CVU generates the detailed report in a text file.</p>
-save [-savedir <i>directory_path</i>]	<p>Use the <code>-save</code> or <code>-save -savedir</code> flags to save validation reports (<code>cvuheckdcreport_timestamp.txt</code> and <code>cvuheckreport_timestamp.htm</code>), where <i>timestamp</i> is the time and date of the validation report.</p> <p>If you use the <code>-save</code> flag by itself, then CVU saves the reports in the <code>CVU_home/cv/report</code> directory, where <i>CVU_home</i> is the location of the CVU binaries.</p> <p>If you use the <code>-save -savedir</code> flags, then specify a directory where you want CVU to save the reports.</p>

cluvfy comp nodeapp

Checks for the existence of node applications, namely VIP, NETWORK, and ONS, on all of the specified nodes.

Syntax

```
cluvfy comp nodeapp [-n node_list] [-verbose]
```

Optionally, you can specify a comma-delimited list of non domain-qualified node names on which to conduct the check. If you do not specify this option, then CVU checks only the local node.

You can also choose verbose output from CVU.

cluvfy comp nodecon

Checks the connectivity among the nodes specified in the node list. If you provide an interface list, then CVU checks the connectivity using only the specified interfaces.

Syntax

```
cluvfy comp nodecon [-n node_list] [-networks network_list] [-i  
interface_list] [-verbose]
```

Parameters

Table A-11 cluvfy comp nodecon Command Parameters

Parameter	Description
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification.
-networks <i>network_list</i>	Specify a forward slash (/)-delimited list of networks on which to conduct the verification.
-i <i>interface_list</i>	Specify a comma-delimited list of interface names. If you do not specify this option, then CVU discovers the available interfaces and checks connectivity using each of them.
-verbose	CVU prints detailed output.

Usage Notes

- You can run this command in verbose mode to identify the mappings between the interfaces, IP addresses, and subnets.
- On Solaris platforms, this command skips testing IP addresses that are marked as deprecated.
- Use the `nodecon` command without the `-networks` parameter and with `-n` set to `all` to use CVU to:
 - Discover all of the network interfaces that are available on the cluster nodes
 - Review the interfaces' corresponding IP addresses and subnets
 - Obtain the list of interfaces that are suitable for use as VIPs and the list of interfaces to private interconnects
 - Verify the connectivity between all of the nodes through those interfaces

Examples

Example A-1 Verifying the connectivity between nodes through specific network interfaces

To verify the connectivity between the nodes `node1` and `node3` through interface `eth0`:

```
cluvfy comp nodecon -n node1/node3 -networks eth0 -verbose
```

Example A-2 Discovering all available network interfaces and verifying the connectivity between the nodes in the cluster through those network interfaces

Use the following command to discover all of the network interfaces that are available on the cluster nodes. CVU then reviews the interfaces' corresponding IP addresses and subnets. Using this information, CVU obtains a list of interfaces that are suitable for use as VIPs and a list of interfaces to private interconnects. Finally, CVU verifies the connectivity between all of the nodes in the cluster through those interfaces.

```
cluvfy comp nodecon -n all -verbose
```

cluvfy comp nodereach

Determines whether a source node can communicate with other, specific nodes.

Syntax

```
cluvfy comp nodereach -n node_list [-srcnode node] [-verbose]
```

Parameters

Table A-12 cluvfy comp nodereach Command Parameters

Parameter	Description
<code>-n <i>node_list</i></code>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification.
<code>-srcnode <i>node</i></code>	Optionally, specify the name of the source node from which CVU performs the reachability test. If you do not specify a source node, then CVU uses the node on which you run the command as the source node.
<code>-verbose</code>	CVU prints detailed output.

Example

To verify that `node3` is reachable over the network from the local node, use the following command:

```
cluvfy comp nodereach -n node3
```

This command produces output similar to the following:

```
Verifying node reachability

Checking node reachability...
Node reachability check passed from node "node1"

Verification of node reachability was successful.
```

cluvfy comp ocr

Checks the integrity of Oracle Cluster Registry (OCR) on all specified nodes.

Syntax

```
cluvfy comp ocr [-n node_list] [-method {sudo -user user_name [-location
directory_path] | root}] [-verbose]
```

Parameters

Table A-13 cluvfy comp ocr Command Parameters

Parameter	Description
-n <i>node_list</i>	The comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
-method {sudo -user <i>user_name</i> [-location <i>directory_path</i>] root}	Specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
-verbose	CVU prints detailed output.

Usage Notes

This command does not verify the integrity of OCR contents. You must use the OCRCHECK utility to verify the contents of OCR.

Example

To verify the integrity of OCR on the local node, run the following command:

```
cluvfy comp ocr
```

This command returns output similar to the following:

```
Verifying OCR integrity

Checking OCR integrity...
```

```
Checking the absence of a non-clustered configuration...
All nodes free of non-clustered, local-only configurations

ASM Running check passed. ASM is running on all specified nodes

Checking OCR config file "/etc/oracle/ocr.loc"...

OCR config file "/etc/oracle/ocr.loc" check successful

Disk group for ocr location "+DATA" available on all the nodes

NOTE:
This check does not verify the integrity of the OCR contents. Execute
'ocrcheck' as a privileged user to verify the contents of OCR.

OCR integrity check passed

Verification of OCR integrity was successful.
```

cluvfy comp ohasd

Checks the integrity of the Oracle High Availability Services daemon.

Syntax

```
cluvfy comp ohasd [-n node_list] [-verbose]
```

Usage Notes

Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. You can use `all` to specify all nodes. If you do not specify this option, then CVU checks only the local node.

You can also print detailed output.

Example

To verify that the Oracle High Availability Services daemon is operating correctly on all nodes in the cluster, use the following command:

```
cluvfy comp ohasd -n all -verbose
```

This command returns output similar to the following:

```
Verifying OHASD integrity

Checking OHASD integrity...
ohasd is running on node "node1"
ohasd is running on node "node2"
ohasd is running on node "node3"
ohasd is running on node "node4"
```

```
OHASD integrity check passed

Verification of OHASD integrity was successful.
```

cluvfy comp olr

Checks the integrity of Oracle Local Registry (OLR) on the local node.

Syntax

```
cluvfy comp olr [-n node_list] [-verbose]
```

Usage Notes

- Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. You can use `all` to specify all nodes. If you do not specify this option, then CVU checks only the local node.
- You can also print detailed output.
- This command does not verify the integrity of the OLR contents. You must use the `ocrcheck -local` command to verify the contents of OLR.

Example

To verify the integrity of the OLR on the current node, run the following command:

```
cluvfy comp olr -verbose
```

This command returns output similar to the following:

```
Verifying OLR integrity
Checking OLR integrity...
Checking OLR config file...
OLR config file check successful

Checking OLR file attributes...
OLR file check successful

WARNING:
This check does not verify the integrity of the OLR contents. Execute
'ocrcheck -local' as a privileged user to verify the contents of OLR.

OLR integrity check passed

Verification of OLR integrity was successful.
```


cluvfy comp peer

Checks the compatibility and properties of the specified nodes against a reference node.

You can check compatibility for non-default user group names and for different releases of the Oracle software. This command compares physical attributes, such as memory and swap space, user and group values, kernel settings, and installed operating system packages.

Syntax

```
cluvfy comp peer -n node_list [-refnode node] [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2}]  
[-orainv orainventory_group] [-osdba osdba_group] [-verbose]
```

Parameters

Table A-14 cluvfy comp peer Command Parameters

Parameter	Description
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification.
-refnode <i>node</i>	Optionally, specify a node that CVU uses as a reference for checking compatibility with other nodes. If you do not specify this option, then CVU reports values for all the nodes in the node list.
-r {10.1 10.2 11.1 11.2 12.1 12.2}	Optionally, specify the software release that CVU checks as required for installation of Oracle Clusterware or Oracle RAC. If you do not specify this option, then CVU assumes Oracle Clusterware 12c or Oracle Database 12c.
-orainv <i>orainventory_group</i>	Optionally, you can specify the name of the Oracle Inventory group. If you do not specify this option, then CVU uses oinstall as the inventory group. Note: This parameter is not available on Windows systems.
-osdba <i>osdba_group</i>	Optionally, you can specify the name of the OSDBA group. If you do not specify this option, then CVU uses dba as the OSDBA group. Note: This parameter is not available on Windows systems.
-verbose	CVU prints detailed output.

Usage Notes

Peer comparison with the `-refnode` option compares the system properties of other nodes against the reference node. If the value of the other node is not greater than the value for that of the reference node, then CVU flags that comparison as a deviation from the reference node. If a group or user exists on neither the reference node nor on the other node, then CVU reports a match to indicate that there is no deviation from the reference node. Similarly, CVU reports as mismatched a comparison with a node that has less total memory than the reference node.

Example

The following command lists the values of several preselected properties on different nodes from Oracle Database 12c:

```
cluvfy comp peer -n node1,node2,node4,node7 -verbose
```

cluvfy comp scan

Checks the Single Client Access Name (SCAN) configuration.

Syntax

```
cluvfy comp scan [-verbose]
```

Usage Notes

Optionally, you can include `-verbose` to print detailed output.

Example

To verify that the SCAN and SCAN listeners are configured and operational on all nodes in the cluster, use the following command:

```
$ cluvfy comp scan
```

This command returns output similar to the following:

```
Verifying scan

Checking Single Client Access Name (SCAN)...

Checking TCP connectivity to SCAN Listeners...
TCP connectivity to SCAN Listeners exists on all cluster nodes

Checking name resolution setup for "node1.example.com"...

Verification of SCAN VIP and Listener setup passed

Verification of scan was successful.
```

cluvfy comp software

Checks the files and attributes installed with the Oracle software.

Syntax

```
cluvfy comp software [-n node_list] [-d oracle_home] [-r {10.1 | 10.2 |
11.1 | 11.2 | 12.1 | 12.2}
  [-allfiles] [-verbose]
```

Parameters

Table A-15 cluvfy comp software Command Parameters

Parameter	Description
<code>-n node_list</code>	Optionally, you can specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
<code>-d oracle_home</code>	Optionally, you can specify the directory where the Oracle Database software is installed. If you do not specify this option, then CVU verifies the files installed in the Grid home.
<code>-r {10.1 10.2 11.1 11.2 12.1 12.2}</code>	Optionally, you can specify the software release that CVU checks as required for installation of Oracle Clusterware or Oracle RAC. If you do not specify this option, then CVU assumes Oracle Clusterware 12c or Oracle Database 12c.
<code>-allfiles</code>	If you specify this parameter, then CVU checks the attributes of all files of the specified Oracle home. If you do not specify this parameter, then CVU checks the attributes of the <code>lib</code> , <code>jlib</code> , and <code>bin</code> files under the specified Oracle home.
<code>-verbose</code>	CVU prints detailed output.

Usage Notes

Add additional information about the command here.

Example

To verify that the installed files for Oracle Clusterware 12c are configured correctly, use a command similar to the following:

```
$ cluvfy comp software -n all -verbose
```

This command returns output similar to the following:

```
Verifying software
Check: Software
    1021 files verified
Software check passed
Verification of software was successful.
```

cluvfy comp space

Checks for free disk space at the location you specify in the `-l` parameter on all the specified nodes.

Syntax

```
clufy comp space [-n node_list] -l storage_location -z disk_space {B | K  
| M | G} [-verbose]
```

Parameters

Table A-16 clufy comp space Command Parameters

Parameter	Description
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this parameter, then CVU checks only the local node.
-l <i>storage_location</i>	Specify the directory path to the storage location you want to check.
-z <i>disk_space</i> {B K M G}	Specify the required disk space, in units of bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). There should be no space between the numeric value and the byte indicator; for example, 2G. Use only whole numbers.
-verbose	CVU prints detailed output.

Usage Notes

The `space` component does not support block or raw devices.



See Also:

The Oracle Certification site on My Oracle Support for the most current information about certified storage options:

<https://support.oracle.com>

Example

You can verify that each node has 5 GB of free space in the `/home/dbadmin/products` directory by running the following command:

```
$ clufy comp space -n all -l /home/dbadmin/products -z 5G -verbose
```

clufy comp ssa

Use the `clufy comp ssa` component verification command to discover and check the sharing of the specified storage locations. CVU checks sharing for nodes in the node list.

Syntax

```
clufy comp ssa [-n node_list | -allnodes] [-s storage_path_list]  
[-t {gi_software | rac_software | data | ocr_vdisk}] [-asm] [-asmdev]
```

```
asm_device_list]
[-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18.1 | 19.1}] [-verbose]
```

Parameters

Table A-17 cluvfy comp ssa Command Parameters

Parameter	Description
-n <i>node_list</i> -allnodes	The comma-delimited list of non domain-qualified node names on which to conduct the verification or all nodes. If you do not specify any of these options, then CVU checks only the local node.
-s <i>storage_path_list</i>	A comma-delimited list of storage paths, for example, /dev/sda,/dev/sdb. If you do not specify the -s option, then CVU discovers supported storage types and checks sharing for each of them.
-t { <i>gi_software</i> <i>rac_software</i> <i>data</i> <i>ocr_vdisk</i> }	The type of Oracle files (Oracle Grid Infrastructure binaries, Oracle RAC, or Oracle Database binaries) that will be stored on the storage device. If you do not specify -t, then CVU discovers or checks the data file type.
-asm	Specify this parameter to discover all storage suitable for use by Oracle ASM.
-asmdev <i>asm_device_list</i>	A comma-delimited list of Oracle ASM devices for which you want to check sharing of storage locations. If the list contains shell metacharacters, then enclose the list in double quotation marks (" ").
-r {10.1 10.2 11.1 11.2 12.1 12.2 18 19 20}	Optionally, you can specify the release number of the product for which you are running the verification. If you do not specify -r, then CVU runs the verification for 20.
-verbose	CVU prints detailed output.

Usage Notes

- The current release of `cluvfy` has the following limitations on Linux regarding shared storage accessibility check.
 - Currently NAS storage and OCFS2 (version 1.2.1 or higher) are supported.

See Also:

Oracle Grid Infrastructure Installation Guide for more information about NAS mount options

- When checking sharing on NAS, `cluvfy` commands require that you have write permission on the specified path. If the `cluvfy` user does not have write permission, `cluvfy` reports the path as not shared.
- To perform discovery and shared storage accessibility checks for SCSI disks on Linux systems, CVU requires the CVUQDISK package. If you attempt to use CVU

and the CVUQDISK package is not installed on all of the nodes in your Oracle RAC environment, then CVU responds with an error. See "[Shared Disk Discovery on Red Hat Linux](#)" for information about how to install the CVUQDISK package.

Examples

To discover all of the shared storage systems available on your system:

```
$ cluvfy comp ssa -allnodes -verbose
```

To discover all the storage suitable for use by Oracle ASM, based on the specified Oracle ASM discovery string:

```
$ cluvfy comp ssa -n node1,node2 -asm -asmdev "/dev/xda*"
```

You can verify the accessibility of specific storage locations, such as an Oracle ASM disk group called OCR13, for storing data files for all the cluster nodes by running a command similar to the following:

```
$ cluvfy comp ssa -allnodes -s OCR13
```

This command produces output similar to the following:

```
Verifying shared storage accessibility
Checking shared storage accessibility...
"OCR13" is shared
Shared storage check was successful on nodes "node1,node2,node3,node4"
Verification of shared storage accessibility was successful.
```

cluvfy comp sys

Checks that the minimum system requirements are met for the specified product on all the specified nodes.

Syntax

```
cluvfy comp sys [-n node_list | -flex -hub hub_list]
  -p {crs | ha | database} [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 |
18.1 | 19.1}] [-osdba osdba_group]
  [-orainv orainventory_group] [-fixup] [-fixupnoexec] [-method {sudo -
user user_name
  [-location directory_path] | root}] [-verbose]
```

Parameters

Table A-18 cluvfy comp sys Command Parameters

Parameter	Description
-n <i>node_list</i> -flex -hub <i>hub_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node. Alternatively, you can specify a list of Hub Nodes on which to conduct the verification.
-p {crs ha database}	Specifies whether CVU checks the system requirements for Oracle Clusterware, Oracle Restart (HA), or Oracle RAC. Note: Oracle does not support Oracle Restart for Oracle Database 10g. If you use the -p ha option with -r 10.1 10.2, then CVU returns an error.
-r {10.1 10.2 11.1 11.2 12.1 12.2 18.1 19.1}	Specifies the Oracle Database release that CVU checks as required for installation of Oracle Clusterware or Oracle RAC. If you do not specify this option, then CVU assumes Oracle Database 19c.
-osdba <i>osdba_group</i>	The name of the OSDBA group. If you do not specify this option, then CVU uses dba as the OSDBA group.
-orainv <i>orainventory_group</i>	The name of the Oracle Inventory group. If you do not specify this option, then CVU uses oinstall as the inventory group.
-fixup	Specifies that if the verification fails, then CVU performs fixup operations, if feasible.
-fixupnoexec	Specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
-method {sudo -user <i>user_name</i> [-location <i>directory_path</i>] root}	Specify whether the privilege delegation method is sudo or root, for root user access. If you specify sudo, then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the sudo executable.
-verbose	CVU prints detailed output.

Example

To verify the system requirements for installing Oracle Clusterware 19c on the cluster nodes *node1,node2* and *node3*, run the following command:

```
cluvfy comp sys -n node1,node2,node3 -p crs -verbose
```

clufy comp vdisk

Checks the voting files configuration and the `udev` settings for the voting files on all the specified nodes.



See Also:

Oracle Grid Infrastructure Installation and Upgrade Guide for Linux for more information about `udev` settings

Syntax

```
clufy comp vdisk [-n node_list] [-verbose]
```

Usage Notes

Optionally, you can specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.

You can also choose verbose output from CVU.

clufy stage [-pre | -post] acfscfg

Use the `clufy stage -pre acfscfg` command to verify your cluster nodes are set up correctly before configuring Oracle Automatic Storage Management Cluster File System (Oracle ACFS). Use the `clufy stage -post acfscfg` to check an existing cluster after you configure Oracle ACFS.

Syntax

```
clufy stage -pre acfscfg -n node_list [-asmdev asm_device_list] [-verbose]
```

```
clufy stage -post acfscfg -n node_list [-verbose]
```

Parameters

Table A-19 clufy stage [-pre | -post] acfscfg Command Parameters

Parameter	Description
<code>-n node_list</code>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, for both before and after configuring Oracle ACFS.
<code>-asmdev asm_device_list</code>	The list of devices you plan for Oracle ASM to use. If you do not specify this option, then CVU uses an internal operating system-dependent value; for example, <code>/dev/raw/*</code> on Linux systems.
<code>-verbose</code>	CVU prints detailed output.

cluvfy stage -post appcluster

Performs the appropriate post stage checks for Oracle Clusterware application cluster installation on all the nodes.

Syntax

```
cluvfy stage -post appcluster -n node_list [-method sudo -user user_name
[-location dir_path] | root] [-verbose]
```

Parameters

Table A-20 cluvfy stage -post appcluster Command Parameters

Parameter	Description
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which you want to run the verification. Specify <code>all</code> to run the verification on all nodes in the cluster.
-method sudo -user <i>user_name</i> [-location <i>dir_path</i>] root	Chose the privilege delegation method, either <code>sudo</code> or <code>root</code> , to be used for root user access. If you choose the <code>sudo</code> method, then you must provide a user name to access all the nodes with <code>root</code> privileges, and, optionally, the full file system path for the <code>sudo</code> executable.

cluvfy stage [-pre | -post] cfs

Use the `cluvfy stage -pre cfs` stage verification command to verify your cluster nodes are set up correctly before configuring OCFS2. Use the `cluvfy stage -post cfs` stage verification command to perform the appropriate checks on the specified nodes after configuring OCFS2.

See Also:

Oracle Grid Infrastructure Installation and Upgrade Guide for your platform for a list of supported shared storage types

Syntax

```
cluvfy stage -pre cfs -n node_list -s storageID_list [-verbose]
```

```
cluvfy stage -post cfs -n node_list -f file_system [-verbose]
```

Parameters

Table A-21 cluvfy stage [-pre | -post] cfs Command Parameters

Parameter	Description
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, for both before and after configuring OCFS2.
-s <i>storageID_list</i>	Specify a comma-delimited list of storage locations to check before configuring OCFS2.
-f <i>file_system</i>	Specify a file system to check after configuring OCFS2.
-verbose	CVU prints detailed output.

Example

To check that a shared device is configured correctly before setting up OCFS2, use a command similar to the following, where you replace `/dev/sdd5` with the name of the shared device on your system:

```
$ cluvfy stage -pre cfs -n node1,node2,node3,node4 -s /dev/sdd5
```

cluvfy stage [-pre | -post] crsinst

Use the `cluvfy stage -pre crsinst` command with either the `-file`, `-n`, `-flex`, or `-upgrade` parameters to check the specified nodes before installing or upgrading Oracle Clusterware. Use the `cluvfy stage -post crsinst` command to check the specified nodes after installing Oracle Clusterware.

Syntax

```
cluvfy stage -pre crsinst -file config_file [-fixup] [-fixupnoexec]
  [-method {sudo -user user_name [-location directory_path] | root}] [-verbose]
```

```
cluvfy stage -pre crsinst -n node_list | -flex -hub hub_list
  [-r {10.1 | 10.2 | 11.1 | 11.2 | 12.1 | 12.2 | 18.1 | 19.1}] [-c
ocr_location_list] [-q voting_disk_list]
  [-osdba osdba_group] [-orainv orainventory_group] [-asm [-presence
{local | flex}
| -asmcredentials client_data_file] [-asmgrp asmadmin_group] [-asmdev
asm_device_list]]
  [-crshome Grid_home] [-fixup] [-fixupnoexec] [-method {sudo -user
user_name
[-location directory_path] | root}]
  [-networks network_list] [-dhcp -clustername cluster_name [-dhcpport
dhcp_port]]
  [-verbose]
```

```
cluvfy stage -pre crsinst -upgrade [-rolling] [-src_crshome src_crshome] -
dest_crshome dest_crshome
  -dest_version dest_version [-fixup] [-fixupnoexec] [-method {sudo -user
```

```
user_name
  [-location directory_path | root}] [-verbose]
```

```
cluvfy stage -post crsinst -n node_list
  [-method {sudo -user user_name [-location directory_path | root}] [-
  verbose]
```

Parameters

Table A-22 cluvfy stage [-pre | -post] crsinst Command Parameters

Parameter	Description
-file <i>config_file</i>	Specify the root script configuration file containing Oracle installation variables.
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification.
-flex -hub <i>hub_list</i>	Alternative to the -n parameter, specify a comma-delimited list of Hub Node names on which to conduct checks.
-r {10.1 10.2 11.1 11.2 12.1 12.2 18.1 19.1}	Specifies the Oracle Clusterware release that CVU checks as required for installation of Oracle Clusterware. If you do not specify this option, then CVU assumes Oracle Clusterware 19c.
-c <i>ocr_location_list</i>	Specify a comma-delimited list of directory paths for OCR locations or files that CVU checks for availability to all nodes. If you do not specify this option, then the OCR locations are not checked.
-q <i>voting_disk_list</i>	Specify a comma-delimited list of directory paths for voting files that CVU checks for availability to all nodes. If you do not specify this option, then CVU does not check the voting file locations.
-osdba <i>osdba_group</i>	Specify the name of the OSDBA group. If you do not specify this option, then CVU uses dba as the OSDBA group.
-orainv <i>orainventory_group</i>	Specify the name of the Oracle Inventory group. If you do not specify this option, then CVU uses oinstall as the inventory group.
-asm [-presence {local flex} - asmcredentials <i>client_data_file</i>	This parameter indicates that Oracle ASM is used for storing the Oracle Clusterware files. Specify the Oracle ASM presence, either LOCAL or FLEX, on this Oracle Clusterware installation. Optionally, for an Oracle ASM client, specify the path to an Oracle ASM client credential file.
-asmgrp <i>asmadmin_group</i>	Specify the name of the OSASM group. If you do not specify this parameter, then CVU uses the same group as the Oracle Inventory group.
-asmdev <i>asm_device_list</i>	Specify a list of devices you plan for Oracle ASM to use that CVU checks for availability to all nodes. If you do not specify this parameter, then CVU uses an internal operating system-dependent value.
-crshome <i>Grid_home</i>	Specify the location of the Oracle Grid Infrastructure or Oracle Clusterware home directory. If you specify this parameter, then the supplied file system location is checked for sufficient free space for an Oracle Clusterware installation.

Table A-22 (Cont.) cluvfy stage [-pre | -post] crsinst Command Parameters

Parameter	Description
-networks <i>network_list</i>	Checks the network parameters of a slash ("/")-delimited list of networks in the form of " <i>if_name</i> " [: <i>subnet_id</i> [:public :cluster_interconnect]]. <ul style="list-style-type: none"> You can use the asterisk (*) wildcard character when you specify the network interface name (<i>if_name</i>), such as eth*, to match interfaces. Specify a subnet number for the network interface for the <i>subnet_id</i> variable and choose the type of network interface.
-dhcp -clustername <i>cluster_name</i> [- dhcpport <i>dhcp_port</i>]	Specify the name of the cluster. Optionally, you can specify the port to which the DHCP packets will be sent. The default value for this port is 67.
-upgrade	Specify this parameter to verify upgrade prerequisites.
-rolling	Specify this parameter to perform a rolling upgrade.
-src_crshome <i>src_crshome</i>	Specify the location of the source Grid home.
-dest_crshome <i>dest_crshome</i>	Specify the location of the destination Grid home.
-dest_version <i>dest_version</i>	Specify the version to which you are upgrading, including any patchset, such as 11.2.0.1.0 or 11.2.0.2.0.
-fixup	Specifies that if the verification fails, then CVU performs fixup operations, if feasible.
-fixupnoexec	Specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
-method {sudo -user <i>user_name</i> [-location <i>dir_path</i>] root}	Specify whether the privilege delegation method is sudo or root, for root user access. If you specify sudo, then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the sudo executable.
-verbose	CVU prints detailed output.

Usage Notes

- To perform checks for a new installation, specify either the -file or -n parameters, and use -upgrade for performing checks for upgrading to another version.
- CVU performs additional checks on OCR and voting files if you specify the -c and -q options with the -n parameter.

cluvfy stage -pre dbcfg

Checks the specified nodes before configuring an Oracle RAC database to verify whether your system meets all of the criteria for creating a database or for making a database configuration change.

Syntax

On Linux and UNIX platforms:

```
cluvfy stage -pre dbcfg -n node_list -d Oracle_home [-fixup] [-fixupnoexec]
  [-method {sudo -user user_name [-location directory_path] | root}]
  [-servicepwd] [-verbose]
```

On Windows platforms:

```
cluvfy stage -pre dbcfg -n node_list -d oracle_home [-fixup] [-fixupnoexec]
  [-verbose] [-servicepwd]
```

Parameters

Table A-23 cluvfy stage -pre dbcfg Command Parameters

Parameter	Description
-n <i>node_list</i>	The comma-delimited list of nondomain qualified node names on which to conduct the verification.
-d <i>Oracle_home</i>	The location of the Oracle home directory for the database that is being checked.
-fixup	Specifies that if the verification fails, then CVU performs fixup operations, if feasible.
-fixupnoexec	Specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
-method {sudo -user <i>user_name</i> [-location <i>directory_path</i>] root}	Specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
-servicepwd	If you specify this option, then CVU performs checks similar to those performed by the <code>cluvfy stage -pre dbinst</code> command when you specify the <code>-serviceuser</code> option. CVU determines the user name from the registry, then prompts you for the password for the service user, even if the wallet exists. CVU checks the password you enter against the password stored in the wallet. If the service password is not in the wallet or you did not specify the <code>-servicepwd</code> option, then CVU does not check the user name and password. Note: This parameter only applies to Windows.
-verbose	CVU prints detailed output.

cluvfy stage -pre dbinst

Checks the specified nodes before installing or creating an Oracle RAC database to verify that your system meets all of the criteria for installing or creating an Oracle RAC database.

Syntax

On Linux and UNIX platforms:

```
cluvfy stage -pre dbinst -n node_list [-r {10.1 | 10.2 | 11.1 | 11.2 |
12.1 | 12.2}]
  [-osdba osdba_group] [-osbackup osbackup_group] [-osdg osdg_group]
  [-oskm oskm_group] [-d oracle_home] [-fixup] [-fixupnoexec]
  [-method {sudo -user user_name [-location directory_path] | root}]
  [-verbose]
```

```
cluvfy stage pre dbinst -upgrade -src_dbhome src_dbhome [-dbname dbname-
list]
  -dest_dbhome dest_dbhome -dest_version dest_version
  [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location
directory_path]
```

On Windows platforms:

```
cluvfy stage -pre dbinst -n node_list [-r {10.1 | 10.2 | 11.1 | 11.2 |
12.1 | 12.2}]
  [-d Oracle_home] [-fixup] [-fixupnoexec] [-serviceuser user_name [-
servicepasswd]]]
  [-verbose]
```

```
cluvfy stage -pre dbinst -upgrade -src_dbhome src_dbhome [-dbname dbname-
list
  -dest_dbhome dest_dbhome -dest_version dest_version [-fixup] [-
fixupnoexec]
  [-verbose]]
```

Parameters

Table A-24 cluvfy stage -pre dbinst Command Parameters

Parameter	Description
-n <i>node_list</i>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification.
-r {10.1 10.2 11.1 11.2 12.1 12.2}	Specifies the Oracle Database release that CVU checks as required for installation of Oracle RAC. If you do not specify this option, then CVU assumes Oracle Database 12c.
-osdba <i>osdba_group</i>	The name of the OSDBA group. If you do not specify this option, then CVU uses dba as the OSDBA group.
-osbackup <i>osbackup_group</i>	Specify the name of the OSBACKUP group.
-osdg <i>osdg_group</i>	Specify the name of the OSDG group.
-oskm <i>oskm_group</i>	Specify the name of the OSKM group.

Table A-24 (Cont.) cluvfy stage -pre dbinst Command Parameters

Parameter	Description
-d <i>oracle_home</i>	The location of the Oracle home directory where you are installing Oracle RAC and creating the Oracle RAC database. If you specify this parameter, then the specified location is checked for sufficient free disk space for a database installation.
-fixup	Specifies that if the verification fails, then CVU performs fixup operations, if feasible.
-fixupnoexec	Specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
-upgrade	Specify this parameter to verify upgrade prerequisites.
-src_dbhome <i>src_dbhome</i>	Specify the location of the source database home from which you are upgrading.
-dbname <i>dbname-list</i>	Specify a comma-delimited list of unique names of the databases you want to upgrade.
-dest_dbhome <i>dest_dbhome</i>	Specify the location of the destination database home to which you are upgrading.
-dest_version <i>dest_version</i>	Specify the version to which you are upgrading, including any patchset, such as 11.2.0.1.0 or 11.2.0.2.0.
-method {sudo -user <i>user_name</i> [-location <i>directory_path</i>] root}	Specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
-serviceuser <i>user_name</i> [- <i>servicepasswd</i>]	<p>If you specify this option, then CVU checks the following:</p> <ul style="list-style-type: none"> Whether the specified user is a domain user. If the specified user is not a domain user, then CVU returns an error and does not perform any subsequent checks on the validation of the user. <p>Note: You must specify the user name of the Oracle home user.</p> <ul style="list-style-type: none"> Whether the specified user is an administrator on all nodes in the cluster. If the user is not an administrator on any node in the cluster, then the check passes. Otherwise, the check fails. If you do not specify the <code>-servicepwd</code> option, then CVU checks whether there is a password stored in the wallet on OCR for this user. If no password exists for the specified user, then CVU continues to run. If you specify the <code>-servicepwd</code> option, then CVU prompts you for the password of the specified user, even if the password exists in the wallet. <p>Note: The <code>-serviceuser</code> and <code>-servicepwd</code> parameters only apply to Windows.</p>
-verbose	CVU prints detailed output.

cluvfy stage [-pre | -post] hacfg

Checks the local node after configuring Oracle Restart.

Syntax

```
clufy stage -pre hacfg [-osdba osdba_group] [-osoper osoper_group] [-orainv orainventory_group]
  [-fixup] [-fixupnoexec] [-method {sudo -user user_name [-location directory_path] | root}]
  [-verbose]
```

```
clufy stage -post hacfg [-verbose]
```

Parameters

Table A-25 clufy stage [-pre | -post] hacfg Command Parameters

Parameter	Description
-osdba <i>osdba_group</i>	Specify the name of the OSDBA group. If you do not specify this option, then CVU uses dba as the OSDBA group.
-osoper <i>osoper_group</i>	Specify the name of the OSOPER group.
-orainv <i>orainventory_group</i>	Specify the name of the Oracle Inventory group. If you do not specify this option, then CVU uses oinstall as the inventory group.
-fixup	Specifies that if the verification fails, then CVU performs fixup operations, if feasible.
-fixupnoexec	Specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
-method {sudo -user <i>user_name</i> [-location <i>dir_path</i>] root}	Specify whether the privilege delegation method is sudo or root, for root user access. If you specify sudo, then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the sudo executable.
-verbose	CVU prints detailed output.

clufy stage -post hwos

Checks network and storage on the specified nodes in the cluster before installing Oracle software. This command also checks for supported storage types and checks each one for sharing.

Syntax

```
clufy stage -post hwos -n node_list [-s storageID_list] [-verbose]
```


Parameters

Table A-26 cluvfy stage -post hwas Command Parameters

Parameter	Description
<code>-n node_list</code>	The comma-delimited list of non domain-qualified node names on which to conduct the verification.
<code>-s storageID_list</code>	Checks the comma-delimited list of storage locations for sharing of supported storage types. If you do not specify the <code>-s</code> parameter, then CVU discovers supported storage types and checks sharing for each of them.
<code>-verbose</code>	CVU prints detailed output.

cluvfy stage [-pre | -post] nodeadd

Use the `cluvfy stage -pre nodeadd` command to verify the specified nodes are configured correctly before adding them to your existing cluster, and to verify the integrity of the cluster before you add the nodes. Use the `cluvfy stage -post nodeadd` command to verify that the specified nodes have been successfully added to the cluster at the network, shared storage, and clusterware levels.

The `cluvfy stage -pre nodeadd` command verifies that the system configuration, such as the operating system version, software patches, packages, and kernel parameters, for the nodes that you want to add, is compatible with the existing cluster nodes, and that the clusterware is successfully operating on the existing nodes. Run this command on any node of the existing cluster.

Syntax

```
cluvfy stage -pre nodeadd -n node_list [-vip vip_list] | -flex [-hub
hub_list
[-vip vip_list]] [-fixup] [-fixupnoexec]
[-method {sudo -user user_name [-location directory_path] | root}] [-
verbose]
```

```
cluvfy stage -post nodeadd -n node_list [-verbose]
```

Parameters

Table A-27 cluvfy stage [-pre | -post] nodeadd Command Parameters

Parameter	Description
<code>-n node_list</code>	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. These are the nodes you are adding or have added to the cluster.
<code>-vip vip_list</code>	A comma-delimited list of virtual IP addresses to be used by the new nodes.

Table A-27 (Cont.) cluvfy stage [-pre | -post] nodeadd Command Parameters

Parameter	Description
-flex [-hub <i>hub_list</i> [-vip <i>vip_list</i>]]	Specify <code>-flex</code> if you are adding a node to an Oracle Flex Cluster. Optionally, you can specify a comma-delimited list of non domain-qualified node names that you want to add to the cluster as Hub Nodes. Additionally, you can specify a comma-delimited list of virtual IP addresses that will be applied to the list of Hub Nodes you specify.
-fixup	This optional parameter specifies that if the verification fails, then CVU performs fixup operations, if feasible.
-fixupnoexec	This optional parameter specifies that if verification fails, then CVU generates the fixup data and displays the instructions for manual execution of the generated fixups.
-method {sudo -user <i>user_name</i> [-location <i>directory_path</i>] root}	Specify whether the privilege delegation method is <code>sudo</code> or <code>root</code> , for root user access. If you specify <code>sudo</code> , then you must specify the user name to access all the nodes with root privileges and, optionally, provide the full file system path for the <code>sudo</code> executable.
-verbose	CVU prints detailed output.

cluvfy stage -post nodedel

Verifies that specific nodes have been successfully deleted from a cluster. Typically, this command verifies that the node-specific interface configuration details have been removed, the nodes are no longer a part of cluster configuration, and proper Oracle ASM cleanup has been performed.

Syntax

```
cluvfy stage -post nodedel -n node_list [-verbose]
```

Usage Notes

- This command takes only a comma-delimited list of non domain-qualified node names on which to conduct the verification. If you do not specify this parameter, then CVU checks only the local node. You can also specify `-verbose` to print detailed output.
- If the `cluvfy stage -post nodedel` check fails, then repeat the node deletion procedure.

See Also:

[Adding and Deleting Cluster Nodes](#)

Troubleshooting and Diagnostic Output for CVU

This section describes the following troubleshooting topics for CVU:

- [Enabling Tracing](#)
- [Known Issues for the Cluster Verification Utility](#)

Enabling Tracing

CVU generates trace files unless you disable tracing. You can disable tracing by setting the `SRVM_TRACE` environment variable to `false` or `FALSE`. For example, in `tcsh` an entry such as `setenv SRVM_TRACE FALSE` disables tracing.

The CVU trace files are created in the `ORACLE_BASE/crsdata/hostname/cvu` directory by default. Oracle Database automatically rotates the log files and the most recently created log file has the name `cvutrace.log.0`. You should remove unwanted log files or archive them to reclaim disk place if needed.

Oracle Clusterware stores log files that CVU generates when it runs periodically in the `ORACLE_BASE/crsdata/hostname/cvu/cvutrc` directory.

To use a non-default location for the trace files, set the `CV_TRACELOC` environment variable to the absolute path of the desired trace directory.

Known Issues for the Cluster Verification Utility

This section describes the following known limitations for Cluster Verification Utility (CVU):

- [Database Versions Supported by Cluster Verification Utility](#)
- [Linux Shared Storage Accessibility \(ssa\) Check Reports Limitations](#)
- [Shared Disk Discovery on Red Hat Linux](#)

Database Versions Supported by Cluster Verification Utility

The current CVU release supports only Oracle Database 10g or higher, Oracle RAC, and Oracle Clusterware; CVU is not backward compatible. CVU cannot check or verify Oracle Database products for releases before Oracle Database 10g.

Linux Shared Storage Accessibility (ssa) Check Reports Limitations

The current release of `cluvfy` has the following limitations on Linux regarding shared storage accessibility check.

- OCFS2 (version 1.2.1 or higher) is supported.
- For sharedness checks on NAS, `cluvfy` commands require you to have write permission on the specified path. If the user running the `cluvfy` command does not have write permission, then `cluvfy` reports the path as not shared.

Shared Disk Discovery on Red Hat Linux

To perform discovery and shared storage accessibility checks for SCSI disks on Red Hat Linux 5.0 (or higher) and Oracle Linux 5.0 (or higher), and SUSE Linux Enterprise Server, CVU requires the `CVUQDISK` package. If you attempt to use CVU and the `CVUQDISK` package is not installed on all of the nodes in your Oracle RAC environment, then CVU responds with an error.

Perform the following procedure to install the CVUQDISK package:

1. Login as the `root` user.
2. Copy the package, `cvuqdisk-1.0.9-1.rpm` (or higher version) to a local directory. You can find this rpm in the `rpm` subdirectory of the top-most directory in the Oracle Clusterware installation media. For example, you can find `cvuqdisk-1.0.9-1.rpm` in the directory `/mountpoint/clusterware/rpm/` where `mountpoint` is the mount point for the disk on which the directory is located.

```
# cp /mount_point/clusterware/rpm/cvuqdisk-1.0.9-1.rpm /u01/oradba
```

3. Set the `CVUQDISK_GRP` environment variable to the operating system group that should own the CVUQDISK package binaries. If `CVUQDISK_GRP` is not set, then, by default, the `oinstall` group is the owner's group.

```
# set CVUQDISK_GRP=oinstall
```

4. Determine whether previous versions of the CVUQDISK package are installed by running the command `rpm -q cvuqdisk`. If you find previous versions of the CVUQDISK package, then remove them by running the command `rpm -e cvuqdisk previous_version` where `previous_version` is the identifier of the previous CVUQDISK version, as shown in the following example:

```
# rpm -q cvuqdisk
cvuqdisk-1.0.2-1
# rpm -e cvuqdisk-1.0.2-1
```

5. Install the latest CVUQDISK package by running the command `rpm -iv cvuqdisk-1.0.9-1.rpm`.

```
# cd /u01/oradba
# rpm -iv cvuqdisk-1.0.9-1.rpm
```

B

Oracle Clusterware Resource Reference

This appendix is a reference for Oracle Clusterware resources. This appendix includes descriptions and usage examples of resource attributes and detailed descriptions and examples of resource attribute action scripts. This appendix includes the following topics:

- [Resource Attributes](#)
- [Examples of Action Scripts for Third-party Applications](#)

Resource Attributes

This section lists and describes attributes used when you register applications as resources in Oracle Clusterware. Use these attributes with the `crsctl add resource` command, as follows:

```
$ crsctl add resource resource_name -type resource_type -group group_name  
{[-attr "attribute_name='attribute_value', attribute_name='attribute_value'  
, ..."] | [-file file_name]}
```

List attribute-value pairs in a comma-delimited list after the `-attr` flag and enclose the value of each attribute in single quotation marks (' '). Some resource attributes you cannot configure and are read only.

Alternatively, you can create a text file that contains the attribute-value pairs. For example:

```
PLACEMENT=favored  
HOSTING_MEMBERS=node1 node2 node3  
RESTART_ATTEMPTS@CARDINALITYID(1)=0  
RESTART_ATTEMPTS@CARDINALITYID(2)=0  
FAILURE_THRESHOLD@CARDINALITYID(1)=2  
FAILURE_THRESHOLD@CARDINALITYID(2)=4  
FAILURE_INTERVAL@CARDINALITYID(1)=300  
FAILURE_INTERVAL@CARDINALITYID(2)=500  
CHECK_INTERVAL=2  
CARDINALITY=2
```

Note:

The length limit for these attributes is 254 characters.

This section includes the following topics:

- [Configurable Resource Attributes](#)

- [Read-Only Resource Attributes](#)
- [Deprecated Resource Attributes](#)

Configurable Resource Attributes

This section describes the following resource attributes that you can configure when registering an application as a resource in Oracle Clusterware:

**Note:**

Values for all attributes must be in lowercase. Attribute names must be in all uppercase letters.

- ACL
- ACTION_SCRIPT
- ACTION_TIMEOUT
- ACTIONS
- ACTIVE_PLACEMENT
- AGENT_FILENAME
- ALERT_TEMPLATE
- AUTO_START
- CARDINALITY
- CARDINALITY_ID
- CHECK_INTERVAL
- CHECK_TIMEOUT
- CLEAN_TIMEOUT
- CRITICAL_RESOURCES
- DELETE_TIMEOUT
- DESCRIPTION
- ENABLED
- FAILURE_INTERVAL
- FAILURE_THRESHOLD
- HOSTING_MEMBERS
- INSTANCE_FAILOVER
- INTERMEDIATE_TIMEOUT
- LOAD
- MODIFY_TIMEOUT
- NAME
- OFFLINE_CHECK_INTERVAL

- ONLINE_RELOCATION_TIMEOUT
- PLACEMENT
- RELOCATE_BY_DEPENDENCY
- RELOCATE_KIND
- RESTART_ATTEMPTS
- SCRIPT_TIMEOUT
- SERVER_CATEGORY
- SERVER_POOLS
- START_CONCURRENCY
- START_DEPENDENCIES
- START_TIMEOUT
- STOP_CONCURRENCY
- STOP_DEPENDENCIES
- STOP_TIMEOUT
- UPTIME_THRESHOLD
- USER_WORKLOAD
- USE_STICKINESS

ACL

Defines the owner of a resource and the access privileges granted to various operating system users and groups. The resource owner defines the operating system user of the owner and its privileges. You configure this optional attribute when you create a resource. If you do not configure this attribute, then the value is based on the identity of the process creating the resource. You can change the value of the attribute if such a change is allowed based on the existing privileges of the resource.

 **Note:**

All operating system user names and user groups, including `owner`, `pgrp`, `user`, and `group`, *must* be registered on *all* servers in the cluster.

In the string:

- `owner`: The operating system user that owns a resource and the user under which the action script or application-specific agent runs, followed by the privileges of the owner.
- `pgrp`: The operating system group that is the primary group of the owner of a resource, followed by the privileges of members of the primary group.
- `other`: Operating system users that are neither the owner nor member of the primary group
- `r`: The read option, which gives the ability to only see a resource, its state, and configuration

- **w**: The write option, which gives the ability to modify a resource's attributes and to delete the resource
- **x**: The execute option, which gives the ability to start, stop, and relocate a resource

By default, the identity of the client that creates a resource is the `owner`. Also by default, `root`, and the user specified in `owner` have full privileges. You can grant required operating system users and operating system groups their privileges by adding the following lines to the `ACL` attribute:

```
user:user_name:rwx
group:group_name:rwx
```

Usage Example

```
ACL=owner:user_1:rwx,pgrp:osdba:rwx,other::r-
```

In the preceding example, the owner of the resource is `user_1`, whose primary group is `osdba`. The user, `user_1`, has all privileges, as does the `osdba` group, while other users can only view the resource.

ACTION_SCRIPT

An absolute file name that includes the path and file name of an action script. The agent specified in the `AGENT_FILENAME` attribute calls the script specified in the `ACTION_SCRIPT` attribute.

Usage Example

```
ACTION_SCRIPT=fully_qualified_path_to_action_script
```

ACTION_TIMEOUT

A single timeout value, in seconds, for all supported actions that Oracle Clusterware can perform on a resource.

Usage Example

```
ACTION_TIMEOUT=30
```

ACTIONS

The `ACTIONS` attribute declares a table of names that lists the actions that Oracle Clusterware can perform on a resource and the permissions that correspond to the actions. The `ACTIONS` attribute contains a space-delimited list of action specifications, where each specification has the following format, where:

- `actionName` is the name of the action (the maximum length is 32 US7ASCII alphanumeric, case-sensitive characters)
- `userName` is an operating system user name that is enabled to perform the action
- `groupName` is an operating system group name that is enabled to perform the action

```
actionName [,user:userName | group:groupName][ ...]
```


If you do not specify a *userName* or *groupName*, then Oracle Clusterware assumes that the actions are universally accessible.

Usage Example

The following example enables multiple actions:

```
ACTIONS='action1 action2,user:user2 action3,group:group1'
```

ACTIVE_PLACEMENT

When set to 1, Oracle Clusterware uses this attribute to reevaluate the placement of a resource during addition or restart of a cluster server. For resources where `PLACEMENT=favored`, Oracle Clusterware may relocate running resources if the resources run on a non-favored server when a favored one joins the cluster.

Usage Example

```
ACTIVE_PLACEMENT=1
```

AGENT_FILENAME

A fully qualified file name of an agent program that a resource type uses to manage its resources. Every resource type must have an agent program to manage its resources. Resource types use agent programs by either specifying a value for this attribute or inheriting it from their base resource type. There are two script agents included with Oracle Clusterware 12C: `application` and `scriptagent`. Oracle Clusterware uses the `application` script agent for resources of the deprecated `application` resource type. The default value for this attribute is `scriptagent`.

 **Note:**

Once the resource is created, you can no longer modify this attribute.

Usage Example

```
AGENT_FILENAME=%Grid_home%/bin/application
```

ALERT_TEMPLATE

Use to specify additional resource attributes that are to be included in resource state alert messages. You can specify the attribute as a space-delimited list of resource attributes. These attributes must be accessible from the resource type to display in alert messages.

Usage Example

```
ALERT_TEMPLATE="DESCRIPTION HOSTING_MEMBERS"
```

AUTO_START

Indicates whether Oracle Clusterware automatically starts a resource after a cluster server restart. Valid `AUTO_START` values are:

- `always`: Restarts the resource when the server restarts regardless of the state of the resource when the server stopped.
- `restore`: Restores the resource to the same state that it was in when the server stopped. Oracle Clusterware attempts to restart the resource if the value of `TARGET` was `ONLINE` before the server stopped.
- `never`: Oracle Clusterware never restarts the resource regardless of the state of the resource when the server stopped.

CARDINALITY

The number of servers on which a resource or resource group can run, simultaneously. This is the upper limit for resource cardinality.

Usage Example

```
CARDINALITY=1
```

You can also use a value such that cardinality always increases and decreases with the number of servers that are assigned to the server pool in which the resource or resource group is configured to run. The value is:

```
CARDINALITY=%CRS_SERVER_POOL_SIZE%
```

Only resources with `PLACEMENT=restricted` and that use the `SERVER_POOLS` attribute can use this value.

CARDINALITY_ID

An integer different for each of the `INSTANCE_COUNT` resource instances.

The value for this parameter varies from 1 to `CARDINALITY*2`. The number of resource instances can be greater than that of the value of `CARDINALITY` only when you use online relocation (`RELOCATE_KIND=online`), or if you decrease the resource instance count.

CHECK_INTERVAL

The time interval, in seconds, between repeated executions of the `check` action. Shorter intervals enable more frequent checks but also increase resource consumption if you use the script agent. Use an application-specific agent to reduce resource consumption.

Usage Example

```
CHECK_INTERVAL=60
```

CHECK_TIMEOUT

The maximum time, in seconds, in which a check action can run. Oracle Clusterware returns an error message if the action does not complete within the time specified. If you do not specify this attribute or if you specify 0 seconds, then Oracle Clusterware uses the value of the `SCRIPT_TIMEOUT` attribute.

Usage Example

```
CHECK_TIMEOUT=30
```

CLEAN_TIMEOUT

The maximum time, in seconds, in which a clean action can run. Oracle Clusterware returns an error message if the action does not complete within the time specified. If you do not specify a value for this attribute or you specify 0 seconds, then Oracle Clusterware uses the value of the [STOP_TIMEOUT](#) attribute.

Usage Example

```
CLEAN_TIMEOUT=30
```

CRITICAL_RESOURCES

Resource group attribute that contains a list of resources marked critical for a resource group.

You specify a space-delimited list of critical resources in the `CRITICAL_RESOURCES` attribute of a resource group. Use this attribute for both local and cluster resource group types. You can add a `type:` option followed by a resource type name to indicate that all member resources of the specified type are critical resources of the group.

Example B-1 Usage Examples

```
CRITICAL_RESOURCES="r1 r2 r3"  
CRITICAL_RESOURCES="appvip type:ora.export.type"
```

DELETE_TIMEOUT

The maximum time, in seconds, in which a delete action can run. Oracle Clusterware returns an error message if the action does not complete within the time specified. If you do not specify a value for this attribute or you specify 0 seconds, then Oracle Clusterware uses the value of the [SCRIPT_TIMEOUT](#) attribute.

Usage Example

```
DELETE_TIMEOUT=30
```

DESCRIPTION

Enter a description of the resource you are adding.

Usage Example

```
DESCRIPTION=Apache Web server
```

ENABLED

Oracle Clusterware uses this attribute to manage the state of the resource. Oracle Clusterware does not attempt to manage a disabled (`ENABLED=0`) resource either directly or because of a dependency to another resource. A disabled resource cannot

be started but it can be stopped. Oracle Clusterware does not actively monitor disabled resources, meaning that Oracle Clusterware does not check their state.

Usage Example

```
ENABLED=1
```

FAILURE_INTERVAL

The interval, in seconds, before which Oracle Clusterware stops a resource if the resource has exceeded the number of failures specified by the `FAILURE_THRESHOLD` attribute. If the value is zero (0), then tracking of failures is disabled.

Usage Example

```
FAILURE_INTERVAL=30
```

FAILURE_THRESHOLD

The number of failures of a resource detected within a specified `FAILURE_INTERVAL` for the resource before Oracle Clusterware marks the resource as unavailable and no longer monitors it. If a resource fails the specified number of times, then Oracle Clusterware stops the resource. If the value is zero (0), then tracking of failures is disabled. The maximum value is 20.

Usage Example

```
FAILURE_THRESHOLD=3
```

HOSTING_MEMBERS

A space-delimited, ordered list of cluster server names that can host a resource. This attribute is required only when using administrator management, and when the value of the `PLACEMENT` attribute is set to *avored* or *restricted*. When registering applications as Oracle Clusterware resources, use the `SERVER_POOLS` attribute, instead.

Note:

For resources of `application` type, Oracle Clusterware places servers listed in the `HOSTING_MEMBERS` attribute in the Generic server pool.

See Also:

- "[Default Server Pools](#)" for more information about the Generic server pool
- "[LOAD](#)" for more information about this attribute
- "[PLACEMENT](#)" for more information about this attribute

To obtain a list of candidate node names, run the `olsnodes` command to display a list of your server names.

Usage Example

```
HOSTING_MEMBERS=server1 server2 server3
```

INSTANCE_FAILOVER

Use the `INSTANCE_FAILOVER` attribute for resources of type `CLUSTER_RESOURCE`. Using this attribute enables you to disallow the failover of resource instances from the servers on which they fail. This enables you to bind the resource to a particular server.

Set to 0 to disable instance failover.

Usage Example

```
INSTANCE_FAILOVER=1
```

INTERMEDIATE_TIMEOUT

Denotes the maximum amount of time in seconds that a resource can remain in the `INTERMEDIATE` state before the resource is declared as failed. The value of `INTERMEDIATE_TIMEOUT` must be greater than 0 to take effect.

Usage Example

```
INTERMEDIATE_TIMEOUT=60
```

LOAD

Oracle Clusterware interprets the value of this attribute along with that of the `PLACEMENT` attribute. When the value of `PLACEMENT` is `balanced`, the value of `LOAD` determines where best to place a resource. A nonnegative, numeric value that quantitatively represents how much server capacity an instance of a resource consumes relative to other resources. Oracle Clusterware attempts to place resources on servers with the least total load of running resources.

Usage Example

```
LOAD=1
```

MODIFY_TIMEOUT

The maximum time, in seconds, in which a modify action can run. Oracle Clusterware returns an error message if the action does not complete within the time specified. If you do not specify a value for this attribute or you specify 0 seconds, then Oracle Clusterware uses the value of the `SCRIPT_TIMEOUT` attribute.

Usage Example

```
MODIFY_TIMEOUT=30
```

NAME

A case-sensitive alphanumeric string that names the resource. Oracle recommends a naming convention that starts with an alphanumeric prefix, such as `myApache`, and complete the name with an identifier to describe it. A resource name can contain any platform-supported characters except the exclamation point (!) and the tilde (~). A resource name cannot begin with a period (.) nor with the string *ora*.

Usage Example

```
NAME=myApache
```

OFFLINE_CHECK_INTERVAL

Controls offline monitoring of a resource. The value represents the interval (in seconds) that Oracle Clusterware monitors a resource when its state is `OFFLINE`. Monitoring is disabled if the value is 0.

Usage Example

```
OFFLINE_CHECK_INTERVAL=30
```

ONLINE_RELOCATION_TIMEOUT

(Optional) Describe the parameter here.

PLACEMENT

Specifies how Oracle Clusterware selects a cluster server on which to start a resource. Valid values are `balanced`, `favor`d, or `restricted`.

If you set the `PLACEMENT` attribute to `favor`ed or `restricted`, then you must also assign values to the `SERVER_POOLS` and `HOSTING_MEMBERS` attributes. If you set the value of the `PLACEMENT` attribute to `balanced`, then the `HOSTING_MEMBERS` attribute is not required.



See Also:

- "[Application Placement Policies](#)" for more information about the `PLACEMENT` attribute
- "[HOSTING_MEMBERS](#)" for more information about this attribute
- "[SERVER_POOLS](#)" for more information about this attribute

Usage Example

```
PLACEMENT=favored
```

RELOCATE_KIND

Use to control how a resource is relocated. During an offline relocation operation, the target resource instance is stopped on the source server and subsequently started on the destination server. An online relocation operation first starts the resource instance on the destination server before stopping it on the source server. Start-only starts the resource instance on the destination server, but will not stop the resource instance on the source server. Instead, the agent stops the resource instance once the relocate is complete.

You can choose either `offline/0`, `online/1`, or `start-only/2` as a value for this parameter.

Usage Examples

```
RELOCATE_KIND=1  
RELOCATE_KIND=start-only
```

RELOCATE_BY_DEPENDENCY

Use to declare whether a resource will be enabled for relocation if requested to do so because of a dependency on the resource for which the relocation was requested. If 0, the resource will not be allowed to relocate because of a dependency on the resource for which relocate request was issued. The valid values are 1 or 0.

Usage Example

```
RELOCATE_BY_DEPENDENCY=1
```

RESTART_ATTEMPTS

The number of times that Oracle Clusterware attempts to restart a resource on the resource's current server before attempting to relocate it. A value of 1 indicates that Oracle Clusterware only attempts to restart the resource once on a server. A second failure causes Oracle Clusterware to attempt to relocate the resource. A value of 0 indicates that there is no attempt to restart but Oracle Clusterware always tries to fail the resource over to another server.

Usage Example

```
RESTART_ATTEMPTS=2
```

SCRIPT_TIMEOUT

The maximum time (in seconds) for an action to run. Oracle Clusterware returns an error message if the action script does not complete within the time specified. The timeout applies to all actions (`start`, `stop`, `check`, and `clean`).

Usage Example

```
SCRIPT_TIMEOUT=45
```

SERVER_CATEGORY

For local resources, the definition of a `local_resource` type is extended to be category-aware. In other words, you can restrict local resources to belong to a particular server category. For cluster resources, the value for the `SERVER_CATEGORY` attribute always functions with the value for the `PLACEMENT` attribute. Set `SERVER_POOLS` to `*` when `PLACEMENT` is restricted and `SERVER_CATEGORY` is used. If you set `PLACEMENT` to `restricted`, then Oracle Clusterware expects one of the following attributes to also be set:

- `HOSTING_MEMBERS`
- `SERVER_POOLS`
- `SERVER_CATEGORY`

For example, a resource, known as `resource1`, can have a policy that sets the value of `PLACEMENT` to be `restricted`, and `SERVER_CATEGORY` is set to `HubCategory`. In such a case, Oracle Clusterware would only enable `resource1` to run on the servers that belong to the `HubCategory`.

If `PLACEMENT` is set to `avored` and if only one of `HOSTING_MEMBERS`, `SERVER_POOLS`, or `SERVER_CATEGORY` is set, then that value expresses a preference. If `HOSTING_MEMBERS` is populated and one of `SERVER_POOLS` or `SERVER_CATEGORY` is set, then the `HOSTING_MEMBERS` indicates placement preference and `SERVER_POOLS` or `SERVER_CATEGORY` indicates a restriction. For example, the `ora.cluster.vip` resource can have a policy that sets the value of `PLACEMENT` to `avored`, and `SERVER_CATEGORY` is set to `HubCategory` and `HOSTING_MEMBERS` is set to `server_name1`. In such a case, Oracle Clusterware restricts the placement of `ora.cluster.vip` to the servers in the `HubCategory` and then it prefers the server known as `server_name1`.

Usage Example

```
SERVER_CATEGORY=my_category
```

SERVER_POOLS

The `SERVER_POOLS` resource attribute is described.

A space-delimited list of the server pools to which a particular resource can belong. If a resource can run on any server in a cluster, then use the default value, `*`, unless the resource is a `cluster_resource` type, in which case, the default value for the `SERVER_POOLS` attribute is empty. Only cluster administrators can specify `*` as the value for this attribute.

- Use the `PLACEMENT` attribute with the `SERVER_POOLS` attribute, as follows: If you set the value of the `PLACEMENT` attribute to either `restricted` or `avored`, then you must also provide a value for the `SERVER_POOLS` attribute when using policy management for the resource.
- If the value for `PLACEMENT` is set to `balanced`, then the resource only runs in the Generic and Free pools, unless `SERVER_POOLS=*`.

This attribute creates an affinity between a resource and one or more server pools regarding placement, and depends on the value of the `PLACEMENT` attribute.

Usage Example

```
SERVER_POOLS=pool1 pool2 pool3
```

See Also:

- [Policy-Based Cluster and Capacity Management](#) for more information about server pools
- ["Role-Separated Management"](#) for more information about cluster administrators
- ["HOSTING_MEMBERS"](#) for more information about this attribute
- ["PLACEMENT"](#) for more information about this attribute

START_CONCURRENCY

Describes the maximum number of start actions that can be concurrent at a time. A value of 0 means "no limit."

Usage Example

```
START_CONCURRENCY=10
```

START_DEPENDENCIES

Specifies a set of relationships that Oracle Clusterware considers when starting a resource. You can specify a space-delimited list of dependencies on several resources and resource types on which a particular resource can depend.

Syntax

```
START_DEPENDENCIES=dependency(resource_set) [dependency(resource_set)] [...]
```

In the preceding syntax example the variables are defined, as follows:

- *dependency*: Possible values are attraction, dispersion, exclusion, hard, pullup, and weak. You can specify each dependency only once, except for pullup, which you can specify multiple times.
- *resource_set*: A comma-delimited list of resource entities—either individual resources or resource types—enclosed in parentheses (), in the form of *res1[, res2[, ...]]*, upon which the resource you are configuring depends.

Each resource entity is defined, as follows:

```
[modifier1:[modifier2:]] {resource_name | type:resource_type}
```

In the preceding syntax example, *resource_name* is the name of a specific resource and *type:resource_type* is the name of a specific resource type. The resource type must be preceded by *type* and the *type* modifier must be the last resource entity in the list.

Optionally, you can specify modifiers to further configure resource entity dependencies. You can modify each dependency by prefixing the following modifiers to the resource entity:

- `attraction([intermediate:]{resource_name | type:resource_type})`—Use the `attraction` start dependency when you want this resource to run on the same server with a particular named resource or any resource of a particular type.

Use `intermediate` to specify that this resource is attracted to resource entities on which it depends that are in the `INTERMEDIATE` state. If not specified, then resources must be in the `ONLINE` state to attract the dependent resource.

If you specify the `attraction` dependency on a resource type for a resource, then any resource of that particular type attracts the dependent resource.

- `exclusion([[preempt_pre: | preempt_post:]] target_resource_name | type:target_resource_type)`—Use the `exclusion` start dependency to keep resources with this dependency from running on the same node.

Use the `preempt_pre` modifier to configure the `exclusion` dependency to stop the specified target resource or resources defined by a specific resource type before starting the source resource.

Use the `preempt_post` modifier to configure the `exclusion` dependency to stop and relocate, if possible, the specified target resource or resources defined by a specific resource type after starting the source resource.

- `dispersion[:active]([intermediate:][pool:]{resource_name | type:resource_type})`—Specify the `dispersion` start dependency for a resource that you want to run on a server that is different from the named resources or resources of a particular type. Resources may still end up running on the same server, depending on availability of servers.

Use the `active` modifier to configure the `dispersion` dependency so that Oracle Clusterware attempts to relocate the dependent resource to another server if it is collocated with another resource and another server comes online. Oracle Clusterware does not relocate resources to newly available servers unless you specify the `active` modifier.

Use the `intermediate` modifier to specify that Oracle Clusterware can relocate the dependent resource if a resource is in either the `ONLINE` or `INTERMEDIATE` state. If not specified, then resources must be in the `ONLINE` state for dispersion of the dependent resource to occur.

Use the `pool` modifier if you want a resource to be located in a different server pool than the target, rather than just a different server.

- `hard([intermediate:][global:][uniform:]{resource_name | type:resource_type})`—Specify a `hard` start dependency for a resource when you want the resource to start only when a particular resource or resource of a particular type starts.

Use the `intermediate` modifier to specify that Oracle Clusterware can start this resource if a resource on which it depends is in either the `ONLINE` or `INTERMEDIATE` state. If not specified, then resources *must* be in the `ONLINE` state for Oracle Clusterware to start this resource.

Use the `global` modifier to specify that resources are *not* required to reside on the same server as a condition to Oracle Clusterware starting this resource. If

not specified, then resources must reside on the same server for Oracle Clusterware to start this resource.

Use the `uniform` modifier to attempt to start all instances of resource B, but only one instance, at least must start to satisfy the dependency.

If you specify the `hard` dependency on a resource type for a resource, then the resource can start if any resource of that particular type is running.

 **Note:**

Oracle recommends that resources with `hard` start dependencies also have `pullup` start dependencies.

- `pullup[:always]([intermediate:][global:]{resource_name | type:resource_type})`—When you specify the `pullup` start dependency for a resource, then this resource starts because of named resources starting.

Use the `always` modifier for `pullup` so that Oracle Clusterware starts this resource despite the value of its `TARGET` attribute, whether that value is `ONLINE` or `OFFLINE`. Otherwise, if you do not specify the `always` modifier, then Oracle Clusterware starts this resource only if the value of the `TARGET` attribute is `ONLINE` for this resource.

Use the `intermediate` modifier to specify that Oracle Clusterware can start this resource if a resource on which it depends is in either the `ONLINE` or `INTERMEDIATE` state. If not specified, then resources must be in the `ONLINE` state for Oracle Clusterware to start this resource.

Use the `global` modifier to specify that resources on which this resource depends are *not* required to reside on the same server as a condition to Oracle Clusterware starting this resource. If not specified, then resources on which this resource depends must reside on the same server for Oracle Clusterware to start this resource.

If you specify the `pullup` dependency on a resource type for a resource, then, when any resource of that particular type starts, Oracle Clusterware can start this resource.

 **Note:**

Oracle recommends that resources with `hard` start dependencies also have `pullup` start dependencies.

- `weak([concurrent:][global:][uniform:]{resource_name | type:resource_type})`—Specify a `weak` start dependency for a resource when you want that resource to start despite whether named resources are running, or not. An attempt to start this resource also attempts to start any resources on which this resource depends if they are not running.

Use the `concurrent` modifier to specify that Oracle Clusterware can start a dependent resource while a resource on which it depends is in the process of starting. If `concurrent` is not specified, then resources must complete startup before Oracle Clusterware can start the dependent resource.

Use the `global` modifier to specify that resources are *not* required to reside on the same server as a condition to Oracle Clusterware starting the dependent resource.

Use the `uniform` modifier to start all instances of the resource everywhere the resource can run. If you do not specify a modifier (the default), then the resource starts on the same server as the resource on which it depends.

If you specify the `weak` start dependency on a resource type for a resource, then the resource can start if any resource of that particular type is running.

**See Also:**

"[Start Dependencies](#)" for more details about start dependencies

START_TIMEOUT

The maximum time (in seconds) in which a start action can run. Oracle Clusterware returns an error message if the action does not complete within the time specified. If you do not specify a value for this attribute or you specify 0 seconds, then Oracle Clusterware uses the value of the [SCRIPT_TIMEOUT](#) attribute.

Usage Example

```
START_TIMEOUT=30
```

**See Also:**

"[SCRIPT_TIMEOUT](#)" for more information about this attribute

STOP_CONCURRENCY

Describes the maximum number of stop actions that can be concurrent at a time. A value of 0 means "no limit."

Usage Example

```
STOP_CONCURRENCY=10
```

STOP_DEPENDENCIES

Specifies a set of relationships that Oracle Clusterware considers when stopping a resource.

Syntax

```
STOP_DEPENDENCIES=dependency(resource_set) [dependency(resource_set)] ...
```

In the preceding syntax example the variables are defined, as follows:

- *dependency*: The only possible value is `hard`.

- *resource_set*: A comma-delimited list, in the form of *res1[, res2 [, ...]]*, of resource entities—either individual resources or resource types—upon which the resource you are configuring depends.

Each resource entity is defined, as follows:

```
[modifier1:[modifier2:][modifier3:]] resource_name | type:resource_type
```

In the preceding syntax example, *resource_name* is the name of a specific resource and *type:resource_type* is the name of a specific resource type. The resource type must be preceded by *type:*.

Optionally, you can specify modifiers to further configure resource entity dependencies. You can modify each dependency by prefixing the following modifiers to the resource entity:

```
hard([intermediate:][global:][shutdown:]){resource_name | type:resource_type}
```

—Specify a **hard** stop dependency for a resource that you want to stop when named resources or resources of a particular resource type stop.

Use *intermediate* to specify that the dependent resource can remain in an **ONLINE** state if a resource is in either the **ONLINE** or **INTERMEDIATE** state. If not specified, then Oracle Clusterware stops the dependent resource unless resources are in the **ONLINE** state.

Use *global* to specify that the dependent resource remains in an **ONLINE** state if a resource is in an **ONLINE** state on any node in the cluster. If not specified, then when resources residing on the same server go offline, Oracle Clusterware stops the dependent resource.

Use *shutdown* to apply this dependency when the Oracle Clusterware stack is shut down. This is a convenient way to affect the order of stopping resources when stopping the stack, without having any affect on planned or unplanned events on the individual resources. This dependency, when used with the *shutdown* modifier, does not go into effect if somebody stops the resource directly, but only when the stack is shut down.

See Also:

"[Stop Dependencies](#)" for more details about stop dependencies

STOP_TIMEOUT

The maximum time (in seconds) in which a *stop* or *clean* action can run. Oracle Clusterware returns an error message if the action does not complete within the time specified. If you do not specify this attribute or if you specify 0 seconds, then Oracle Clusterware uses the value of the [SCRIPT_TIMEOUT](#) attribute.

Usage Example

```
STOP_TIMEOUT=30
```

 **See Also:**

- "[SCRIPT_TIMEOUT](#)" for more information about this attribute
- "[START_TIMEOUT](#)" for more information about this attribute

UPTIME_THRESHOLD

The value for `UPTIME_THRESHOLD` represents the length of time that a resource must be up before Oracle Clusterware considers the resource to be stable. By setting a value for the `UPTIME_THRESHOLD` attribute, you can indicate the stability of a resource.

Enter values for this attribute as a number followed by a letter that represents seconds (s), minutes (m), hours (h), days (d), or weeks (w). For example, a value of 7h represents an uptime threshold of seven hours.

After the time period you specify for `UPTIME_THRESHOLD` elapses, Oracle Clusterware resets the value for `RESTART_COUNT` to 0 at the next resource state change event, such as stop, start, relocate, or failure. Oracle Clusterware can alert you when the value for `RESTART_COUNT` reaches the value that you set for `RESTART_ATTEMPTS`. The counter is effectively reset the next time the resource fails or restarts. The threshold represents the amount of time that restarts are to be counted and discarded. If the resource fails after the threshold, it will still restart.

 **Note:**

Oracle Clusterware writes an alert to the clusterware alert log file when the value for `RESTART_COUNT` reaches the value that you set for `RESTART_ATTEMPTS`.

 **See Also:**

- "[RESTART_ATTEMPTS](#)" for more information about this attribute
- "[RESTART_COUNT](#)" for more information about this attribute

USER_WORKLOAD

Use to indicate whether a resource is a workload generating resource for what-if analysis. Possible values are `yes` or `no`.

Usage Example

```
USER_WORKLOAD=yes
```

USE_STICKINESS

Use to indicate that a resource should run where it last ran, if possible, and to not permit load-balancing that would otherwise apply. If set to 1, Oracle Clusterware attempts to start the resource where it last ran. Enabling `USE_STICKINESS` also disables load-balancing. The default value is 0. Possible values are 0 and 1.

Usage Example

```
USE_STICKINESS=1
```

Read-Only Resource Attributes

You can view these attributes when you run the `crsctl status resource` command on a particular resource. Oracle Clusterware sets these attributes when you register resources.

- `ACTION_FAILURE_EVENT_TEMPLATE`
- `INSTANCE_COUNT`
- `INTERNAL_STATE`
- `LAST_SERVER`
- `LAST_STATE_CHANGE`
- `PROFILE_CHANGE_EVENT_TEMPLATE`
- `RESTART_COUNT`
- `STATE`
- `STATE_CHANGE_EVENT_TEMPLATE`
- `STATE_DETAILS`
- `TARGET`
- `TARGET_SERVER`
- `TYPE`

ACTION_FAILURE_EVENT_TEMPLATE

This is an internally-managed attribute for an `ora.*` resource. You cannot edit this attribute.

INSTANCE_COUNT

The `INSTANCE_COUNT` attribute is an internally managed attribute that contains the number of instances that the resource currently has.

INTERNAL_STATE

An internally managed, read-only attribute that describes what, if any, action the policy engine is currently executing on the resource. Possible values and their meanings are as follows:

- `STARTING`: The policy engine is currently starting the resource

- **STOPPING:** The policy engine is currently stopping the resource
- **CLEANING:** The policy engine is currently cleaning the resource
- **STABLE:** The policy engine is not currently executing any action on the resource

Note, however, that the resource can still be locked as part of some other command.

LAST_SERVER

For `cluster_resource`-type resources, this is an internally managed, read-only attribute that contains the name of the server on which the last `start` action for the resource succeeded.

For `local_resource`-type resources, this is the name of the server to which the resource instance is pinned.

LAST_STATE_CHANGE

An internally managed, read-only attribute that describes when the policy engine registers the current state of the resource. Note that this may either be the timestamp of when state of the resource changed or when the policy engine discovered the state, as occurs when CRSD restarts.

PROFILE_CHANGE_EVENT_TEMPLATE

This is an internally-managed attribute for an `ora.*` resource. You cannot edit this attribute.

RESOURCE_LIST

List of resources belonging to a resource group. Oracle Clusterware populates and updates this attribute when resources are added to the group.

Use this attribute with resource groups.

RESTART_COUNT

An internally-managed attribute used by the Oracle Clusterware daemon to count the number of attempts to restart a resource, starting from zero up to the value specified in the `RESTART_ATTEMPTS` attribute. You cannot edit this attribute.

STATE

An internally-managed attribute that reflects the current state of the resource as reported by Oracle Clusterware. The state of a resource can be one of the following:

- **ONLINE:** The resource is online and resource monitoring is enabled (see [CHECK_INTERVAL](#)).
- **OFFLINE:** The resource is offline and only offline resource monitoring is enabled, if configured (see [OFFLINE_CHECK_INTERVAL](#)).
- **INTERMEDIATE:** The resource is either partially online or was known to be online before and subsequent attempts to determine its state have failed; resource monitoring is enabled (see [CHECK_INTERVAL](#)).

- **UNKNOWN:** The resource is unmanageable and its current state is unknown; manual intervention is required to resume its operation. A resource in this state is not monitored.

STATE_CHANGE_EVENT_TEMPLATE

This is an internally-managed attribute for an `ora.*` resource. You cannot edit this attribute.

STATE_DETAILS

An internally managed, read-only attribute that contains details about the state of a resource.

The four resource states—`ONLINE`, `OFFLINE`, `UNKNOWN`, and `INTERMEDIATE`—may map to different resource-specific values, such as `mounted`, `unmounted`, and `open`. Resource agent developers can use the `STATE_DETAILS` attribute to provide a more detailed description of this mapping, resource to the resource state.

Providing details is optional. If details are not provided, then Oracle Clusterware uses only the four possible resource states. Additionally, if the agent cannot provide these details (as may also happen to the value of the resource state), then Oracle Clusterware sets the value of this attribute to provide minimal details about why the resource is in its current state.

TARGET

An internal, read-only attribute that describes the desired state of a resource. Using the `crsctl start resource_name` or `crsctl stop resource_name` commands, however, can affect the value of this attribute.

TARGET_SERVER

This is an internally-managed attribute that contains the name of the server where the resource is starting. This value is relevant when the resource is starting.

TYPE

The type of resource indicated when you create a resource. This attribute is required when creating a resource and cannot be changed after the resource is created.



See Also:

"[Oracle Clusterware Resource Types](#)" for details of resource types

Deprecated Resource Attributes

The following resource attributes are deprecated in Oracle Clusterware 12c:

DEGREE

The number of instances of a cluster resource that can run on a single server.

Examples of Action Scripts for Third-party Applications

This section includes examples of third-party applications using script agents.

[Example B-2](#) shows an action script that fails over the Apache Web server.

Example B-2 Apache Action Script

```
#!/bin/sh

HTTPDCONFLOCATION=/etc/httpd/conf/httpd.conf
WEBPAGECHECK=http://<MyVIP>:80/icons/apache_pb.gif

case $1 in
'start')
    /usr/sbin/apachectl -k start -f $HTTPDCONFLOCATION
    RET=$?
    ;;
sleep(10)
    ;;
'stop')
    /usr/sbin/apachectl -k stop
    RET=$?
    ;;
'clean')
    /usr/sbin/apachectl -k stop
    RET=$?
    ;;
'check')
    /usr/bin/wget -q --delete-after $WEBPAGECHECK
    RET=$?
    ;;
*)
    RET=0
    ;;
esac
# 0: success; 1 : error
if [ $RET -eq 0 ]; then
exit 0
else
exit 1
fi
```

[Example B-3](#) shows the **xclock** script, which is a simple action script using **xclock** available as a default binary on all Linux and UNIX platforms.

Example B-3 xclock Action Script

```
#!/bin/bash
# start/stop/check script for xclock example
# To test this change BIN_DIR to the directory where xclock is based
# and set the DISPLAY variable to a server within your network.

BIN_DIR=/usr/X11R6/bin
LOG_DIR=/tmp
```

```

BIN_NAME=xclock
DISPLAY=yourhost.domain.com:0.0
export DISPLAY
exit_code=0

if [ ! -d $BIN_DIR ]
then
    echo "start failed"
    exit 2
fi

PID1=`ps -ef | grep $BIN_NAME | grep -v grep | grep -v xclock_app | awk
'{ print $2 }'`
case $1 in
'start')
    if [ "$PID1" != "" ]
    then
        status_p1="running"
    else
        if [ -x $BIN_DIR/$BIN_NAME ]
        then
            umask 002
            ${BIN_DIR}/${BIN_NAME} & 2>${LOG_DIR}/${BIN_NAME}.log
            status_p1="started"
        else
            echo `basename $0`: $BIN_NAME: Executable not found"
            exit_code=1
        fi
    fi
    echo "$BIN_NAME: $status_p1"
    exit $exit_code
;;

'stop')
    if [ "${PID1}" != "" ]
    then
        kill -9 ${PID1} && echo "$BIN_NAME daemon killed"
    else
        echo "$BIN_NAME: no running Process!"
    fi
    exit $exit_code
;;

'check')
    if [ "$PID1" != "" ]
    then
        echo "running"
        exit 0
    else
        echo "not running"
        exit 1
    fi
;;*)
echo "Usage: "`basename $0`" {start|stop|check}"
;;

esac

```

Example B-4 shows an example of a shell script for an agent to monitor a file. When the agent is started, it creates the file (which is specified through an attribute) and when it is stopped, it deletes the file. The CHECK action consists of only checking

whether the file exists. The variables with the `_CRS_` prefix are attribute values that are provided to the script in its environment.

Example B-4 Action Script Example

```
#!/bin/sh
TOUCH=/bin/touch
RM=/bin/rm
PATH_NAME=/tmp/${_CRS_NAME}

#
# These messages go into the CRSD agent log file.
echo " ***** `date` ***** "
echo "Action script '${_CRS_ACTION_SCRIPT}' for resource[${_CRS_NAME}] called for
action $1"
#

case "$1" in
  'start')
    echo "START entry point has been called.."
    echo "Creating the file: $PATH_NAME"
    $TOUCH $PATH_NAME
    exit 0
    ;;

  'stop')
    echo "STOP entry point has been called.."
    echo "Deleting the file: $PATH_NAME"
    $RM $PATH_NAME
    exit 0
    ;;

  'check')
    echo "CHECK entry point has been called.."
    if [ -e $PATH_NAME ]; then
      echo "Check -- SUCCESS"
      exit 0
    else
      echo "Check -- FAILED"
      exit 1
    fi
    ;;

  'clean')
    echo "CLEAN entry point has been called.."
    echo "Deleting the file: $PATH_NAME"
    $RM -f $PATH_NAME
    exit 0
    ;;

esac
```

C

OLSNODES Command Reference

This appendix describes the syntax and command options for the `olsnodes` command.

This appendix contains the following topics:

- [Using OLSNODES](#)
 - [Overview](#)
 - [Operational Notes](#)
- [Summary of the OLSNODES Command](#)

Using OLSNODES

This section contains topics which relate to using the OLSNODES command.

- [Overview](#)
- [Operational Notes](#)

Overview

The `olsnodes` command provides the list of nodes and other information for all nodes participating in the cluster.

You can use this command to quickly check that your cluster is operational, and all nodes are registered as members of the cluster. This command also provides an easy method for obtaining the node numbers.

Operational Notes

Usage Information

This command is used by the Cluster Verification Utility (CLUVFY) to obtain a list of node names when the `-n all` option is used.

This command utility is located in the `$ORA_CRS_HOME/bin` directory. You can only use this command if the CRS daemon is started.

Privileges and Security

You can run this command as either the `root` user, the user that installed Oracle Clusterware, or the user that installed Oracle Database.

Summary of the OLSNODES Command

The `olsnodes` command does not use keywords, but accepts one or more options. The available options are described in [Table C-1](#).

Syntax

```
olsnodes [[-n] [-s] [-t] [node_name | -l [-p]] | [-c]] [-a] [-g] [-v]
```

If you issue the `olsnodes` command without any command parameters, the command returns a listing of the nodes in the cluster:

```
[root@node1]# olsnodes
node1
node2
node3
node4
```

Table C-1 OLSNODES Command Options

Options	Description
-n	Lists all nodes participating in the cluster and includes the assigned node numbers.
-s	Displays the status of the node: <i>active</i> or <i>inactive</i> .
-t	Displays node type: <i>pinned</i> or <i>unpinned</i> .
<i>node_name</i>	Displays information for a particular node.
-l [-p]	Lists the local node and includes the private interconnect for the local node. The <code>-p</code> option is only valid when you specify along with the <code>-l</code> option.
-c	Displays the name of the cluster.
-a	Displays only active nodes in the cluster with no duplicates.
-g	Logs cluster verification information with more details.
-v	Logs cluster verification information in verbose mode. Use in debug mode and only at the direction of My Oracle Support.

Examples

Example 1: List all nodes currently in the cluster

To list the nodes that are currently members of the cluster, use the command:

```
[root@node1]# olsnodes
node1
node2
node3
node4
```

Example 2: List the node names and node numbers for cluster members

To list the node name and the node number for each node in the cluster, use the command:

```
[root@node1]# olsnodes -n
node1 1
node2 2
```

```
node3 3  
node4 4
```

Example 3: Display node roles for cluster members

To list the node roles for each node in the cluster, use the command:

```
[root@node1]# olsnodes -a  
node1 Hub  
node2 Hub  
node3  
node4
```

D

Oracle Interface Configuration Tool (OIFCFG) Command Reference

The Oracle Interface Configuration Tool (OIFCFG) command-line interface helps you to define and administer network interfaces. You can use OIFCFG commands Oracle Clusterware environments to:

- Allocate and deallocate network interfaces to components
- Direct components to use specific network interfaces
- Retrieve component configuration information

This appendix includes the following topics:

- [Starting the OIFCFG Command-Line Interface](#)
- [Summary of the OIFCFG Usage](#)

Starting the OIFCFG Command-Line Interface

Before you invoke OIFCFG, ensure that you have started Oracle Clusterware on at least the local node and preferably on all nodes if you intend to include the `-global` option in the command syntax.

Note:

To change the global network interface, Oracle Clusterware must be running on all cluster nodes.

Run OIFCFG from the `Grid_home/bin/` directory as the user who installed the Oracle Clusterware software. For example:

```
$ ./oifcfg
```

Run the `oifcfg -help` command to display online help for OIFCFG.

```
$ ./oifcfg -help
```

Summary of the OIFCFG Usage

This section contains the following topics:

- [OIFCFG Command Format](#)
- [OIFCFG Commands](#)
- [OIFCFG Command Parameters](#)
- [OIFCFG Usage Notes](#)

- [OIFCFG Examples](#)

OIFCFG Command Format

```
oifcfg iflist [-p] [-n]
oifcfg setif {-node nodename | -global} {if_name/subnet:if_type[,if_type]}[,...]
oifcfg getif [-node nodename | -global] [ -if if_name[/subnet] [-type if_type]]
oifcfg delif {{-node nodename | -global} [if_name[/subnet]] [-force] | -force}
oifcfg [-help]
```

OIFCFG Commands

Use these commands to define and manage network interfaces.

You can enter any of the OIFCFG commands listed in [Table D-1](#).

Table D-1 OIFCFG Commands

Command	Description
<code>oifcfg iflist [-p] [-n]</code>	Shows the available interfaces that you can configure with <code>setif</code> . The <code>iflist</code> command queries the operating system to find which network interfaces are present on this node. You can specify two options with this command: <ul style="list-style-type: none"> • <code>-p</code>: Displays a heuristic assumption of the interface type (PRIVATE, PUBLIC, or UNKNOWN) • <code>-n</code>: Displays the netmask If the interface has an IPv6 network, then OIFCFG prints the IPv6 subnet on a separate line, similar to the following: <pre>eth1 fec0::80 PUBLIC 64 eth1 10.229.184.0 PUBLIC 255.255.248.0</pre>
<code>oifcfg setif</code>	Sets an interface type (public, cluster interconnect, or Oracle ASM) for an interface.
<code>oifcfg getif</code>	Displays the interfaces for which an interface type has been defined with the <code>setif</code> command, along with the type for that interface. <p>Note: The <code>-node</code> command parameter that you can use with this command returns no result if you not set the interface details using the <code>oifcfg setif -node</code> command.</p>
<code>oifcfg delif</code>	Deletes the stored network configuration for global or node-specific interfaces. You can use the <code>-force</code> option without specifying the <code>-node</code> or <code>-global</code> options to delete the stored network configurations from all nodes in the cluster.

OIFCFG Command Parameters

This section lists the parameters for the OIFCFG commands. Note that some parameters are optional, depending on which command you run.

- `-node node_name`: The name of the Oracle Clusterware node as listed in the output from the `olsnodes` command. [OLSNODES Command Reference](#) describes the `olsnodes` command.

- `-global`: A network interface can be stored either as a *global interface* (as reported by the `iflist` command) or as a *node-specific interface*:
 - An interface is stored as a global interface when all of the nodes of an Oracle Real Application Clusters (Oracle RAC) cluster have the same interface connected to the same subnet. The global interface (and configuring all nodes with the same network interface for each public subnet and the same network interface for each private subnet) is not only the recommended configuration, but it is also the default installation configuration.
 - An interface can be stored as a node-specific (local) interface.

 **Note:**

Oracle supports interface name wildcards for different interface names across nodes. Oracle does not support using `-node` to configure different specific interface names on different nodes.

- `-if if_name`: The name by which the interface is configured in the system.
- `subnet`: The subnet address of the interface.
- `-type if_type`: One or more comma-delimited interface types: `public`, `cluster_interconnect`, or `asm`.
- `-help`: Display online help for OIFCFG commands.

Related Topics

- [OLSNODES Command Reference](#)

OIFCFG Usage Notes

- A network interface specification takes the following form:

`if_name/subnet:if_type`

The specification uniquely identifies the network interface using the:

- Interface name

When using the `oifcfg setif` command, you can specify interface names that contain wildcard characters, such as an asterisk (*), which matches any string. However, you must surround interface names that contain wildcards with double quotation marks (" ").

 **Note:**

Oracle recommends that you *do not* use wildcards in a cluster with pre-Oracle Database 11g release 2 (11.2.0.2) databases because Oracle resolves the interface name by expanding the wildcard on the local node and also issues a PRIF-0029 warning every time you run the `oifcfg getif` command.

If you use wildcards with Oracle Databases that are pre-11g release 2 (11.2.0.2), then those databases must use the `CLUSTER_INTERCONNECTS` parameter, instead.

- Associated subnet
- Interface type

The interface type indicates the purpose for which the network is configured. The supported interface types are:

- * `public`: An interface that can be used for communication with components external to Oracle RAC instances, such as Oracle Net and Virtual Internet Protocol (VIP) addresses.
- * `cluster_interconnect`: A private interface used for the cluster interconnect to provide interinstance or Cache Fusion¹ communication.
- * `asm`: An interface that can be used for communication with Oracle ASM.

If you set the interface type to `cluster_interconnect`, then it affects instances as they start and changes do not take effect until you restart the instances.

For example, the following specification identifies `qfe0` as a cluster interconnect located at the address 204.152.65.0:

```
qfe0/204.152.65.0:cluster_interconnect
```

- The Oracle Universal Installer uses OIFCFG to identify and display available interfaces.
- The effect of changing the interface names depends on which name you are changing, and whether you are also changing the IP address. In cases where you change only the interface names, the ramifications are minor. If you change the name for the public interface that is stored in the Oracle Cluster Registry (OCR), you must modify the nodeapps for each node. Therefore, you must stop the nodeapps for this change to take effect.
- You must restart Oracle Clusterware on all members of the cluster when you make global changes. For local changes, you need only to perform a node restart. Interconnect changes for the database occur at instance startup. However, the interconnect for Oracle Clusterware might be different.
- Because interconnects are chosen when instances start, just issuing OIFCFG commands does not have an immediate effect on the running system. Instead, changes take effect after restarting the component that might be affected by the command.

¹ Cache Fusion is a diskless cache coherency mechanism that provides copies of blocks directly from a holding instance's memory cache to a requesting instance's memory cache.

- Changes you make to cluster-wide network classifications with OIFCFG require all known cluster nodes to be up so that their Grid Plug and Play profiles can be updated.

OIFCFG Examples

The following examples show some common uses for the OIFCFG commands:

Listing the Names of Network Interfaces

You can use OIFCFG to list the interface names and the subnets of all of the interfaces available on the local node by executing the `iflist` keyword, as shown in this example:

```
oifcfg iflist
eth0      172.19.141.0
eth1      172.21.65.0
```

Retrieving Network Information

You can also retrieve specific OIFCFG information with a `getif` command.

For example, after you install Oracle Clusterware, you can verify that the public and cluster interconnect have been set to the desired values by entering the following command:

```
$ oifcfg getif
```

This command returns values for `global public` and `global cluster_interconnect`, similar to the following:

```
eth0 172.19.141.0 global public
eth1 172.21.65.0 global cluster_interconnect
```

Storing a Global Interface

To store an interface, use the `setif` keyword. For example, to store the interface `eth0`, with the subnet `172.19.141.0`, as a global interface (to be used as an interconnect for all of the Oracle RAC instances in your cluster and Oracle Clusterware), use the command:

```
oifcfg setif -global eth0/172.19.141.0:cluster_interconnect
```

Note:

Ensure that all nodes are running when you run the `setif` command because Oracle cannot update Grid Plug and Play profiles on nodes that are not running.

Deleting the Stored Interface

Use the `oifcfg delif` command to delete the stored configuration for global or node-specific interfaces. A specific node-specific or global interface can be deleted by supplying the interface name, with an optional subnet, on the command line.



Note:

If you only enter `oifcfg delif`, with no other arguments given on the command line, then OIFCFG deletes all interfaces on all nodes in the cluster.

For example, the following command deletes the global interface named `eth1` for the subnet `172.21.65.0`:

```
oifcfg delif -global eth1/172.21.65.0
```

The following command deletes all of the global interfaces assigned with OIFCFG:

```
oifcfg delif -global
```

E

Oracle Clusterware Control (CRSCTL) Utility Reference

This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.

Note:

Do not use CRSCTL commands on Oracle entities (such as resources, resource types, and server pools) that have names beginning with *ora* unless you are directed to do so by My Oracle Support. The Server Control utility (SRVCTL) is the correct utility to use on Oracle entities.

This appendix includes the following topics:

- [CRSCTL Overview](#)
- [Dual Environment CRSCTL Commands](#)
- [Oracle RAC Environment CRSCTL Commands](#)
- [Oracle Restart Environment CRSCTL Commands](#)
- [Troubleshooting and Diagnostic Output](#)

CRSCTL Overview

CRSCTL is an interface between you and Oracle Clusterware, parsing and calling Oracle Clusterware APIs for Oracle Clusterware objects.

CRSCTL provides cluster-aware commands with which you can perform check, start, and stop operations on the cluster. You can run these commands from any node in the cluster on another node in the cluster, or on all nodes in the cluster, depending on the operation.

You can use CRSCTL commands to perform several operations on Oracle Clusterware, such as:

- Starting and stopping Oracle Clusterware resources
- Enabling and disabling Oracle Clusterware daemons
- Checking the health of the cluster
- Managing resources that represent third-party applications
- Integrating Intelligent Platform Management Interface (IPMI) with Oracle Clusterware to provide failure isolation support and to ensure cluster integrity
- Debugging Oracle Clusterware components

This section includes the following topics:

- [Clusterized \(Cluster Aware\) Commands](#)
- [CRSCTL Operational Notes](#)
- [Deprecated Subprograms or Commands](#)

Clusterized (Cluster Aware) Commands

You can run clusterized commands on one node to perform operations on another node in the cluster. These are referred to as remote operations. This simplifies administration because, for example, you no longer have to log in to each node to check the status of the Oracle Clusterware on all of your nodes.

Clusterized commands are completely operating system independent; they rely on the OHASD (Oracle High Availability Services daemon). If this daemon is running, then you can perform remote operations, such as the starting, stopping, and checking the status of remote nodes.

Clusterized commands include the following:

- [crsctl check cluster](#)
- [crsctl start cluster](#)
- [crsctl stop cluster](#)

CRSCTL Operational Notes

Information about using the CRSCTL command-line interface.

Usage Information

- The CRSCTL utility is located in the *Grid_home/bin* directory. To run CRSCTL commands, type in `crsctl` at the operating system prompt followed by the command and arguments, as shown in the following example:

```
crsctl stop crs
```

- There are three categories of CRSCTL commands:
 - Those that you use in either the Oracle Real Application Clusters (Oracle RAC) environment or in the Oracle Restart environment
 - Those that you use in the Oracle RAC environment, only
 - Those that you use in the Oracle Restart environment, only
- Many CRSCTL commands use the `-f` parameter to force the command to run and ignore any checks.

For example, if you specify the force parameter for the `crsctl stop resource` command on a resource that is running and has dependent resources that are also running, then the force parameter omits the error message and instead stops or relocates all the dependent resources before stopping the resource you reference in the command.

- *Do not* use versions of CRSCTL earlier than 12c release 1 (12.1) to manage Oracle Clusterware 12c.

Filters

You can use filters to narrow down Oracle Clusterware entities upon which a CRSCTL command operates, as follows:

- Simple filters are attribute-value pairs with an operator.
- Operators must be surrounded by spaces, as shown in the examples.
- You can combine simple filters into expressions called expression filters using Boolean operators.

Supported filter operators are:

```
=
>
<
!=
co: Contains
st: Starts with
en: Ends with
```

Supported Boolean operators are AND and OR.

Examples of filters are:

- `TYPE = type1`
- `((TYPE = type1) AND (CHECK_INTERVAL > 50))`
- `(TYPE = type1) AND ((CHECK_INTERVAL > 30) OR (AUTO_START co never))`
- `NAME en network.res`
- `TYPE st ora.db`

Using CRSCTL Help

To print the help information for CRSCTL, use the following command:

```
crsctl -help
```

If you want help for a specific command, such as `start`, then enter the command and append `-help` to the end, as shown in the following example:

```
crsctl start -help
```

You can also use the abbreviations `-h` or `-?` (this parameter functions in Linux, UNIX, and Windows environments) instead of `-help`.

Deprecated Subprograms or Commands

[Table E-1](#) lists deprecated commands and their replacements that you can use to perform the same or similar functionality.

Table E-1 Deprecated CRSCTL Commands and Replacements

Deprecated Command	Replacement Commands
crs_stat	crsctl check cluster crsctl status resource
crs_register	crsctl add resource crsctl add type crsctl modify resource crsctl modify type
crs_unregister	crsctl stop resource crsctl delete resource
crs_start	crsctl start resource crsctl start crs crsctl start cluster
crs_stop	crsctl stop resource crsctl stop crs crsctl stop cluster
crs_getperm	crsctl getperm resource crsctl getperm type
crs_profile	crsctl add resource crsctl add type crsctl status resource crsctl status type crsctl modify resource crsctl modify type
crs_relocate	crsctl relocate resource
crs_setperm	crsctl setperm resource crsctl setperm type

Table E-1 (Cont.) Deprecated CRSCTL Commands and Replacements

Deprecated Command	Replacement Commands
<code>crsctl add crs administrator</code>	Use the access control list (ACL) to control who can add server pools.
<code>crsctl check crsd</code>	<code>crsctl check crs</code>
<code>crsctl check cssd</code>	<code>crsctl check css</code>
<code>crsctl check evmd</code>	<code>crsctl check evm</code>
<code>crsctl debug res log resource_name:level</code>	<code>crsctl set log</code>
<code>crsctl set css votedisk</code>	<code>crsctl add css votedisk</code> <code>crsctl delete css votedisk</code> <code>crsctl query css votedisk</code> <code>crsctl replace votedisk</code>
<code>crsctl start resources</code>	<code>crsctl start resource -all</code>
<code>crsctl stop resources</code>	<code>crsctl stop resource -all</code>
<code>crsctl set css logfilesize</code>	<code>crsctl get tracefileopts css</code> <code>crsctl set tracefileopts css</code>
<code>crsctl get css logfilesize</code>	<code>crsctl set tracefileopts css</code>

Table E-1 (Cont.) Deprecated CRSCTL Commands and Replacements

Deprecated Command	Replacement Commands
<pre>crsctl set trace module_name "component:debugging_lev el [,component:debugging_lev el][,...]"</pre>	<pre>crsctl get trace</pre>

Dual Environment CRSCTL Commands

Use these commands in either the Oracle RAC or the Oracle Restart environments.

The commands listed in this section manage the Oracle Clusterware resources in both Oracle RAC and Oracle Restart environments.

crsctl check css

Checks the status of Cluster Synchronization Services. This command is most often used when Oracle Automatic Storage Management (Oracle ASM) is installed on the local server.

Syntax

```
crsctl check css
```

Example

This command returns output similar to the following:

```
CRS-4529: Cluster Synchronization Services is online
```

crsctl check evm

Checks the status of the Event Manager.

Syntax

```
crsctl check evm
```

Example

This command returns output similar to the following:

```
CRS-4533: Event Manager is online
```

crsctl get hostname

Retrieves the host name of the local server.

Syntax

```
crsctl get hostname
```

Example

Oracle Clusterware returns the host name of the local server:

```
$ crsctl get hostname
```

```
node2
```

crsctl add resource

Registers a resource to be managed by Oracle Clusterware.

A resource can be an application process, a database, a service, a listener, and so on.

Syntax

```
crsctl add resource resource_name -type resource_type [-group group_name [-file file_path |  
-attr "attribute_name=attribute_value[,...]"]  
[-i] [-f]
```

Parameters

Table E-2 crsctl add resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify a short, descriptive name for the resource.
-type <i>resource_type</i>	Specify the type of resource that you are adding preceded by the -type flag.
-group <i>group_name</i>	Specify a resource group to which you want to add the resource.
-file <i>file_path</i>	Path name (either absolute or relative) for a text file containing line-delimited attribute name-value pairs that define the resource.

Table E-2 (Cont.) crsctl add resource Command Parameters

Parameter	Description
-attr "attribute_name=attribute_value[,...]"	<p>You can specify attributes for a resource you are adding in two different ways:</p> <ul style="list-style-type: none"> Following the -attr flag, you can specify one or more comma-delimited attribute name-value pairs enclosed in double quotations marks (" "). For example: <pre>-attr "CHECK_INTERVAL=30,START_TIMEOUT=25"</pre> <p>Some attributes can have multiple values. In those cases, separate the values with a space and enclose the list of values in single quotation marks. For example: <pre>-attr "SERVER_POOL_NAMES='ora.pool1 ora.pool2',START_TIMEOUT=25"</pre> </p> Additionally, you can specify attribute values for resource instances with a particular cardinality value, and with a particular degree value. This method can be useful for applications that are tied to a particular server. Following the -attr flag, the syntax is as follows: <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> <p>If you specify the @SERVERNAME(<i>server_name</i>) syntax, then the attribute value you specify for the attribute you specify is limited to resource instances residing on the server you specify. Alternatively, if you specify the @CARDINALITYID(<i>cid</i>) syntax, then the attribute value you specify for the attribute you specify is limited to resource instances with a specific cardinality ID (<i>cid</i>). Optionally, you can combine the @DEGREEID(<i>did</i>) syntax with either the SERVERNAME or CARDINALITYID syntax, or both, to limit the attribute value to resources with the specific DEGREE. Examples: <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre> </p>
-i	<p>If you specify -i, then CRSTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.</p>

Table E-2 (Cont.) crsctl add resource Command Parameters

Parameter	Description
-f	<p>Use the force parameter:</p> <ul style="list-style-type: none"> To add a resource that has dependencies on other resources that do not yet exist. The force parameter overrides checks that would prevent a command from being completed. To add a resource if the resource has hard dependencies on other resources and the owner of the resources does not execute permissions on one or more of the dependencies. If you do not specify the force parameter in this case, an error displays. To add resources of application type because you may need to move servers into the Generic server pool. If the servers currently host resources that must be stopped, then the force parameter is required.

Usage Notes

- Both the *resource_name* and *-type resource_type* parameters are required. You can create an associated resource type using the [crsctl add type](#) command.
- Any user can create a resource but only clusterware administrators can create resources of type *local_resource* or resources of type *cluster_resource* that have *SERVER_POOLS=**.

Once a resource is defined, its ACL controls who can perform particular operations with it. The Oracle Clusterware administrator list is no longer relevant.

On Windows, a member of the Administrators group has full control over everything.



See Also:

"[crsctl setperm resource](#)" for more information about setting ACLs

- If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' ').
- Following is an example of an attribute file:

```
PLACEMENT=favored
HOSTING_MEMBERS=node1 node2 node3
RESTART_ATTEMPTS@CARDINALITYID(1)=0
RESTART_ATTEMPTS@CARDINALITYID(2)=0
FAILURE_THRESHOLD@CARDINALITYID(1)=2
FAILURE_THRESHOLD@CARDINALITYID(2)=4
FAILURE_INTERVAL@CARDINALITYID(1)=300
FAILURE_INTERVAL@CARDINALITYID(2)=500
CHECK_INTERVAL=2
CARDINALITY=2
```

- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources.

Examples

Example E-1 Example 1

To register a VIP as a resource with Oracle Clusterware:

```
$ crsctl add resource app.appvip -type app.appvip.type -attr
"RESTART_ATTEMPTS=2,
START_TIMEOUT=100,STOP_TIMEOUT=100,CHECK_INTERVAL=10,
USR_ORA_VIP=172.16.0.0,
START_DEPENDENCIES=hard(ora.net1.network)pullup(ora.net1.network),
STOP_DEPENDENCIES=hard(ora.net1.network)"
```

Example E-2 Example 2

To register a resource based on the `test_type1` resource type:

```
$ crsctl add resource r1 -type test_type1 -attr "PATH_NAME=/tmp/r1.txt"
$ crsctl add resource r1 -type test_type1 -attr "PATH_NAME=/tmp/r2.txt"
```

Example E-3 Example 3

To register a Samba server resource of the `generic_application` resource type, using the `EXECUTABLE_NAMES` attribute:

```
# crsctl add resource my_samba -type generic_application -attr
"EXECUTABLE_NAMES=smbd,START_PROGRAM='/etc/rc.d/init.d/smb start',
STOP_PROGRAM='/etc/rc.d/init.d/smb stop'"
```

Example E-4 Example 4

To register a DNS server of the `generic_application` resource type, using the `EXECUTABLE_NAMES` attribute:

```
# crsctl add resource my_dns -type generic_application -attr
"EXECUTABLE_NAMES=named,START_PROGRAM='/etc/rc.d/init.d/named start',
STOP_PROGRAM='/etc/rc.d/init.d/named stop'"
```

Example E-5 Example 5

To register an Apache web server of the `generic_application` resource type using the `PID_FILES` attribute:

```
# crsctl add resource my_apache -type generic_application -attr
"START_PROGRAM='/usr/sbin/httpd -k start',STOP_PROGRAM='/usr/sbin/httpd -k
stop',
PID_FILES=/etc/httpd/run/httpd.pid"
```

Example E-6 Example 6

To register an application of `generic_application` resource type using environment variables:

```
# crsctl add resource my_app -type generic_application -attr
"START_PROGRAM='/opt/my_app start', EXECUTABLE_NAMES=my_app,
ENVIRONMENT_VARS='USE_NETAPP=no,USE_BACKUP=yes,CLEAN_ON_KILL=yes'"
```

Related Topics

- [Oracle Clusterware Resource Reference](#)

crsctl delete resource

Removes resources from the Oracle Clusterware configuration.

Syntax

```
crsctl delete resource {"resource_name [...]"} [-w "filter"] [-i] [-f]
```

Parameters**Table E-3 crsctl delete resource Command Parameters**

Parameter	Description
<i>resource_name</i>	Specify one or more resource names that you want to delete. Multiple resources must be space-delimited and enclosed in double quotation marks (" ").
-w " <i>filter</i> "	Specify a resource filter that Oracle Clusterware uses to limit the number of resources evaluated for deletion. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> • "TYPE == cluster_resource": This filter limits Oracle Clusterware to relocate only resources of <code>cluster_resource</code> type • "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute • "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <i>and</i> the name of the resource contains the number 2 <p>Note: All operators must be surrounded by spaces. See Also: Filters for more information about operators</p>
-i	If you specify <code>-i</code> , then CRSCTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-f	Use this parameter to delete a resource that is online.

Usage Notes

- Either the *resource_name* or *-w "filter"* parameter is required
- You must have read and write permissions to delete the specified resources
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources

Example

To delete a resource from Oracle Clusterware:

```
# crsctl delete resource myResource
```

crsctl eval add resource

Predicts the effects of adding a resource without making changes to the system. This command may be useful to application administrators.

Syntax

```
crsctl eval add resource resource_name -type resource_type [-file file_path
| -attr "attribute_name=attribute_value[,...]"] [-group group_name] [-f]
```

Parameters

Table E-4 crsctl eval add resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify a short, descriptive name for the resource.
<i>-type resource_type</i>	Specify the type of resource that you are adding following the <i>-type</i> flag.
<i>-file file_path</i>	Path name (either absolute or relative) for a text file containing line-delimited attribute name-value pairs that define the resource.

Table E-4 (Cont.) crsctl eval add resource Command Parameters

Parameter	Description
-attr "attribute_name=attribute_value"	<p>You can specify attributes for a resource you are adding in two different ways:</p> <ul style="list-style-type: none"> Following the -attr flag, you can specify one or more comma-delimited attribute name-value pairs enclosed in double quotations marks (" "). For example: -attr "CHECK_INTERVAL=30,START_TIMEOUT=25" Some attributes can have multiple values. In those cases, separate the values with a space and enclose the list of values in single quotation marks. For example: -attr "SERVER_POOL_NAMES='ora.pool1 ora.pool2',START_TIMEOUT=25" Optionally, you can specify attribute values for resource instances with a particular cardinality value, and with a particular degree value. This method can be useful for applications that are tied to a particular server. Following the -attr flag, the syntax is as follows: <code>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</code> If you specify the @SERVERNAME(server_name) syntax, then the attribute value you specify for the attribute you specify is limited to resource instances residing on the server you specify. Alternatively, if you specify the @CARDINALITYID(cid) syntax, then the attribute value you specify for the attribute you specify is limited to resource instances with a specific cardinality ID (cid). Optionally, you can combine the @DEGREEID(did) syntax with either the SERVERNAME or CARDINALITYID syntax, or both, to limit the attribute value to resources with the specific DEGREE. Examples: CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65
-group group_name	Specify a resource group to which you want to add the resource.

Table E-4 (Cont.) crsctl eval add resource Command Parameters

Parameter	Description
-f	Use the force parameter: <ul style="list-style-type: none"> To add a resource that has dependencies on other resources that do not yet exist. The force parameter overrides checks that would prevent a command from being completed. To add a resource if the resource has hard dependencies on other resources and the owner of the resources does not execute permissions on one or more of the dependencies. If you do not specify the force parameter in this case, an error displays. To add resources of <code>application</code> type because you may need to move servers into the Generic server pool. If the servers currently host resources that must be stopped, then the force parameter is required.

crsctl eval fail resource

Predicts the consequences of a resource failing.

Syntax

```
crsctl eval fail resource {resource_name | -w "filter"} [-n server]
[-env "attr=val[,...]" ]
```

Parameters

Table E-5 crsctl eval fail resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify the name of a resource for which you want to simulate a failure. You must specify either the name of a resource or resource filters using the <code>-w</code> parameter.
<code>-w "filter"</code>	Alternatively, you can specify a resource filter that Oracle Clusterware uses to limit the number of resources evaluated. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> <code>"TYPE == cluster_resource"</code>: This filter limits Oracle Clusterware to relocate only resources of <code>cluster_resource</code> type <code>"CHECK_INTERVAL > 10"</code>: This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <code>"(CHECK_INTERVAL > 10) AND (NAME co 2)"</code>: This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <i>and</i> the name of the resource contains the number 2 <p>Note: All operators must be surrounded by spaces. See Also: "Filters" for more information about operators</p>

Table E-5 (Cont.) crsctl eval fail resource Command Parameters

Parameter	Description
-n <i>server</i>	Specify the name of the server on which the resource that you want to simulate a failure resides.
-env "attr=val[,...]"	You can optionally override one or more resource profile attribute values with the -env command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks ("").

Example

This command returns output similar to the following:

```
$ crsctl eval fail res cs1

Stage Group 1:
-----
--
Stage Number   Required      Action
-----
--
           1           Y   Resource 'cs1' (1/1) will be in state
                                [ONLINE|INTERMEDIATE] on server
                                [mjkeenan_node_0]
           Y   Resource 'cs1' (2/1) will be in state
                                [ONLINE|INTERMEDIATE] on server
                                [mjkeenan_node_1]
-----
--
```

crsctl eval relocate resource

Simulates relocating a resource without making changes to the system.

Syntax

```
crsctl eval relocate resource {resource_name | -all} {-s source_server | -
w "filter"}
[-n destination_server] [-f]
```

Parameters

Table E-6 crsctl eval relocate resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify the name of a resource you want to relocate.

Table E-6 (Cont.) crsctl eval relocate resource Command Parameters

Parameter	Description
-all	Alternatively, you can use this parameter to specify all resources for relocation.
-s <i>source_server</i>	Specify a particular server <i>from</i> which you want to relocate the resources.
-w " <i>filter</i> "	<p>Alternative to using <code>-s</code>, you can specify a resource filter that CRSCTL uses to limit the number of resources relocated. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include:</p> <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to relocate only resources of <code>cluster_resource</code> type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <i>and</i> the name of the resource contains the number 2 <p>See Also: "Filters" for more information</p>
-n <i>destination_server</i>	Specify the name of the server <i>to</i> which you want to relocate resources. If you do not specify a destination server, then CRSCTL relocates the resources to the best server according to the attribute profile of each resource.
-f	<p>Specify the <code>-f</code> parameter to force the relocation of the resource when it has other resources running that depend on it. CRSCTL stops or relocates dependent resources when you use this parameter.</p> <p>Note: When you are relocating resources that have cardinality greater than 1, you must use either <code>-k</code> or <code>-s</code> to narrow down which resource instances are to be relocated.</p>

crsctl eval modify resource

Predicts the effects of modifying a resource without making changes to the system.

Syntax

```
crsctl eval modify resource resource_name -attr
"attribute_name=attribute_value[,...]"
[-group group_name] [-f]
```

Parameters

Table E-7 crsctl eval modify resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify the name of the resource you want to evaluate.
-attr "attribute_name=attribute_value[,...]"	<p>You can specify attributes for a resource you want to evaluate in two different ways:</p> <ul style="list-style-type: none"> Following the -attr flag, you can specify one or more comma-delimited attribute name-value pairs to modify enclosed in double quotations marks (" "). For example: <pre>-attr "CHECK_INTERVAL=30, START_TIMEOUT=25"</pre> <p>Some attributes can have multiple values. In those cases, separate the values with a space and enclose the list of values in single quotation marks. For example: <pre>-attr "SERVER_POOL_NAMES='ora.pool1 ora.pool2',START_TIMEOUT=25"</pre> </p> Alternatively, you can specify attribute values for resources on a particular server, with a particular cardinality value, and with a particular degree value. This method can be useful for applications that are somehow tied to a particular server. Following the -attr flag, the syntax is as follows: <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> <p>If you specify the @SERVERNAME(<i>server_name</i>) syntax, then the attribute value you specify for the attribute you specify is limited to resources residing on the server you specify. Alternatively, if you specify the @CARDINALITYID(<i>cid</i>) syntax, then the attribute value you specify for the attribute you specify is limited to resource instances with a specific cardinality ID (<i>cid</i>). Optionally, you can combine the @DEGREEID(<i>did</i>) syntax with either the SERVERNAME or CARDINALITYID syntax, or both, to limit the attribute value to resources with the specific DEGREE. Examples: <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre> </p>
-group <i>group_name</i>	Specify a resource group to which the resource belongs that you want to evaluate.
-f	Specify this parameter to evaluate what happens if you run the command with the force parameter.

crsctl eval start resource

Predicts the effects of starting a resource without making changes to the system.

Syntax

```
crsctl eval start resource {resource_name [...] | -w "filter"}
    [-n server_name] [-f]
```

Parameters

Table E-8 crsctl eval start resource Command Parameters

Parameter	Description
<i>resource_name</i> [...]	Specify one or more space-delimited resource names to start.
-w " <i>filter</i> "	Specify a resource filter surrounded by double quotation marks (" ") that Oracle Clusterware uses to match resources. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: " Filters " for more information
-n <i>server_name</i>	Specify the name of the server on which the resources you want to start reside. If you do not specify a server, then Oracle Clusterware starts the resources on the best server according to the attribute profile of each resource.
-f	Use the -f parameter to relocate a resource running on another server on which the resource you want to start has a hard start dependency. If you do not specify the force parameter in this case, then the start command fails.

crsctl eval stop resource

Predicts the effect of stopping a resource without making changes to the system.

Syntax

```
crsctl eval stop resource {resource_name [...] | -w "filter"} [-f]
```

Parameters

Table E-9 crsctl eval stop resource Command Parameters

Parameter	Description
<i>resource_name</i> [...]	Specify one or more space-delimited resource names to stop.

Table E-9 (Cont.) crsctl eval stop resource Command Parameters

Parameter	Description
-w " <i>filter</i> "	Specify a resource filter that Oracle Clusterware uses to limit the number of resources stopped. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to stop only resources of cluster_resource type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to stop resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to stop resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute and the name of the resource contains the number 2 See Also: Filters for more information
-f	Specify the -f parameter to force the stopping of the resource when it has other resources running that depend on it. Dependent resources are relocated or stopped when you use this parameter.

crsctl getperm resource

Displays the user and group permissions for the specified resource.

Syntax

```
crsctl getperm resource resource_name [{-u user_name | -g group_name}]
```

Parameters

Table E-10 crsctl getperm resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify the name of the resource for which you want to obtain permissions.
-u <i>user_name</i>	If you specify -u, then Oracle Clusterware obtains permissions for a particular user.
-g <i>group_name</i>	If you specify -g, then Oracle Clusterware obtains permissions for a particular group.

Usage Notes

- The *resource_name* parameter is required
- You must have read permission on the specified resources to obtain their permissions
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources

**See Also:**

[Oracle Clusterware Resource Reference](#) for more information about resources and resource attributes

Examples

The `crsctl getperm resource` command returns output similar to the following, depending on the command option you choose:

```
$ crsctl getperm resource app.appvip
```

```
Name: app.appvip
owner:root:rwX,pgrp:oinstall:rwX,other::r--
```

```
$ crsctl getperm resource app.appvip -u oracle
```

```
Name: app.appvip
rwX
```

```
$ crsctl getperm resource app.appvip -g dba
```

```
Name: app.appvip
r--
```

crsctl modify resource

Modifies the attributes of a particular resource in Oracle Clusterware.

Syntax

```
crsctl modify resource resource_name -attr "attribute_name=attribute_value"
[-i] [-f] [-delete] [-group group_name] [-env "env1=val1,env2=val2,..."]
```

Parameters**Table E-11 crsctl modify resource Command Parameters**

Parameter	Description
<i>resource_name</i>	The name of the resource you want to modify.

Table E-11 (Cont.) crsctl modify resource Command Parameters

Parameter	Description
-attr "attribute_name= attribute_value"	<p>You can specify attributes for a resource you want to modify in two different ways:</p> <ul style="list-style-type: none"> Following the <code>-attr</code> flag, you can specify one or more comma-delimited attribute name-value pairs to modify enclosed in double quotations marks (" "). For example: <pre>-attr "CHECK_INTERVAL=30, START_TIMEOUT=25"</pre> <p>Some attributes can have multiple values. In those cases, separate the values with a space and enclose the list of values in single quotation marks. For example: <pre>-attr "SERVER_POOL_NAMES= 'ora.pool1 ora.pool2', START_TIMEOUT=25"</pre> </p> Alternatively, you can specify attribute values for resources on a particular server, with a particular cardinality value, and with a particular degree value. This method can be useful for applications that are somehow tied to a particular server. Following the <code>-attr</code> flag, the syntax is as follows: <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> <p>If you specify the <code>@SERVERNAME(server_name)</code> syntax, then the attribute value you specify for the attribute you specify is limited to resources residing on the server you specify.</p> <p>Alternatively, if you specify the <code>@CARDINALITYID(cid)</code> syntax, then the attribute value you specify for the attribute you specify is limited to resource instances with a specific cardinality ID (<code>cid</code>).</p> <p>Optionally, you can combine the <code>@DEGREEID(did)</code> syntax with either the <code>SERVERNAME</code> or <code>CARDINALITYID</code> syntax, or both, to limit the attribute value to resources with the specific <code>DEGREE</code>.</p> <p>Examples:</p> <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre>
-i	<p>If you specify <code>-i</code>, then CRSTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.</p>

Table E-11 (Cont.) crsctl modify resource Command Parameters

Parameter	Description
-f	Use the -f parameter when: <ul style="list-style-type: none"> The resource has a hard dependency on a non-existing resource The owner of the resource does not have execute permissions on one or more hard dependencies The modification results in servers being moved into the Generic pool and resources being stopped or relocated to accomplish the server move
-delete	If you specify the -delete parameter, then Oracle Clusterware deletes the named attribute.
-group <i>group_name</i>	Specify a resource group to which the resource belongs. The resource group must exist, and the resource cannot be a member of any other explicit resource group. If you do not specify a resource group, then CRSDCTL adds the resource to a resource group with the same name as the resource.
-env "env1=val1,env2=val2 ,..."	You can optionally override one or more resource profile attribute values for this command. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks ("").

Usage Notes

- The *resource_name* parameter is required
- If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' '). For example:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```

- You must have read and write permissions on the specified resources to modify them
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources

Example

To modify the attributes of the *appsvip* resource:

```
$ crsctl modify resource appsvip -attr USR_ORA_VIP=10.1.220.17 -i
```

Related Topics

- [Oracle Clusterware Resource Reference](#)

crsctl relocate resource

Relocates resources to another server in the cluster.

Syntax

```
crsctl relocate resource {resource_name [-k cid] | -all | -w "filter"} -s
source_server |
[-n destination_server] [-env "env1=val1,env2=val2,..."] [-i] [-f]
```

Parameters

Table E-12 crsctl relocate resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify the name of a resource you want to relocate.
-k <i>cid</i>	Optionally, you can specify the resource cardinality ID. If you specify this parameter, then CRSCTL relocates the resource instance that has the cardinality you specify.
-all	Alternatively, you can use this parameter to specify all resources for relocation.
-w " <i>filter</i> "	Alternative to using -k or -all, you can specify a resource filter that CRSCTL uses to limit the number of resources relocated. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to relocate only resources of <i>cluster_resource</i> type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute <i>and</i> the name of the resource contains the number 2 See Also: " Filters " for more information
-s <i>source_server</i>	Specify a particular server <i>from</i> which you want to relocate the resources.
-n <i>destination_server</i>	Specify the name of the server <i>to</i> which you want relocate resources. If you do not specify a destination server, then CRSCTL relocates the resources to the best server according to the attribute profile of each resource.
-env " <i>env1=val1,env2=val2</i> <i>,...</i> "	You can optionally override one or more resource profile attribute values for this command. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

Table E-12 (Cont.) crsctl relocate resource Command Parameters

Parameter	Description
-f	Specify the -f parameter to force the relocation of the resource when it has other resources running that depend on it. CRSCTL stops or relocates dependent resources when you use this parameter. Note: When you are relocating resources that have cardinality greater than 1, you must use either -k or -s to narrow down which resource instances are to be relocated.

Usage Notes

- Any one of the three following options is required to specify which resources you want to relocate:
 - You can specify one particular resource to relocate.
 - Or you can specify one particular or all the resources to relocate from a particular source server.
 - Thirdly, you can specify a resource filter that CRSCTL uses to match resources to relocate.
- If a resource has a degree ID greater than 1, then CRSCTL relocates all instances of the resource.
- You must have read and execute permissions on the specified resources to relocate them
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources.

Examples

To relocate one particular resource from one server to another:

```
# crsctl relocate resource myResource1 -s node1 -n node3
```

crsctl restart resource

Restarts idle resources in the cluster, instead of having to run two commands to stop and start the resource.

Syntax

```
crsctl restart resource {resource_name [...] | -w "filter"} [-k cid] [-d did]
    [-env "env1=val1,env2=val2,..."] [-i] [-f]
```

Parameters

Table E-13 crsctl restart resource Command Parameters

Parameter	Description
<i>resource_name</i> [...]	Specify one or more space-delimited names of resources you want to restart.
-w " <i>filter</i> "	Specify a resource filter surrounded by double quotation marks (" ") that Oracle Clusterware uses to match resources. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: "Filters" for more information
-k <i>cid</i>	Specify the resource cardinality ID. If you specify this parameter, then Oracle Clusterware restarts the resource instances that have the cardinality you specify.
-d <i>did</i>	Specify the resource degree ID. If you specify this parameter and the degree ID is greater than 1, then Oracle Clusterware restarts all resource instances that meet this criteria. Note: You cannot use the -d parameter without specifying the -k parameter.
-env " <i>env1=val1,env2=val2</i> ..."	You can optionally override one or more resource profile attribute values with the -env command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-f	Use the -f parameter to relocate a resource running on another server on which the resource you want to restart has a hard start dependency. If you do not specify the force parameter in this case, then the start command fails.

Usage Notes

- Either of the following options is required to specify which resources you want to restart:
 - You can specify one or more resources by name to restart
 - You can specify a resource filter that Oracle Clusterware uses to match resources to restart
- You must have read and execute permissions on the specified resources to restart them
- Do not use this command to restart any resources with names that begin with *ora* because these resources are Oracle resources

crsctl setperm resource

Sets permissions for a particular resource.

Syntax

```
crsctl setperm resource resource_name {-u acl_string | -x acl_string |
-o user_name | -g group_name}
```

Parameters

Table E-14 crsctl setperm resource Command Parameters

Parameter	Description
<i>resource_name</i>	Specify the name of the resource for which you want to set permissions.
-u -x -o -g	<p>You can set only one of the following permissions for a resource:</p> <ul style="list-style-type: none"> -u <i>acl_string</i>: You can update the access control list (ACL) for a resource -x <i>acl_string</i>: You can delete the ACL for a resource -o <i>user_name</i>: You can change the owner of a resource by entering a user name -g <i>group_name</i>: You can change the primary group of a resource by entering a group name <p>Specify a user, group, or other ACL string, as follows:</p> <pre>user:user_name[:readPermwritePermexecPerm] group:group_name[:readPermwritePermexecPerm] other[:readPermwritePermexecPerm]</pre> <ul style="list-style-type: none"> user: User ACL group: Group ACL other: Other ACL readPerm: Read permission for the resource; the letter r grants a user, group, or other read permission, the minus sign (-) denies read permission writePerm: Write permission for the resource; the letter w grants a user, group, or other write permission, the minus sign (-) denies write permission execPerm: Execute permission for the resource; the letter x grants a user, group, or other execute permission, the minus sign (-) denies execute permission



See Also:

[Oracle Clusterware Resource Reference](#) for more information about resources and resource attributes

Usage Notes

- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources.

- You must have read and write permissions on the specified resources to set their permissions

Examples

To grant read, write, and execute permissions on a resource for user Scott:

```
$ crsctl setperm resource myResource -u user:scott:rwX
```

crsctl start resource

Starts idle resources on a particular server in the cluster.

Syntax

```
crsctl start resource {resource_name [...] | -w "filter" | -all}
    [-n server_name | -s server_pool_names] [-k cid] [-d did]
    [-env "env1=val1,env2=val2,..."] [-begin] [-end] [-i] [-f] [-l]
```

Parameters

Table E-15 crsctl start resource Command Parameters

Parameter	Description
<i>resource_name</i> [...]	Specify one or more space-delimited names of resources you want to start.
-w " <i>filter</i> "	Specify a resource filter surrounded by double quotation marks (" ") that Oracle Clusterware uses to match resources. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: " Filters " for more information
-all	Use this parameter to start all resources in the cluster.
-n <i>server_name</i>	Specify the name of the server on which the resources you want to start reside. If you do not specify a server, then Oracle Clusterware starts the resources on the best server according to the attribute profile of each resource.
-s <i>server_pool_names</i>	Specify a single server pool name or a space-delimited list of server pools in which a resource resides that you want to start.
-k <i>cid</i>	Specify the resource cardinality ID. If you specify this parameter, then Oracle Clusterware starts the resource instances that have the cardinality you specify.
-d <i>did</i>	Specify the resource degree ID. If you specify this parameter and the degree ID is greater than 1, then Oracle Clusterware starts all resource instances that meet this criteria. Note: You cannot use the -d parameter without specifying the -k parameter.
-env "env1=val1,env2=val2,..."	You can optionally override one or more resource profile attribute values with the -env command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").

Table E-15 (Cont.) crsctl start resource Command Parameters

Parameter	Description
-begin	You can specify this parameter to begin a transparent HA action.
-end	You can specify this parameter to end a transparent HA action.
-i	If you specify <code>-i</code> , then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-f	Use the <code>-f</code> parameter to relocate a resource running on another server on which the resource you want to start has a hard start dependency. If you do not specify the force parameter in this case, then the start command fails.
-l	Use the <code>-l</code> parameter to leave the resources in the state they were in if the start command fails.

Usage Notes

- Any one of the three following options is required to specify which resources you want to start:
 - You can specify one or more resources to start
 - You can specify a resource filter that Oracle Clusterware uses to match resources to start
 - You can specify the `-all` parameter to start all resources on the specified server
- You must have read and execute permissions on the specified resources to start them
- Do not use this command to start any resources with names that begin with `ora` because these resources are Oracle resources
- Oracle does not support starting managed applications outside of the Oracle Grid Infrastructure

Example

To start a resource:

```
# crsctl start resource myResource -n server1
```

crsctl status resource

Obtains the status and configuration information of any number of particular resources.

Syntax

```
crsctl status resource resource_name [...] | -w "filter" [-p | -v] | [-f | -l
```

```
| -g] [[-k cid | -n server_name] [ -e [-p | -v]] [-d did]] | [-s -k cid  
[-d did]] [-p [-attr attribute_name[,...]] [-noname]]
```

To print the status of the resources in tabular form:

```
crsctl status resource resource_name [...] | -w "filter" -t
```

To print a list of the resource dependencies:

```
crsctl status resource [resource_name [...]] -dependency [-stop | -pullup]
```

Parameters

Table E-16 crsctl status resource Command Parameters

Parameter	Description
<code>resource_name [...]</code> <code> -w "filter"</code>	<p>Specify one or more space-delimited names of resources of which you want to check the status.</p> <p>Optionally, you can specify a resource filter that CRSTL uses to limit the number of resources displayed. The filter must be enclosed in double quotation marks (" "). Values that contain parentheses or spaces must be enclosed in single quotation marks (' '). Operators must be surrounded by spaces. Examples of resource filters include:</p> <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits the display to only resources of cluster_resource type. "CHECK_INTERVAL > 10": This filter limits the display to resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits the display to resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute <i>and</i> the name of the resource contains the number 2. "START_DEPENDENCIES='hard(appsvip)'": This filter limits the display to resources that have a hard start dependency on the appsvip resource. <p>See Also: "Filters" for more information</p>
<code>[-p -v] [-f -l]</code> <code> -g]</code>	<p>You can optionally specify the following parameters:</p> <ul style="list-style-type: none"> Specify either the <code>-p</code> parameter to display the static configuration of the resource or the <code>-v</code> parameter to display the run-time configuration of the resource. Specify the <code>-f</code> parameter to display the full configuration of the resource; or specify the <code>-l</code> parameter to display all cardinal and degree values of the resource; or specify the <code>-g</code> parameter to check whether the specified resources are registered

Table E-16 (Cont.) crsctl status resource Command Parameters

Parameter	Description
[[<i>-k cid</i> <i>-n server_name</i>] [<i>-e [p -v]</i>] [<i>-d did</i> [<i>-s -k cid</i> [<i>-d did</i>]]]	<p>You can specify one of the following two options:</p> <ul style="list-style-type: none"> Specify the <i>-k cid</i> parameter to specify a cardinality ID of the resources you want to query. Or you can specify the <i>-n</i> parameter to specify a particular server on which to check resources. Optionally, you can specify the <i>-d</i> parameter with the <i>-n</i> parameter to specify the degree ID of resources you want to check. If you specify a degree ID greater than 1, then Oracle Clusterware checks all resource instances on the server that meet this criteria. Specify the <i>-s</i> parameter with the <i>-k</i> parameter to obtain a list of target servers for relocation. You can further limit the output by specifying a degree ID with the <i>-d</i> parameter. <p>Use the <i>-e</i> parameter to evaluate the special values of a resource instance. You must also specify <i>-p</i> or <i>-v</i> with the <i>-e</i> parameter.</p>
<i>-t</i>	Specify the <i>-t</i> parameter to display the output in tabular form.
<i>-p</i> [<i>-attr attribute_name[,...]</i> [<i>-noname</i>]	Specify <i>-attr</i> (followed by a single attribute name or a comma-delimited list of attribute names) or <i>-noname</i> following <i>-p</i> to either print specific attributes or bypass the default printing of the resource name, respectively.
<i>-dependency</i> [<i>-stop</i> <i>-pullup</i>]	<p>Specify the <i>-dependency</i> parameter to display resource dependencies. If you do not specify either the <i>-stop</i> or <i>-pullup</i> option, then CRSCTL displays the start dependencies of the resource.</p> <p>Use either of the following options with the <i>-dependency</i> parameter:</p> <ul style="list-style-type: none"> Specify the <i>-stop</i> parameter to display resource stop dependencies. Specify the <i>-pullup</i> parameter to display resource pull up dependencies.

Usage Notes

- Either a space-delimited list of resources or a resource filter is required.
- You must have read permissions on the specified resources to obtain their status.
- Use `crsctl status resource` to query the status information of any resource deployed in the cluster. Oracle recommends, however, that you use the respective SRCVTL command to query the status information of Oracle (`ora.*`) resources.

Examples

The `crsctl status resource` command returns output similar to the following:

```
$ crsctl status resource ora.stai14.vip

NAME=ora.stai14.vip
TYPE=ora.cluster_vip_net1.type
```

```
TARGET=ONLINE
STATE=ONLINE on stail14
```

The following example shows the start dependencies for a resource named `ora.newdb.db`:

```
$ crsctl status resource ora.newdb.db -dependency
ora.newdb.db(ora.database.type)

ora.ACFS_DG1.dg(ora.diskgroup.type)[hard,pullup]
ora.asm(ora.asm.type)[hard,pullup]
ora.LISTENER.lsnr(ora.listener.type)[weak]
  type:ora.cluster_vip_net1.type[hard:type,pullup:type]
  ora.net1.network(ora.network.type)[hard,pullup]
ora.dbhome_dg.dbhome_dg_v.acfs(ora.acfs.type)[hard,pullup]
ora.asm(ora.asm.type)[pullup:always]
```

crsctl stop resource

Stops running resources.

Syntax

```
crsctl stop resource {resource_name [...] | -w "filter" | -all} [-n
server_name
  | -s server_pool] [-k cid] [-d did] [-env "env1=val1,env2=val2,..."]
  [-begin | -end] [-i] [-f] [-l]
```

Parameters

Table E-17 crsctl stop resource Command Parameters

Parameter	Description
<code>resource_name [...]</code>	Alternative to using <code>-w</code> or <code>-all</code> , you can specify one or more space-delimited resource names that you want to stop.
<code>-w "filter"</code>	Alternative to specifying a resource name or using <code>-all</code> , you can specify a resource filter that CRSCTL uses to limit the number of resources stopped. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> <code>"TYPE == cluster_resource"</code>: This filter limits Oracle Clusterware to stop only resources of <code>cluster_resource</code> type <code>"CHECK_INTERVAL > 10"</code>: This filter limits Oracle Clusterware to stop resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <code>"(CHECK_INTERVAL > 10) AND (NAME co 2)"</code>: This filter limits Oracle Clusterware to stop resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <i>and</i> the name of the resource contains the number 2 <p>See Also: "Filters" for more information</p>

Table E-17 (Cont.) crsctl stop resource Command Parameters

Parameter	Description
-all	Alternative to specifying a resource name or using -w, you can use this parameter to stop all resources in the cluster.
-n <i>server_name</i> -s <i>server_pool</i>	Optionally, you can specify the name of the server on which the resource instances you want to stop reside, or the name of a server pool in which the resource instances you want to stop reside. If you do not specify a server or server pool, then CRSCTL stops all instances of the resource.
-k <i>cid</i>	Optionally, you can specify the resource cardinality ID. If you specify this parameter, then CRSCTL stops the resource instances that have the cardinality you specify.
-d <i>did</i>	Specify the resource degree ID. If you specify this parameter and the degree ID is greater than 1, then CRSCTL stops all resource instances that meet this criteria.
-env "env1=val1,env2=val2 ,..."	You can optionally override one or more resource profile attribute values with the -env command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks ("").
-begin -end	Optionally, you can specify either of these parameters to begin or end a transparent HA action, respectively.
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-f	Specify the -f parameter to force the stopping of the resource when it has other resources running that depend on it. Dependent resources are relocated or stopped when you use this parameter.
-l	Use the -l parameter to leave the resources in the state they were in if the stop command fails.

Usage Notes

- Any one of the three following options is required to specify which resources you want to stop:
 - You can specify by name one or more resources to stop.
 - You can specify a resource filter that CRSCTL uses to match resources to stop
 - You can specify the -all parameter to stop all resources on a particular server
- You must have read and execute permissions on the specified resources to stop them
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources
- Oracle does not support stopping managed applications outside of the Oracle Grid Infrastructure

Example

To stop a resource:

```
$ crsctl stop resource -n node1 -k 2
```

crsctl add resourcegroup

Creates a resource group based on a resource group type.

Use this command to create an empty resource group into which you can add resources.

Syntax

```
crsctl add resourcegroup group_name -type group_type
  [[-file file_path] | [-attr "attribute_name=
  attribute_value"]] [-f] [-i]
```

Parameters**Table E-18 crsctl add resourcegroup Command Parameters**

Parameter	Description
<i>group_name</i>	Specify the name of the resource group.
-type <i>group_type</i>	Specify a group type for the resource group.
-file <i>file_path</i> -attr	Provide either a file or a list of attributes that define the resource group.
"attribute_name=attribute_value[,...]"	Path name (either absolute or relative) for a text file containing line-delimited attribute name-value pairs that define the resource group.
	Following the -attr flag, you can specify one or more comma-delimited attribute name-value pairs enclosed in double quotations (" ").
-f	Use this parameter to add a resource group that is online.
-i	If you specify -i, then CRSCTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

crsctl check resourcegroup

Initiates a check operation on all member resources of the specified groups.

Syntax

```
crsctl check resourcegroup {group_name [...] | -w "filter"}
  [-n server_name] [-k cid]
```

Parameters

Table E-19 crsctl check resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to check.
-w " <i>filter</i> "	Alternatively, you can specify a resource group filter surrounded by double quotation marks (" ") that Oracle Clusterware uses to match resource groups. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: Filters for more information about using filters.
-n <i>server_name</i>	Optionally, you can specify the name of the server on which the resource group you want to check resides.
-k <i>cid</i>	Optionally, you can specify the resource group cardinality ID. If you specify this parameter, then Oracle Clusterware checks the resource instances that have the cardinality you specify.

crsctl delete resourcegroup

Deletes one or more resource groups from Oracle Clusterware.

Syntax

```
crsctl delete resourcegroup {"group_name [...] | -w "filter"} [-f] [-i]
```

Parameters

Table E-20 crsctl delete resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify one or more group names that you want to delete. Multiple groups must be space-delimited and enclosed in double quotation marks (" ").

Table E-20 (Cont.) crsctl delete resourcegroup Command Parameters

Parameter	Description
-w " <i>filter</i> "	<p>Specify a resource filter that Oracle Clusterware uses to limit the number of resources evaluated for deletion. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include:</p> <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to relocate only resources of cluster_resource type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute <i>and</i> the name of the resource contains the number 2 <p>Note: All operators must be surrounded by spaces. See Also: Filters for more information about operators</p>
-f	Use this parameter to delete a resource group that is online.
-i	If you specify -i, then CRSTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

Usage Notes

- You must specify either a group name (or several group names) or a resource filter.
- You can only delete a resource group if no other resource groups have dependencies to it.

crsctl eval add resourcegroup

Predicts the effects of adding a resource group without making changes to the system.

Syntax

```
crsctl eval add resourcegroup group_name -type group_type [-file file_path
| -attr "attribute_name=attribute_value[,...]"] [-f]
```

Parameters

Table E-21 crsctl eval add resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify the name of the resource group.
-type <i>group_type</i>	Specify a group type for the resource group.

Table E-21 (Cont.) crsctl eval add resourcegroup Command Parameters

Parameter	Description
<code>-file file_path -attr</code>	Provide either a file or a list of attributes that define the resource group.
<code>"attribute_name=attribute_value[,...]"</code>	Path name (either absolute or relative) for a text file containing line-delimited attribute name-value pairs that define the resource group. Following the <code>-attr</code> flag, you can specify one or more comma-delimited attribute name-value pairs enclosed in double quotation marks (" ").
<code>-f</code>	Use this parameter to add a resource group that is online.

crsctl eval fail resourcegroup

Predicts the consequences of a resource failing.

Syntax

```
crsctl eval fail resourcegroup {group_name | -w filter} [-n server]
[-env "attr=val[,...]"
```

Parameters

Table E-22 crsctl eval fail resourcegroup Command Parameters

Parameter	Description
<code>group_name</code>	Specify the name of a resource group you want to evaluate.
<code>-w filter</code>	Alternative you specifying a particular resource group, you can specify a resource group filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource groups. For example, <code>-w "TYPE = ora.database.type"</code> or <code>-w "NAME = cs1"</code> . See Also: Filters for more information about filters
<code>-n server</code>	Optionally, you can specify the name of the server to which resource groups reside that you want to evaluate.
<code>-env</code> <code>"attr=val[,...]"</code>	You can optionally override one or more resource group profile attribute values with the <code>-env</code> command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").

crsctl eval relocate resourcegroup

Predicts the effect of relocating a resource group without making changes to the system.

Syntax

```
crsctl eval relocate resourcegroup {group_name | -all} -s server | -w
filter}
    {-n server} [-f]
```

Parameters

Table E-23 crsctl eval relocate resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify the name of a resource group that you want to evaluate.
-all	Alternative to specifying a particular resource group, you can use this parameter to evaluate relocating all resource groups.
-s <i>server</i>	Specify the name of the source server on which the resource group resides that you want to relocate. This parameter is required.
-w <i>filter</i>	Alternative to specifying a particular resource group, you can specify a resource group filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource groups. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". This parameter is required if you do not use -s. See Also: Filters for more information about filters
-n <i>server</i>	Specify the name of the server to which you want to relocate the resource groups. This parameter is required.
-f	Use this parameter to evaluate the operation with the force option.

crsctl eval start resourcegroup

Predicts the effects of starting a resource group without making changes to the system.

Syntax

```
crsctl eval start resourcegroup {group_name | -w filter} [-n server] [-f]
```

Parameters

Table E-24 crsctl eval start resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to evaluate. This parameter is required.

Table E-24 (Cont.) crsctl eval start resourcegroup Command Parameters

Parameter	Description
<code>-w filter</code>	Alternatively, you can specify a resource group filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource groups. For example, <code>-w "TYPE = ora.database.type"</code> or <code>-w "NAME = cs1"</code> . This parameter is required if you do not specify a particular resource group. See Also: Filters for more information about filters
<code>-n server</code>	Specify the name of the server on which the resource group you want to evaluate resides.
<code>-f</code>	Use the <code>-f</code> parameter to evaluate a resource group on a server on which one of the resources in the resource group you want to start has a hard start dependency.

crsctl eval stop resourcegroup

Predicts the effect of stopping a resource group without making changes to the system.

Syntax

```
crsctl eval stop resourcegroup {group_name | -w filter} [-f]
```

Parameters

Table E-25 crsctl eval stop resourcegroup Command Parameters

Parameter	Description
<code>group_name</code>	Specify a resource group or a number of resource groups in a space-delimited list that you want to stop.
<code>-w "filter"</code>	Alternative to specifying a resource group, you can specify a resource group filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource groups. For example, <code>-w "TYPE = ora.database.type"</code> or <code>-w "NAME = cs1"</code> . See Also: Filters for more information about using filters.
<code>-f</code>	Use the <code>-f</code> parameter to stop a resource running on another server on which one of the resources in the resource group you want to stop has a hard stop dependency.

crsctl export resourcegroup

Exports the complete information and attributes of a resource groups and its member resources to a script file.

This command also exports configuration information of the resource group type and member resource types. You can run the script file on another cluster to import the resource group configuration.

Syntax

```
crsctl export resourcegroup [group_name [...]] -file file_name
```

Parameters

Table E-26 crsctl export resource Command Parameters

Parameter	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to export.
-file <i>file_name</i>	Specify a name for the output script file that you will run to import the group.

crsctl modify resourcegroup

Modifies a resource group to add attributes or update the values of existing attributes.

Syntax

```
crsctl modify resourcegroup group_name -attr "attribute_name=  
attribute_value" [-i] [-f] [-delete]
```

Parameters

Table E-27 crsctl modify resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify the name of the resource group you want to modify.

Table E-27 (Cont.) crsctl modify resourcegroup Command Parameters

Parameter	Description
-attr "attribute_name= attribute_value"	<p>You can specify attributes for a resource group you want to modify in two different ways:</p> <ul style="list-style-type: none"> Following the <code>-attr</code> flag, you can specify one or more comma-delimited attribute name-value pairs to modify enclosed in double quotations marks (" "). For example: <pre>-attr "CHECK_INTERVAL=30, START_TIMEOUT=25"</pre> <p>Some attributes can have multiple values. In those cases, separate the values with a space and enclose the list of values in single quotation marks. For example: <pre>-attr "SERVER_POOL_NAMES= 'ora.pool1 ora.pool2', START_TIMEOUT=25"</pre> </p> Alternatively, you can specify attribute values for resources on a particular server, with a particular cardinality value, and with a particular degree value. This method can be useful for applications that are somehow tied to a particular server. Following the <code>-attr</code> flag, the syntax is as follows: <pre>attribute_name{@SERVERNAME(server_name) [@DEGREEID(did)] @CARDINALITYID(cid) [@DEGREEID(did)]}=attribute_value</pre> <p>If you specify the <code>@SERVERNAME(server_name)</code> syntax, then the attribute value you specify for the attribute you specify is limited to resources residing on the server you specify.</p> <p>Alternatively, if you specify the <code>@CARDINALITYID(cid)</code> syntax, then the attribute value you specify for the attribute you specify is limited to resource instances with a specific cardinality ID (<code>cid</code>).</p> <p>Optionally, you can combine the <code>@DEGREEID(did)</code> syntax with either the <code>SERVERNAME</code> or <code>CARDINALITYID</code> syntax, or both, to limit the attribute value to resources with the specific <code>DEGREE</code>.</p> <p>Examples:</p> <pre>CHECK_INTERVAL@SERVERNAME(node1)=45 STOP_TIMEOUT@CARDINALITYID(2)=65 STOP_TIMEOUT@SERVERNAME(node1)@DEGREEID(2)=65 STOP_TIMEOUT@CARDINALITYID(3)@DEGREEID(2)=65</pre>
-i	<p>If you specify <code>-i</code>, then CRSTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource group or its dependents. Sometimes, Oracle Clusterware locks resource groups or other objects to prevent commands from interfering with each other.</p>

Table E-27 (Cont.) crsctl modify resourcegroup Command Parameters

Parameter	Description
-f	Use the -f parameter when: <ul style="list-style-type: none"> The resource group has a hard dependency on a non-existing resource. The owner of the resource group does not have execute permissions on one or more hard dependencies. The modification results in servers being moved into the Generic pool and resources being stopped or relocated to accomplish the server move.
-delete	If you specify the -delete parameter, then Oracle Clusterware deletes the named attribute.

Usage Notes

You can modify the name of the resource group and add or change existing attributes.

crsctl relocate resourcegroup

Relocates online resource groups from a source server to a target server.

This command first stops the resource groups you specify on the source server and then starts them on the target server.

Syntax

```
crsctl relocate resourcegroup {group_name [...] [-k cid] | -all | -w
"filter"}
[-env "attr=val[,...]" [-f] [-i]
```

Parameters

Table E-28 crsctl relocate resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to relocate.
-k <i>cid</i>	Optionally, you can specify the resource group cardinality ID. If you specify this parameter, then CRSCTL relocates the resource instances that have the cardinality you specify.
-all	Specify this parameter to relocate all resource groups on the source server.
-w " <i>filter</i> "	Specify a resource group filter surrounded by double quotation marks (") that CRSCTL uses to match resource groups. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: Filters for more information about filters
-s <i>source_server_name</i>	Specify the name of the source server on which the resource groups reside that you want to relocate. This parameter is required.

Table E-28 (Cont.) crsctl relocate resourcegroup Command Parameters

Parameter	Description
-n <i>target_server_name</i>	Optionally, you can specify the name of the server to which you want to relocate the resource groups. If you do not specify a target server, then CRSCTL relocates the resource groups on the best server according to the attribute profile of each resource group.
-env "attr=val[,...]"	You can optionally override one or more resource group profile attribute values with the -env command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
-f	Specify this option force the relocation of the resource groups you specify.
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

Example

To relocate all resource groups running on node1 to node2:

```
$ crsctl relocate resourcegroup resGrp1 -s node1 -n node2
```

crsctl restart resourcegroup

Restarts resource groups on the server where they currently reside.

Syntax

```
crsctl restart resourcegroup {group_name [...] | -w "filter"} [-k cid]
    [-env "env1=val1,env2=val2,..."] [-f] [-i]
```

Parameters

Table E-29 crsctl restart resourcegroup Command Parameters

Parameters	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to restart.
-w "filter"	Specify a resource group filter surrounded by double quotation marks (" ") that Oracle Clusterware uses to match resource groups. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: Filters for more information about filters

Table E-29 (Cont.) crsctl restart resourcegroup Command Parameters

Parameters	Description
-k <i>cid</i>	Specify the resource group cardinality ID. If you specify this parameter, then Oracle Clusterware restarts the resource instances that have the cardinality you specify.
-env "env1=val1,env2=val2 ,..."	You can optionally override one or more resource group profile attribute values with the <code>-env</code> command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
-f	Use the <code>-f</code> parameter to restart a resource, running on the server on which the resource group resides that you want to restart, that has a hard start dependency on another resource. If you do not specify the force parameter in this case, then the start command fails.
-i	If you specify <code>-i</code> , then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

Example

The following command restarts resource group `resGrp1`:

```
$ crsctl restart resourcegroup resGrp1
```

crsctl start resourcegroup

Starts one or more resource groups on specific or preconfigured servers or server pools.

CRSCTL starts resource groups to which there is a hard or weak start dependency, and all enabled member resources of the group. This command reports progress messages for each resource and resource group that CRSCTL starts.

Syntax

```
crsctl start resourcegroup {group_name [...] | -w "filter" | -all}
  [-n server_name | -s server_pool] [-k cid] [-env "attr=val[,...]" ]
  [-f] [-i] [-l]
```

Parameters**Table E-30 crsctl start resourcegroup Command Parameters**

Parameter	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to start.

Table E-30 (Cont.) crsctl start resourcegroup Command Parameters

Parameter	Description
-w <i>"filter"</i>	Specify a resource group filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource groups. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: Filters for more information about filters
-all	Specify this parameter to start all resource groups on the cluster.
-n <i>server_name</i>	Specify the name of the server on which the resource group you want to start resides. If you do not specify a server, then CRSCTL starts the resources on the best server according to the attribute profile of each resource.
-s <i>server_pool</i>	Specify a single server pool name or a space-delimited list of server pools in which a resource group resides that you want to start.
-k <i>cid</i>	Specify the resource group cardinality ID. If you specify this parameter, then CRSCTL starts the resource instances that have the cardinality you specify.
-env "attr=val[,...]"	You can optionally override one or more resource group profile attribute values with the -env command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
-f	Use the -f parameter to start a resource group on a server on which one of the resources in the resource group you want to start has a hard start dependency. If you do not specify the force parameter in this case, then the start command fails.
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-l	Use the -l parameter to leave the resource groups in the state they were in if the start command fails.

Examples

To start a resource group in a server pool:

```
$ crsctl start resourcegroup rg1 -s pool1
```

crsctl status resourcegroup

Displays the state of resource groups.

Syntax

```
crsctl status resourcegroup [group_name [...] | -w "filter"]  
[-p [-attr "attribute_name=attribute_value" [-noname]] | -v [-attr
```

```
"attribute_name=attribute_value" [-noname]] | [-f] [-l] [-g] | [[-k  
cid |  
-n server_name] [-e [-p | -v]]] [-s -k cid] | -noauto]]
```

Parameters

Table E-31 crsctl status resourcegroup Command Parameters

Parameter	Description
<i>group_name</i>	Specify a resource group or a number of resource groups in a space-delimited list that you want to query. If you do not specify a resource group name or filters, then CRSCTL displays the status for all resource groups.
-w " <i>filter</i> "	Specify a resource group filter surrounded by double quotation marks (") that Oracle Clusterware uses to match resource groups. For example, -w "TYPE = ora.database.type" or -w "NAME = cs1". See Also: Filters for more information about filters
-p -attr "attribute_name=attribute_value" [-noname]	
-v [-attr "attribute_name=attribute_value" [-noname]]	Optionally, you can specify this parameter to display the status for all member resources of a particular resource group.
-f	Optionally, you can specify this parameter to display the full configuration of the resource group.
-l	
-g	
-k <i>cid</i>	Specify the resource group cardinality ID. If you specify this parameter, then CRSCTL displays the status of the resource groups that have the cardinality you specify.
-n <i>server_name</i>	
-e	Optionally, you can specify this parameter to exclude <i>automatic</i> resource groups. The default is to display the status of all resource groups.
-p -v -s -k <i>cid</i> -noauto	

Example

The following command shows the status of resource group *rg1*, which resides on *node1*:

```
$ crsctl status resourcegroup rg1 -n node1
```

crsctl stop resourcegroup

Stops one or more resource groups on optionally specified preconfigured servers or server pools.

CRSCTL stops all member resources of the resource group. You must use the `-f` to stop the resource group if other resource groups have stop dependencies on it.

Syntax

```
crsctl stop resourcegroup {group_name [...] | -w "filter" | -all}
  [-n server_name | -s server_pool] [-k cid] [-env "attr=val[,...]"
  [-f] [-i] [-l]
```

Parameters

Table E-32 crsctl stop resourcegroup Command Parameters

Parameter	Description
<code>group_name</code>	Alternative to using <code>-w</code> or <code>-all</code> , you can specify a resource group or a number of resource groups in a space-delimited list that you want to stop.
<code>-w "filter"</code>	Alternative to specifying a resource group name or using <code>-all</code> , you can specify a resource group filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource groups. For example, <code>-w "TYPE = ora.database.type"</code> or <code>-w "NAME = cs1"</code> . See Also: Filters for more information about using filters.
<code>-all</code>	Alternative to specifying a resource name or using <code>-w</code> , you can specify this parameter to stop all resource groups on the cluster.
<code>-n server_name</code>	Specify the name of the server on which the resource group you want to stop resides. If you do not specify a server, then CRSCTL stops the resources on the best server according to the attribute profile of each resource.
<code>-s server_pool</code>	Specify a single server pool name or a space-delimited list of server pools in which a resource group resides that you want to stop.
<code>-k cid</code>	Specify the resource group cardinality ID. If you specify this parameter, then Oracle Clusterware stops the resource instances that have the cardinality you specify.
<code>-env "attr=val[,...]"</code>	You can optionally override one or more resource group profile attribute values with the <code>-env</code> command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
<code>-f</code>	Use the <code>-f</code> parameter to stop a resource running on another server on which one of the resources in the resource group you want to stop has a hard stop dependency. If you do not specify the force parameter in this case, then the stop command fails.

Table E-32 (Cont.) crsctl stop resourcegroup Command Parameters

Parameter	Description
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-l	Use the -l parameter to leave the resource groups in the state they were in if the stop command fails.

Example

The following command stops resource group `rg1`:

```
$ crsctl stop resourcegroup rg1
```

crsctl add resourcegroup type

Registers a new resource group type in Oracle Clusterware.

Oracle recommends that applications that are made up of multiple resources define a resource group type to obtain a distinct identity and also to create new group-level attributes.

Syntax

```
crsctl add resourcegroup type group_type_name -basetype base_group_type
  {-file file_path | -attr "attribute_name=attribute_value"} [-i]
```

Parameters**Table E-33 crsctl add resourcegroup type Command Parameters**

Parameter	Description
<i>group_type_name</i>	Specify the name of the resource group type.
-basetype <i>base_group_type</i>	Specify the base group type of the resource group type.
-file <i>file_path</i> - attr " <i>attribute_name=attr ibute_value</i> "	You must provide either a file or a list of attributes that define the resource group type. Path name (either absolute or relative) for a text file containing line-delimited attribute name-value pairs that define the resource group type. Following the -attr flag, you can specify one or more comma-delimited attribute name-value pairs enclosed in double quotations marks (" ").

Table E-33 (Cont.) crsctl add resourcegroup Command Parameters

Parameter	Description
-i	If you specify -i, then CRSTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

crsctl delete resourcegroup

Deletes one or more resource group types from Oracle Clusterware.

Syntax

```
crsctl delete resourcegroup { "group_type_name [...]" | -w "filter" [-i]
```

Parameters

Table E-34 crsctl delete resourcegroup Command Parameters

Parameter	Description
<i>group_type_name</i>	Specify one or more group type names that you want to delete. Multiple group types must be space-delimited and enclosed in double quotation marks (" ").
-w "filter"	Specify a resource filter that Oracle Clusterware uses to limit the number of resources evaluated for deletion. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to relocate only resources of cluster_resource type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute <i>and</i> the name of the resource contains the number 2 <p>Note: All operators must be surrounded by spaces. See Also: Filters for more information about operators</p>
-i	If you specify -i, then CRSTL returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.

crsctl modify resourcegroupstype

Modifies a resource group type to add attributes or update the values of existing attributes.

Syntax

```
crsctl modify resourcegroupstype group_type_name -attr  
"attribute_name=attribute_value"
```

Usage Notes

You can modify the name of the resource group type and add or change existing attributes.

crsctl get tracefileopts

Returns the tracing settings for a specific module.

Syntax

```
crsctl get tracefileopts module_name
```

Usage Notes

Specify one of the following modules to obtain tracing information:

- mdns: Multicast domain name server
- gpnp: Grid Plug and Play service
- css: Cluster Synchronization Services
- crf: Cluster Health Monitor
- crs: Cluster Ready Services
- ctss: Cluster Time Synchronization Service
- evm: Event Manager
- gipc: Grid Interprocess Communication

crsctl set tracefileopts

Sets the tracing parameters for a specific module.

Syntax

```
crsctl set tracefileopts module_name [-filesize file_size[K|k|M|m|G|g]]  
[-numsegments number_of_segments]
```

Parameters

Table E-35 crsctl set tracefileopts Command Parameters

Parameter	Description
<i>module_name</i>	Specify one of the following modules to set tracing information: mdns: Multicast domain name server gppp: Grid Plug and Play service css: Cluster Synchronization Services crf: Cluster Health Monitor crs: Cluster Ready Services ctss: Cluster Time Synchronization Service evm: Event Manager gipc: Grid Interprocess Communication
<code>-filesize file_size</code>	Specify the size of a single trace file in kilobytes (K k), megabytes (M m), or gigabytes (G g).
<code>-numsegments number_of_segments</code>	Specify the number of trace files to save before Oracle Clusterware rotates the older files out.

crsctl add type

Creates a resource type in Oracle Clusterware.

Syntax

```
crsctl add type type_name -basetype base_type_name {-attr
"ATTRIBUTE=attribute_name
| -file file_path,TYPE={string | int} [,DEFAULT_VALUE=default_value]
[,FLAGS=typeFlags"} [-i]
```

Parameters

Table E-36 crsctl add type Command Parameters

Parameter	Description
<i>type_name</i>	Specify a name for the resource type in the form of xxx.yyy.type. Resource type names must be unique and cannot be changed after the resource type is registered.
<code>-basetype base_type_name</code>	The name of an existing base type. Any resource type that you create must either have local_resource or cluster_resource as its base resource type.

Table E-36 (Cont.) crsctl add type Command Parameters

Parameter	Description
-attr "ATTRIBUTE= <i>attribute</i> _name	<p>You can specify the resource type attributes using the <code>-attr</code> argument. Each type attribute definition can contain up to four type attribute keywords that must be displayed in the order shown. Enter a comma-delimited description of one or more resource type attributes enclosed in double quotation marks (" "). The keywords for an attribute include:</p> <ol style="list-style-type: none"> 1. ATTRIBUTE: Specify a name for the attribute. The name is case-sensitive and cannot contain spaces. 2. TYPE: Specify whether the attribute type is <code>integer</code> or <code>string</code>. 3. DEFAULT_VALUE: (Optional) If the attribute is required, then a default value is not required. For attributes that are not required, you must specify a default value that Oracle Clusterware uses when you create resources based on this resource type. 4. FLAGS: (Optional) Specify one or more of the following types, separated by a vertical bar (): <ul style="list-style-type: none"> CONFIG: After you register a resource with this resource type, you can configure the attribute. READONLY: After you register a resource with this resource type, you cannot modify this attribute. REQUIRED: You must specify the name and value of this attribute when you create a resource that is based on this resource type. If you specify that this attribute is not required, then Oracle Clusterware uses the default value of this attribute that you specify. HOTMOD: If you change the value of an attribute for resources of this type, then the changes are applied immediately with the need to restart the resource. <p>You <i>cannot</i> use multiple <code>-attr</code> arguments to define multiple arguments for the resource type. Instead, you can specify multiple types within the double quotation marks after the <code>-attr</code> flag. For example:</p> <pre>"ATTRIBUTE=FOO,TYPE=integer,DEFAULT_VALUE=0, FLAGS=REQUIRED HOTMOD,ATTRIBUTE=BAR,TYPE=string"</pre> <p>The preceding example defines two type attributes, <code>FOO</code> and <code>BAR</code>. When you specify the <code>ATTRIBUTE</code> keyword, it ends the previous type attribute (if any) and begins a new type attribute.</p>

Table E-36 (Cont.) crsctl add type Command Parameters

Parameter	Description
<code>-file file_path</code>	<p>Path name (either absolute or relative) for a text file containing line-delimited resource type keyword-value pairs that define the resource type. An example of the contents of the file is:</p> <pre>ATTRIBUTE=FOO TYPE=integer DEFAULT_VALUE=0 FLAGS=REQUIRED ATTRIBUTE=BAR TYPE=string</pre> <p>Note: The keywords must be in the following order: ATTRIBUTE, TYPE, DEFAULT_VALUE, and FLAGS. When you specify the ATTRIBUTE keyword, it ends the previous type attribute (if any) and begins a new type attribute.</p> <p>The preceding example defines two type attributes, FOO and BAR.</p> <p>Note: All operators must be surrounded by spaces.</p> <p>See Also: "Filters" for more information about operators</p>
<code>-i</code>	<p>If you specify <code>-i</code>, then the command fails if Oracle Clusterware cannot process the request immediately.</p>



See Also:

"[Oracle Clusterware Resource Types](#)" for more information about resource types

Usage Notes

- Both the `type_name` and `base_type_name` parameters are required
- You can either specify a file containing the type information or you can specify the type information on the command line
- Do not use this command for any resource types with names that begin with `ora` because these resource types are Oracle resource types
- You must have read permissions on the base type

Example

To create a resource type for `demoActionScript`:

```
# crsctl add type test_type1 -basetype cluster_resource -attr
"ATTRIBUTE=FOO,TYPE=integer,DEFAULT_VALUE=0"
```

crsctl delete type

Removes resource types from the Oracle Clusterware configuration.

Syntax

```
crsctl delete type type_name [-i]
```

Usage Notes

- The *type_name* parameter is required. You can specify more than one type by separating each type by a space.
- If you specify *-i*, then the command fails if Oracle Clusterware cannot process the request immediately.
- Do not use this command for any resource types with names that begin with *ora* because these resource types are Oracle resource types.

Example

To delete two resource types, run the following command as a user who has write permissions on the resource type:

```
$ crsctl delete type test_type1 test_type2
```

crsctl getperm type

Obtains permissions for a particular resource type.

Syntax

```
crsctl getperm type resource_type [-u user_name] | [-g group_name]
```

Parameters

Table E-37 crsctl getperm type Command Parameters

Parameter	Description
<i>resource_type</i>	Specify the resource type for which you want to obtain permissions.
<i>-u user_name</i>	If you specify <i>-u</i> , then Oracle Clusterware obtains permissions for a particular user.
<i>-g group_name</i>	If you specify <i>-g</i> , then Oracle Clusterware obtains permissions for a particular group.

Usage Notes

- The *resource_type* parameter is required
- Do not use this command for any resource types with names that begin with *ora* because these resource types are Oracle resource types



See Also:

"[Oracle Clusterware Resource Types](#)" for more information about resource types

Example

This command returns output similar to the following:

```
$ crsctl getperm type app.appvip.type

Name: app.appvip.type
owner:root:rw,pgroup:oinstall:rw,other::r--
```

crsctl modify type

Modifies an existing resource type.

Syntax

```
crsctl modify type type_name -attr "ATTRIBUTE=attribute_name,TYPE={string
| int}
[,DEFAULT_VALUE=default_value [,FLAGS=[READONLY][| REQUIRED]]" [-i] [-f]]
```

Parameters

Table E-38 crsctl modify type Command Parameters

Parameter	Description
<i>type_name</i>	Specify the name of the resource type you want to modify. You cannot modify resource type names.
-attr	You can modify the following resource type keywords: <ul style="list-style-type: none"> • TYPE • DEFAULT_VALUE • FLAGS <p>Note: Although you must specify the ATTRIBUTE keyword, you cannot modify it.</p> <p>See Also: crsctl add type for descriptions of these keywords</p>
-i	If you specify the -i parameter, then the command fails if Oracle Clusterware cannot process the request immediately.

Usage Notes

- The *type_name* parameter is required
- Do not use this command for any resource types with names that begin with *ora* because these resource types are Oracle resource types



See Also:

"Oracle Clusterware Resource Types" for more information about resource types

Example

The following example modifies the two type attributes FOO and BAR:

```
$ crsctl modify type myType.type -attr "ATTRIBUTE=FOO,DEFAULT_VALUE=0
ATTRIBUTE=BAR,DEFAULT_VALUE=baz"
```

crsctl setperm type

Sets permissions for resources of a particular resource type.

Syntax

```
crsctl setperm type resource_type_name {-u acl_string | -x acl_string |
-o user_name | -g group_name}
```

Parameters

Table E-39 crsctl setperm type Command Parameters

Parameter	Description
<i>resource_type_name</i>	Specify the name of the resource type for which you want to set permissions.

Table E-39 (Cont.) crsctl setperm type Command Parameters

Parameter	Description
-u -x -o -g	<p>You can specify only one of the following parameters for a resource type:</p> <ul style="list-style-type: none"> -u <i>acl_string</i>: You can update the access control list (ACL) for a resource type -x <i>acl_string</i>: You can delete the ACL for a resource type -o <i>user_name</i>: You can change the owner of a resource type by entering a user name -g <i>group_name</i>: You can change the primary group of a resource type by entering a group name <p>Specify a user, group, or other ACL string, as follows:</p> <pre>user:user_name[:readPermwritePermexecPerm] group:group_name[:readPermwritePermexecPerm] other[:readPermwritePermexecPerm]</pre> <ul style="list-style-type: none"> user: User ACL group: Group ACL other: Other ACL readPerm: Read permission for the resource type; the letter r grants a user, group, or other read permission, the minus sign (-) denies read permission writePerm: Write permission for the resource type; the letter w grants a user, group, or other write permission, the minus sign (-) denies write permission execPerm: Execute permission for the resource type; the letter x grants a user, group, or other execute permission, the minus sign (-) denies execute permission

Usage Notes

- The *resource_type_name* parameter is required
- You must have read and write permissions on the specified resources to set their permissions
- Do not use this command for any resource types with names that begin with *ora* because these resource types are Oracle resource types

Example

To grant read, write, and execute permissions on a resource type for user Scott:

```
$ crsctl setperm type resType -u user:scott:rwx
```

crsctl status type

Obtains the configuration information of one or more particular resource types.

Syntax

```
crsctl status type [resource_type_name [...] | -w "filter"] [-g] [-p] [-f]
[-r]
```

Parameters

Table E-40 crsctl status type Command Parameters

Parameter	Description
<i>resource_type_name</i> [...] -w " <i>filter</i> "	Specify one or more space-delimited resource type names of which you want to check the status. Alternatively, you can specify a resource type filter surrounded by double quotation marks (" ") that CRSCTL uses to match resource types. For example, -w "TYPE = ora.database.type". See Also: " Filters " for more information
[-g] [-p] [-f] [-r]	You can specify the following parameters as options when CRSCTL checks the status of specific resource types: <ul style="list-style-type: none"> -g: Use this parameter to check if the specified resource types are registered -p: Use this parameter to display static configuration of the specified resource types -f: Use this parameter to display the full configuration of the resource types -r: Use this parameter to print all of the base types

Usage Notes

The *resource_type_name* parameter or a filter is required.

Example

The `crsctl status type` command returns output similar to the following:

```
$ crsctl status type ora.network.type

TYPE_NAME=ora.network.type
BASE_TYPE=ora.local_resource.type
```

crsctl add wallet

Creates and adds users to a wallet.

Syntax

```
crsctl add wallet -type wallet_type [-name name] [-user user_name -passwd]
```

Parameters

Table E-41 crsctl add wallet Command Parameters

Parameter	Description
<code>-type wallet_type</code>	<p>Speicfy the type of wallet you want to create, such as APPQOSADMIN, APPQOSUSER, APPQOSDB, OSUSER, or CVUDB.</p> <ul style="list-style-type: none"> • OSUSER: This wallet type stores a low-privileged Windows user's user name and password that the agent uses when you create a Windows service on a policy-managed database or in general to update the Windows service's password. • CVUDB: This wallet type stores a database user name and password that the health check component of CVU uses to connect to the database and perform database checks.
<code>-name name</code>	You must specify a name for the wallet to create APPQOSDB and CVUDB wallets.
<code>-user user_name -passwd</code>	Specify the user name you want to add to the wallet and provide the password through standard input. The user name is required to create an OSUSER wallet.

Usage Notes

If you are using a policy-managed database, then you must have a wallet. Otherwise, wallets are optional.

Example

To add a wallet:

```
$ crsctl add wallet -type OSUSER -user lp_oracle_home_user -passwd
```

In the preceding example, `lp_oracle_home_user` is a low-privileged Oracle home user who owns the home where the policy-managed database was created.

crsctl delete wallet

Removes wallets or users from a wallet.

Syntax

```
crsctl delete wallet -type wallet_type [-name name] [-user user_name]
```

Parameters

Table E-42 crsctl delete wallet Command Parameters

Parameter	Description
<code>-type wallet_type</code>	Type of wallet you want to remove, such as APPQOSADMIN, APPQOSUSER, APPQOSDB, OSUSER, or CVUDB. <ul style="list-style-type: none"> OSUSER: This wallet type stores a low-privileged Windows user's user name and password that the agent uses when you create a Windows service on a policy-managed database or in general to update the Windows service's password. CVUDB: This wallet type stores a database user name and password that the health check component of CVU uses to connect to the database and perform database checks.
<code>-name name</code>	You must specify the name of the wallet to remove an APPQOSDB wallet.
<code>-user user_name</code>	You must specify a user name to remove a user from an OSUSER wallet.

Example

To delete a user from the OSUSER wallet:

```
$ crsctl delete wallet -type OSUSER -user lp_oracle_home_user
```

In the preceding example, `lp_oracle_home_user` is a low-privileged Oracle home user who owns the home where the policy-managed database was created. Additionally, the command does not delete the wallet if it contains other users.

crsctl modify wallet

Modifies the password for a specific user in a specific wallet.

Syntax

```
crsctl modify wallet -type wallet_type [-name name] [-user user_name -  
passwd]
```


Parameters

Table E-43 crsctl modify wallet Command Parameters

Parameter	Description
<code>-type wallet_type</code>	Specify the type of wallet you want to modify, such as APPQOSADMIN, APPQOSUSER, APPQOSDB, OSUSER, or CVUDB. <ul style="list-style-type: none"> OSUSER: This wallet type stores a low-privileged Windows user's user name and password that the agent uses when you create a Windows service on a policy-managed database or in general to update the Windows service's password. CVUDB: This wallet type stores a database user name and password that the health check component of CVU uses to connect to the database and perform database checks.
<code>-name name</code>	You must specify the wallet name to modify an APPQOSDB wallet.
<code>-user user_name -passwd</code>	You must specify the user name for whom you want to modify the password. Modify the password through standard input.

Usage Notes

You cannot use this command to change a user name.

Example

To modify the password of a low-privileged Oracle home user:

```
$ crsctl modify wallet -type OSUSER -user lp_oracle_home_user -passwd
```

crsctl query wallet

Queries low-privileged users from a wallet.

Syntax

```
crsctl query wallet -type wallet_type [-name name] [-user user_name] [-all]
```

Parameters

Table E-44 crsctl query wallet Command Parameters

Parameter	Description
<code>-type wallet_type</code>	Type of wallet you want to query, such as APPQOSADMIN, APPQOSUSER, APPQOSDB, OSUSER, or CVUDB. <ul style="list-style-type: none"> OSUSER: This wallet type stores a low-privileged Windows user's user name and password that the agent uses when you create a Windows service on a policy-managed database or in general to update the Windows service's password. CVUDB: This wallet type stores a database user name and password that the health check component of CVU uses to connect to the database and perform database checks.

Table E-44 (Cont.) crsctl query wallet Command Parameters

Parameter	Description
-name <i>name</i>	You must specify the name of the wallet to query an APPQOSDB wallet.
-user <i>user_name</i>	You must specify a user name to query a user from an OSUSER wallet.
-all	Specify -all to list all of the users in a specific wallet.

Example

To list all of the users in the OSUSER wallet:

```
$ crsctl query wallet -type OSUSER -all
```

Oracle RAC Environment CRSCTL Commands

You can use the following commands only in an Oracle RAC environment.

The commands listed in this section manage the Oracle Clusterware stack in an Oracle RAC environment, which consists of the following:

- Oracle Clusterware, the member nodes and server pools
- Oracle ASM (if installed)
- Cluster Synchronization Services
- Cluster Time Synchronization Services

crsctl request action

Performs a specific action on specific resource.

Syntax

```
crsctl request action action_name {-r resource_name [...] | -w "filter"}  
[-env "env1=val1,env2=val2,..."] [-i]
```

Parameters**Table E-45 crsctl request action Command Parameters**

Parameter	Description
<i>action_name</i>	Specify the name of the action you want perform. Actions supported by a particular resource are listed in the ACTIONS resource attribute of that resource.
-r <i>resource_name</i> [...]	Specify a particular resource. Multiple resource names must be separated by a space.

Table E-45 (Cont.) crsctl request action Command Parameters

Parameter	Description
-w " <i>filter</i> "	<p>Alternative to specifying resource names, you can specify a resource filter that Oracle Clusterware uses to limit the number of resources on which actions are performed. Examples of resource filters include:</p> <ul style="list-style-type: none"> • <code>TYPE == cluster_resource</code>: This filter limits Oracle Clusterware to perform actions on only resources of <code>cluster_resource</code> type • <code>CHECK_INTERVAL > 10</code>: This filter limits Oracle Clusterware to perform actions on only resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute • <code>(CHECK_INTERVAL > 10) AND (NAME co 2)</code>: This filter limits Oracle Clusterware to perform actions on only resources that have a value greater than 10 for the <code>CHECK_INTERVAL</code> resource attribute <i>and</i> the name of the resource contains the number 2 <p>See Also: "Filters" for more information</p>
-env " <i>env1=val1,env2=val2</i> <i>,...</i> "	<p>You can optionally override one or more resource profile attribute values with the <code>-env</code> command parameter. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks ("<code>"</code>").</p>
-i	<p>If you specify <code>-i</code>, then the command fails if Oracle Clusterware cannot process the request immediately.</p>

Usage Notes

The *action_name* must be a supported action, defined both in the agent and in the `ACTIONS` resource attribute, and the user must have permission to run the action. You can set permissions in the `ACTIONS` resource attribute.

Example

To initiate the specified action inside the application specific agent of a particular resource:

```
$ crsctl request action action1 -r res1
```

crsctl add category

Adds a server category.

Syntax

```
crsctl add category category_name [-attr "attr_name=attr_value  
[,attr_name=attr_value[,...]]"] [-i]
```

Parameters

Table E-46 crsctl add category Command Parameters

Parameter	Description
<i>category_name</i>	Specify a name for the server category you want to add.
-attr "attr_name=attr_value[,...]"	Specify the name-value pair of a category attribute you want to add preceded by the -attr flag. Note: The <i>attr_name</i> and <i>attr_value</i> parameters must be enclosed in double quotation marks (" "), and separate multiple name-value pairs with commas. For example: -attr "EXPRESSION='(CPU_COUNT > 2) AND (MEMORY_SIZE > 2048)'" See Also: <ul style="list-style-type: none"> • Table 3-4 for a list of attributes you can use with this command • "Filters" for information about operators
-i	If you specify -i, then the command fails if Oracle Clusterware cannot process the request immediately.

Usage Notes

If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' '). For example:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```

Example

To add a server category:

```
$ crsctl add category cat1 -attr "EXPRESSION='(CPU_COUNT > 2) AND (MEMORY_SIZE > 2048)'"
```

crsctl delete category

Deletes a server category.

Syntax

```
crsctl delete category category_name [category_name [...]] [-i]
```

Usage Notes

- Specify the name of the server category or a space-delimited list of server categories that you want to delete.
- If you specify -i, then the command fails if Oracle Clusterware cannot process the request immediately.

Example

To delete a server category:

```
$ crsctl delete category blue_server -i
```

crsctl modify category

Modifies an existing server category.

Syntax

```
crsctl modify category category_name [-attr "attr_name=attr_value  
[,attr_name=attr_value[,...]]" [-i] [-f]
```

Parameters

Table E-47 crsctl modify category Command Parameters

Parameter	Description
<i>category_name</i>	Specify the name of the server category you want to modify.
-attr "attr_name=attr_value e[,...]"	Specify the name-value pair of a category attribute you want to modify preceded by the -attr flag. Note: The <i>attr_name</i> and <i>attr_value</i> parameters must be enclosed in double quotation marks (" "), and separate multiple name-value pairs with commas. For example: "ACL='owner:st-cdc\cdctest:rwx,pggrp:---', ACTIVE_CSS_ROLE=hub" See Also: <ul style="list-style-type: none"> • Table 3-4 for a list of attributes you can use with this command • "Filters" for information about operators
-i	If you specify -i, then the command fails if Oracle Clusterware cannot process the request immediately.
-f	Use this parameter to force the operation.

Usage Notes

If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' '). For example:

```
"START_DEPENDENCIES=s'hard(res1,res2,res3)'"
```

Example

To modify a server category:

```
$ crsctl modify category blue_server -attr "EXPRESSION=(LOCATION=hub)"
```

crsctl status category

Obtains information about a server category.

Syntax

```
crsctl status category {category_name [category_name [...]] | [-w "filter"  
|  
-server server_name]}
```

Parameters

Table E-48 crsctl status category Command Parameters

Parameter	Description
<i>category_name</i> [...]	Specify the name of the server category or a space-delimited list of server categories for which you want to obtain the status.
-w " <i>filter</i> "	Alternatively, you can specify a category filter preceded by the -w flag. See Also: " Filters " for more information
-server <i>server_name</i>	Alternatively, you can specify a particular server to list all of the categories that the server matches.

Examples

To obtain the status of a server category using filters:

```
$ crsctl stat category -w "ACTIVE_CSS_ROLE = hub"
```

```
NAME=my_category_i  
ACL=owner:scott:rwx,pgrp:svrtech:rwx,other::r--  
ACTIVE_CSS_ROLE = hub  
EXPRESSION=(CPU_COUNT > 3)
```

To obtain the status of a server category by server:

```
$ crsctl stat category -server node1
```

```
NAME=my_category  
ACL=owner:scott:rwx,pgrp:svrtech:rwx,other::r--  
ACTIVE_CSS_ROLE = hub  
EXPRESSION=
```

crsctl check cluster

Checks the status of the Oracle Clusterware stack from any node in the cluster.

Syntax

```
crsctl check cluster [-all | [-n server_name [...]]]
```

Usage Notes

- You can check the status of the Oracle Clusterware stack on all nodes in the cluster with the `-all` parameter or you can specify one or more space-delimited nodes. If you do not specify either parameter, then Oracle Clusterware checks the status of the Oracle Clusterware stack on the local server.
- You can use this cluster-aware command on any node in the cluster.

Example

The `crsctl check cluster` command returns output similar to the following:

```
$ crsctl check cluster -all
*****
node1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*****
node2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*****
```

crsctl start cluster

Starts the Oracle Clusterware stack when run on any node in the cluster.

Syntax

```
crsctl start cluster [[-all] | [-n server [...]] | -hub | -w "filter"]
```

Parameters

Table E-49 crsctl start cluster Command Parameters

Parameter	Description
<code>-all</code>	Use this parameter to start the Oracle Clusterware stack on all servers in the cluster.
<code>-n server [...]</code>	Optionally, you can specify one or more named servers in the cluster (separate multiple server names by a space), or the local server, if you do not specify either <code>-all</code> or <code>-n</code> .
<code>-hub</code>	Optionally, you can specify <code>-hub</code> to start the Oracle Clusterware stack on Hub Nodes in the cluster.

Table E-49 (Cont.) crsctl start cluster Command Parameters

Parameter	Description
-w " <i>filter</i> "	Optionally, you can use this parameter to start only servers that match a filter, such as "ACTIVE_CSS_ROLE = hub". See Also: Filters for more information about filters.

crsctl stop cluster

Stops the Oracle Clusterware stack on all servers in the cluster or specific servers when you run the command on any node in the cluster.

Syntax

```
crsctl stop cluster [[-all] | [-n server [...]] | -hub | -w "filter"] [-f]
```

Parameters

Table E-50 crsctl stop cluster Command Parameters

Parameter	Description
-all	Use this parameter to start the Oracle Clusterware stack on all servers in the cluster.
-n <i>server</i> [...]	Optionally, you can specify one or more named servers in the cluster (separate multiple server names by a space), or the local server, if you do not specify either -all or -n.
-hub	Optionally, you can specify -hub to stop the Oracle Clusterware stack on a specific Hub Node in the cluster.
-w " <i>filter</i> "	Optionally, you can use this parameter to stop only servers that match a filter, such as "ACTIVE_CSS_ROLE = hub". See Also: Filters for more information about filters.

Usage Notes

- If you do not specify -all or one or more space-delimited server names, then Oracle Clusterware stops the Oracle Clusterware stack on the local server.
- You can use this cluster-aware command on any node in the cluster.
- This command attempts to gracefully stop resources managed by Oracle Clusterware while attempting to stop the Oracle Clusterware stack.

If any resources that Oracle Clusterware manages are still running after you run the `crsctl stop cluster` command, then the command fails. Use the `-f` parameter to unconditionally stop all resources and stop the Oracle Clusterware stack.

- If you intend to stop Oracle Clusterware on all or a list of nodes, then use the `crsctl stop cluster` command, because it prevents certain resources from being relocated to other servers in the cluster before the Oracle Clusterware stack is stopped on a particular server. If you must stop the Oracle High Availability

Services on one or more nodes, then wait until the `crsctl stop cluster` command completes and then run the `crsctl stop crs` command on any particular nodes, as necessary.

Example

To stop the Oracle Clusterware stack on a particular server:

```
# crsctl stop cluster -n node1
```

crsctl get cluster class

Obtains cluster class information.

Syntax

```
crsctl get cluster class
```

Usage Notes

This command has no parameters.

crsctl get cluster configuration

Displays the cluster configuration information.

Syntax

```
crsctl get cluster configuration
```

Usage Notes

This command has no parameters.

crsctl set cluster disabledtlsciphersuite

Adds a cipher suite to or removes a cipher suite from a list of disabled cipher suites used to secure inter-node communication.

Syntax

```
crsctl set cluster disabledtlsciphersuite add | delete cipher_suite_name
```

Usage Notes

Use the `add` or `delete` verb to perform the operation you want and specify the name of the cipher suite you want to add or remove.

crsctl get cluster extended

Determines whether the cluster is an extended cluster.

Syntax

```
crsctl get cluster extended
```

Usage Notes

This command has no parameters.

Example

This command returns output similar to the following:

```
$ crsctl get cluster extended
CRS-XXXX: The cluster is 'EXTENDED'
$ crsctl get cluster extended
CRS-XXXX: The cluster is 'NOT EXTENDED'
```

crsctl get cluster hubsize

Obtains the value of Hub Nodes in an Oracle Flex Cluster.

Syntax

```
crsctl get cluster hubsize
```

Example

This command returns output similar to the following:

```
CRS-4950: Current hubsize parameter value is 32
```

crsctl set cluster hubsize

Sets the maximum number of Hub Nodes for an Oracle Flex Cluster.

Syntax

```
crsctl set cluster hubsize
```

Example

The following command example sets the maximum number of Hub Nodes to 32:

```
$ crsctl set cluster hubsize 32
```

crsctl get cluster mode

Ascertain whether the cluster is configured for Oracle Flex Clusters or obtains the current status.

Syntax

```
crsctl get cluster mode [config | status]
```

Usage Notes

- Specify the `config` option to obtain the mode in which the cluster is configured.
- Specify the `status` option to obtain the current status of the cluster.

crsctl set cluster mode

Changes a standard cluster to an Oracle Flex Cluster.

Syntax

```
crsctl set cluster mode flex
```

Usage Notes

This command takes only the `flex` parameter, which is required.

crsctl get cluster name

Displays the current cluster name.

Syntax

```
crsctl get cluster name
```

Usage Notes

This command has no parameters.

crsctl add cluster site

Use the `crsctl add cluster site` command to add a site to an extended cluster.

Syntax

```
$ crsctl add cluster site site_name [-guid site_guid]
```

Usage Notes

- You must specify a name for the site you are adding. Optionally, you can provide a GUID for the site.
- An extended cluster can have a maximum of five sites.
- You cannot run this command in a cluster that is not extended.

Examples

The following command example adds a site called `SiteA` to an extended cluster:

```
$ crsctl add cluster site SiteA
```

crsctl delete cluster site

Use the `crsctl delete cluster site` command to delete a specific site from an extended cluster.

Syntax

```
$ crsctl delete cluster site site_name
```

Usage Notes

- You must specify the name of the site that you want to delete.
- You cannot run this command in a cluster that is not extended.
- You cannot delete a site that has nodes or disks. You must delete the nodes and drop the disks before you can delete the site.

Examples

The following command deletes a site named `SiteA` from an extended cluster:

```
$ crsctl delete cluster site SiteA
```

crsctl modify cluster site

Modifies the configuration of a site in an extended cluster.

Syntax

```
crsctl modify cluster site site_name {-n node1, node2, ... | -s  
{rejuvenate | quarantine} | [-local]}
```

Parameters

Table E-51 crsctl modify cluster site Command Parameters

Parameter	Description
<i>site_name</i>	Specify the name of the site you want to modify.
-n <i>node1, node2, ...</i>	Specify this parameter to make a global change in Oracle Cluster Registry (OCR) and also push the changes to the Oracle Local Registry (OLR) on all the nodes.

Table E-51 (Cont.) crsctl modify cluster site Command Parameters

Parameter	Description
-s {rejuvenate quarantine}	<p>Specify a state for the site.</p> <ul style="list-style-type: none"> rejuvenate: Use this option to enable a previously quarantined site. A site is typically quarantined by Oracle Clusterware and Oracle ASM when it detects a storage split. quarantine: Use this option if you want to explicitly quarantine a site. Oracle Clusterware will not start any database instances nor Oracle ASM on a quarantined site. In the event of a storage split, Oracle Clusterware detects the condition and sets the site to quarantine status.
-local	<p>Optionally, you can specify this parameter to make changes only to the Oracle Local Registry on the local node. Oracle provides this command for certain scripts that can run this when OCR is not available or for a user, if the node goes down when a global change is performed.</p>

Examples

The following command example adds a node called `node11` to the SiteA:

```
$ crsctl modify cluster site SiteA -n node1, node2,node11
```

crsctl query cluster site

Displays site information in an extended cluster.

Syntax

```
$ crsctl query cluster site {site_name | -all}
```

Usage Notes

- You can specify a particular site by name or you use `-all` to get the information for all the sites in the extended cluster.
- The command returns the names of the sites, the site GUID, the site state, and the components of the site.
- The Oracle Clusterware technology stack must be active and up to run this command.
- An extended cluster can have up to five sites.

Examples

This command returns output similar to the following:

```
$ crsctl query cluster site -all
Site 'SiteA' identified by GUID '7b7b3bef4c1f5ff9ff8765bceb45433a' in
state 'ONLINE' contains nodes 'node1,node2,node3', and disks 'disk1,
disk2, disk3'.
```

```
Site 'SiteB' identified by GUID '23453bef4c1f5ff9ff8765bceb45433a' in
state 'QUARANTINED' contains nodes 'node4, node5, node6', and disks
'disk4, disk5, disk6'.
Site 'SiteQ' identified by GUID '98763bef4c1f5ff9ff8765bceb45433a' in
state 'ONLINE' contains no nodes and disk 'disk7'.
Site 'SiteD' identified by GUID '4abcd453c6bc6fc1ffd3a58849d5ba41' in
state 'ONLINE' contains nodes 'cuj1234' and no disk.
```

crsctl get cluster tlsciphersuite

Lists available cipher suites used by transport layer security for securing inter-node communication.

Syntax

```
crsctl get cluster tlsciphersuite [disabled|all]
```

Usage Notes

Optionally, you can specify `disabled` to list only those cipher suites disabled for use, or you can specify `all` to list all cipher suites.

crsctl get cluster type

Use the `crsctl get cluster type` command to determine whether the cluster is an Oracle Flex Cluster or an application cluster.

Syntax

```
crsctl get cluster type
```

This command returns the cluster type, either `flex` or `application`.

crsctl set cluster type

Sets the type of the cluster.

Syntax

```
crsctl Set cluster type flex | application
```

Usage Notes

Specify either `flex` or `application` to choose which type of cluster you want to set.

crsctl get cpu equivalency

The `crsctl get cpu equivalency` command obtains the value of the `CPU_EQUIVALENCY` server configuration attribute.

Syntax

```
crsctl get cpu equivalency
```

Example

This command returns output similar to the following:

```
$ crsctl get cpu equivalency
```

```
CRS-4963: Current cpu equivalency parameter value is 1000
```

Related Topics

- [Server Configuration and Server State Attributes](#)
Oracle Clusterware assigns each server a set of attributes as soon as you add a server to a cluster.

crsctl set cpu equivalency

The `crsctl set cpu equivalency` command sets a value for the `CPU_EQUIVALENCY` server configuration attribute.

Syntax

```
crsctl set cpu equivalency number
```

Usage Notes

This command takes only a numerical value to set the `CPU_EQUIVALENCY` server configuration attribute.

Example

The following example sets the value of the `CPU_EQUIVALENCY` server configuration attribute to 2000:

```
$ crsctl set cpu equivalency 2000
```

```
CRS-4416: Server attribute 'CPU_EQUIVALENCY' successfully changed.  
Restart Oracle High Availability Services for new value to take effect.
```

Related Topics

- [Server Configuration and Server State Attributes](#)
Oracle Clusterware assigns each server a set of attributes as soon as you add a server to a cluster.

crsctl get restricted placement

The `crsctl get restricted placement` command obtains the value of the placement policy for resources that use `WORKLOAD_CPU` server configuration attribute.

Syntax

```
crsctl get restricted placement
```

Example

This command returns output similar to the following:

```
$ crsctl get restricted placement
CRS-46857: Current restricted placement value (of PACK_FLAVOR attribute)
is 'spread'
```

Related Topics

- [Server Configuration and Server State Attributes](#)
Oracle Clusterware assigns each server a set of attributes as soon as you add a server to a cluster.

crsctl set restricted placement

The `crsctl set restricted placement` command sets a value for the placement policy for resources that use the `WORKLOAD_CPU` server configuration attribute.

Syntax

```
crsctl set restricted placement {fill|spread}
```

Options

Option	Description
fill	Sets the placement policy for <code>WORKLOAD_CPU</code> resources to the value of <code>fill</code> .
spread	Sets the placement policy for <code>WORKLOAD_CPU</code> resources to the value of <code>spread</code> .

Usage Notes

Specify either `fill` or `spread` to set the value of the placement policy for resources that use the `WORKLOAD_CPU` attribute.

Example

The following example sets the value of the placement policy for resources that use `WORKLOAD_CPU` to `fill`.

```
$ crsctl set restricted placement fill
```


Related Topics

- [Server Configuration and Server State Attributes](#)
Oracle Clusterware assigns each server a set of attributes as soon as you add a server to a cluster.

crsctl check crs

Checks the status of Oracle High Availability Services and the Oracle Clusterware stack on the local server.

Syntax

```
crsctl check crs
```

Example

This command returns output similar to the following:

```
CRS-4638: Oracle High Availablity Services is online  
CRS-4537: Cluster Ready Services is onlin  
CRS-4529: Cluster Synchronization Services is online  
CRS-4533: Event Manager is online
```

crsctl config crs

Displays Oracle High Availability Services automatic startup configuration.

Syntax

```
crsctl config crs
```

Example

This command returns output similar to the following:

```
CRS-4622: Oracle High Availability Services autostart is enabled.
```

crsctl disable crs

Prevents the automatic startup of Oracle High Availability Services when the server boots.

Syntax

```
crsctl disable crs
```

Usage Notes

- This command only affects the local server

- If you disable Oracle High Availability Services automatic startup, you must use the `crsctl start crs` command to start Oracle High Availability Services

Example

This command returns output similar to the following:

```
CRS-4621: Oracle High Availability Services autostart is disabled.
```

crsctl enable crs

Enables automatic startup of Oracle High Availability Services when the server boots.

Syntax

```
crsctl enable crs
```

Usage Notes

This command only affects the local server

Example

This command returns output similar to the following:

```
CRS-4622: Oracle High Availability Services autostart is enabled.
```

crsctl start crs

Starts Oracle High Availability Services on the local server.

Syntax

```
crsctl start crs [-excl [-nocrs] [-cssonly]] | [-wait | -waithas | -nowait] | [-noautostart]
```

Parameters

Table E-52 crsctl start crs Command Parameters

Parameter	Description
<code>-excl [-nocrs] [-cssonly]</code>	Starts Oracle Clusterware in exclusive mode with two options: <ul style="list-style-type: none"> • Specify the <code>-nocrs</code> parameter to start Oracle Clusterware in exclusive mode without starting CRSD. • Specify the <code>-cssonly</code> parameter to start CSSD, only.

Table E-52 (Cont.) crsctl start crs Command Parameters

Parameter	Description
-wait -waithas -nowait	Choose one of the following: <ul style="list-style-type: none"> Specify <code>-wait</code> to wait until startup is complete and display all progress and status messages. Specify <code>-waithas</code> to wait until startup is complete and display OHASD progress and status messages. Specify <code>-nowait</code> to not wait for OHASD to start
-noautostart	Start only OHASD.

Usage Notes

- You must run this command as `root`
- This command starts Oracle High Availability Services only on the local server

Example

To start Oracle High Availability Services on the local server:

```
# crsctl start crs
```

crsctl stop crs

Stops Oracle High Availability Services on the local server.

Syntax

```
crsctl stop crs [-f]
```

Usage Notes

- You must run this command as `root`.
- This command attempts to gracefully stop resources managed by Oracle Clusterware while attempting to stop Oracle High Availability Services on the local server.

If any resources that Oracle Clusterware manages are still running after you run the `crsctl stop crs` command, then the command fails. Use the `-f` parameter to unconditionally stop all resources and stop Oracle High Availability Services on the local server.
- If you intend to stop Oracle Clusterware on all or a list of nodes, then use the [crsctl stop cluster](#) command, because it prevents certain resources from being relocated to other servers in the cluster before the Oracle Clusterware stack is stopped on a particular server. If you must stop the Oracle High Availability Services on one or more nodes, then wait until the `crsctl stop cluster` command completes and then run the `crsctl stop crs` command on any particular nodes, as necessary.
- Before attempting to shut down the Oracle Clusterware technology stack on all nodes with an Oracle ASM instance running in parallel in an Oracle Clusterware

standard Cluster with Oracle Flex ASM enabled, you must first relocate at least one Oracle ASM instance to another node where Oracle ASM is *not* running.

- CRSCTL uses a `shutdown abort stop` option when running this command to improve high availability and failover times.
- If you shut down the database with SRVCTL, before you run `crsctl stop crs`, then you must restart the resources after Oracle Clusterware comes up.
- In Oracle Clusterware 11g release 2 (11.2.0.3), when you run this command in Solaris Sparc and Solaris X64 environments, drivers remain loaded on shutdown and subsequent startup. This *does not* happen in Linux environments.

Example

To stop Oracle High Availability Services on the local server:

```
# crsctl stop crs
```

crsctl query crs activeversion

Displays the active version and the configured patch level of the Oracle Clusterware software running in the cluster.

During a rolling upgrade, however, the active version is not advanced until the upgrade is finished across the cluster, until which time the cluster operates at the pre-upgrade version. Additionally, during a rolling patch, the active patch level is not advanced until the patching is finished across the cluster, until which time the cluster operates at the pre-upgrade patch level.

Syntax

```
crsctl query crs activeversion [-f]
```

Usage Notes

If you specify the `-f` parameter, then this command also prints the patch level for each configured node in the cluster.

Example

This command returns output similar to the following:

```
Oracle Clusterware active version on the cluster is [12.2.0.1.0]. The
cluster
upgrade state is [NORMAL]. The cluster active patch level is [456789126].
```

crsctl add crs administrator

Adds a user to the list of cluster administrators.

Syntax

```
crsctl add crs administrator -u user_name [-f]
```

Parameters

Table E-53 crsctl add crs administrator Command Parameters

Parameter	Description
-u <i>user_name</i>	The name of the user to whom you want to give Oracle Clusterware administrative privileges.
-f	Use this parameter to override the user name validity check.

Usage Notes

- This command is deprecated in Oracle Clusterware 12c.
- You must run this command as `root` or a cluster administrator, or an administrator on Windows systems
- By default, `root`, the user that installed Oracle Clusterware, and the `*` wildcard are members of the list of users who have Oracle Clusterware administrative privileges. Run the [crsctl delete crs administrator](#) command to remove the wildcard and enable role-separated management of Oracle Clusterware.

Example

To add a user to the list of Oracle Clusterware administrators:

```
# crsctl add crs administrator -u scott
```

Related Topics

- [Role-Separated Management](#)
Role-separated management is an approach to managing cluster resources and workloads in a coordinated fashion in order to reduce the risks of resource conflicts and shortages.

crsctl delete crs administrator

Removes a user from the list of Oracle Clusterware administrators.

Syntax

```
crsctl delete crs administrator -u user_name [-f]
```

Parameters

Table E-54 crsctl delete crs administrator Parameters

Parameter	Description
<code>-u <i>user_name</i></code>	The name of the user whose Oracle Clusterware administrative privileges you want to remove. By default, the list of users with Oracle Clusterware administrative privileges consists of the user who installed Oracle Clusterware, <code>root</code> , and <code>*</code> . The user who installed Oracle Clusterware and <code>root</code> are permanent members this list. The <code>*</code> value gives Oracle Clusterware administrative privileges to all users and must be removed to enable role-separated management.
<code>-f</code>	Use this parameter to override the user name validity check.

Usage Notes

- The `user_name` parameter is required
- You must run this command as `root` or a cluster administrator, or an administrator on Windows systems
- To enable role-separated management, you must remove the `*` value enclosed in double quotation marks (" ")

Example

To remove a user from the list of cluster administrators:

```
# crsctl delete crs administrator -u scott
```



See Also:

["Role-Separated Management"](#)

crsctl query crs administrator

Displays the list of users with Oracle Clusterware administrative privileges.

Syntax

```
crsctl query crs administrator
```

Usage Notes

Add additional information about the command here.

Example

This command returns output similar to the following:

```
CRS Administrator List: scott
```

crsctl query crs autostart

Obtains the values of the Oracle Clusterware automatic resource start criteria.

Syntax

```
crsctl query crs autostart
```

Example

This command returns output similar to the following:

```
'Autostart delay':      60
'Autostart servercount': 2
```

crsctl set crs autostart

Sets the Oracle Clusterware automatic resource start criteria. The autostart delay and minimum server count criteria delay Oracle Clusterware resource autostart until one of the two conditions are met.

Syntax

```
crsctl set crs autostart [delay delay_time] [servercount count]
```

Parameters

Table E-55 crsctl set crs autostart Command Parameters

Parameter	Description
delay <i>delay_time</i>	Specify the number of seconds to delay Oracle Clusterware autostart.
servercount <i>count</i>	Specify the minimum number of servers required for Oracle Clusterware autostart.

Example

To ensure that Oracle Clusterware delays resource autostart for 60 seconds after the first server in the cluster is ONLINE:

```
$ crsctl set crs autostart delay 60
```

To ensure that Oracle Clusterware waits for there to be at least two servers ONLINE before it initiates resource autostart:

```
$ crsctl set crs autostart servercount 2
```

To ensure that Oracle Clusterware delays resource autostart until either of the previous two conditions are met (in no particular order):

```
$ crsctl set crs autostart delay 60 servercount 2
```

crsctl query crs releasepatch

Displays the patch level which is updated in the Grid home patch repository while patching a node.

The patch level corresponds to only the local node on which you run the command. You can run this command while the Oracle Clusterware stack is not running.

Syntax

```
crsctl query crs releasepatch
```

Example

This command returns output similar to the following for a node which has no patches applied:

```
Oracle Clusterware release patch level is [3180840333] and the complete  
list of  
patches is [13559647] on the local node.
```

crsctl query crs releaseversion

Displays the version of the Oracle Clusterware software stored in the binaries on the local node.

Syntax

```
crsctl query crs releaseversion
```

Example

This command returns output similar to the following:

```
Oracle High Availability Services release version on the local node is  
[12.2.0.1.0]
```

crsctl query crs site

Use the `crsctl query crs site` command to determine to which site a specific host or disk belongs.

Syntax

```
crsctl query crs site {-n host_name | -d disk_name}
```

Usage Notes

- You must specify either a host name or a disk name.
- The Oracle Clusterware technology stack must be active to run this command.

Examples

This command returns output similar to the following:

```
$ crsctl query crs site -n hostA
Host 'hostA' is a part of site 'SiteA'
```

crsctl query crs softwarepatch

Displays the configured patch level of the installed Oracle Clusterware.

Syntax

```
crsctl query crs softwarepatch [host_name]
```

Usage Notes

You can optionally specify a host name to display the patch level of Oracle Clusterware installed on that particular host. Otherwise, CRSCTL displays the patch level of Oracle Clusterware installed on the local host.

Example

This command returns output similar to the following:

```
Oracle Clusterware patch level on node [node1] is [456789126]
```

crsctl query crs softwareversion

Displays the latest version of the software that has been successfully started on the specified node.

Syntax

```
crsctl query crs softwareversion [node_name]
```

Usage Notes

If you do not provide a node name, then Oracle Clusterware displays the version of Oracle Clusterware running on the local server.

Example

This command returns output similar to the following:

```
Oracle Clusterware version on node [node1] is [12.2.0.1.0]
```

crsctl get css

Obtains the value of a specific Cluster Synchronization Services parameter.

Syntax

```
crsctl get css parameter
```

Usage Notes

- Cluster Synchronization Services parameters and their default values include:

```
clusterguid  
diagwait  
disktimeout (200 (seconds))  
misscount (30 (seconds))  
reboottime (3 (seconds))  
priority (4 (UNIX), 3 (Windows))  
logfilesize (50 (MB))
```

- This command does not display default values.
- This command only affects the local server.

Example

This command returns output similar to the following:

```
$ crsctl get css disktimeout  
CRS-4678: Successful get disktimeout 200 for Cluster Synchronization  
Services.
```

crsctl pin css

Pins many specific nodes.

Pinning a node means that the association of a node name with a node number is fixed. If a node is not pinned, its node number may change if the lease expires while it is down. The lease of a pinned node never expires.

Syntax

```
crsctl pin css -n node_name [ node_name [...]]
```

Usage Notes

- You can specify a space-delimited list of servers

- Any pre-12c release 1 (12.1) Oracle software must reside on a pinned server.
- A node may be unpinned with [crsctl unpin css](#).
- Deleting a node with the [crsctl delete node](#) command implicitly unpins the node.

Example

To pin the node named `node2`:

```
# crsctl pin css -n node2
```

crsctl set css

Sets the value of a Cluster Synchronization Services parameter.

Syntax

```
crsctl set css parameter value
```

Usage Notes

- Do not use the `crsctl set css` command to set the following parameters unless instructed to do so by My Oracle Support.
- Cluster Synchronization Services parameters include:

```
diagwait
disktimeout
logfilesize
misscount
priority
reboottime
```

crsctl unpin css

Unpins many servers.

If a node is not pinned, its node number may change if the lease expires while it is down.

Syntax

```
crsctl unpin css -n node_name [node_name [...exit]]
```

Usage Notes

- You can specify a space-delimited list of nodes.
- Unpinned servers that stop for longer than a week are no longer reported by `olsnodes`. These servers are dynamic when they leave the cluster, so you do not need to explicitly remove them from the cluster.
- Deleting a node with the [crsctl delete node](#) command implicitly unpins the node.

- During upgrade of Oracle Clusterware, all servers are pinned, whereas after a fresh installation of Oracle Clusterware 12c, all servers you add to the cluster are unpinned.
- You *cannot* unpin a server that has an instance of Oracle RAC that is older than 12c release 1 (12.1) if you installed Oracle Clusterware 12c on that server.

Example

To unpin two nodes:

```
$ crsctl unpin css -n node1 node4
```

crsctl unset css

Unsets the value of a Cluster Synchronization Services parameter and restores it to its default value.

Syntax

```
crsctl unset css parameter
```

Usage Notes

- You can specify the following Cluster Synchronization Services parameters:
 - diagwait
 - disktimeout
 - misscount
 - reboottime
 - priority
 - logfilesize

Example

To restore the `reboottime` Cluster Synchronization Services parameter to its default value:

```
$ crsctl unset css reboottime
```

crsctl get ipmi binaryloc

Retrieves the location of the `ipmiutil` binary.

Syntax

```
get ipmi binaryloc
```

Usage Notes

Run the command to retrieve the `ipmiutil` binary location on each node. A value is returned only if the location has been previously set with the `set ipmi binaryloc` command.

Examples

To retrieve the `ipmiutil` binary location on a node:

```
crsctl get ipmi binaryloc
```

crsctl set ipmi binaryloc

Sets the location of the `ipmiutil` binary.

Syntax

```
set ipmi binaryloc binary_location
```

Usage Notes

Run the command to set the `ipmiutil` binary location on each node.

Before running any other CRSCTL commands that modify the IPMI configuration, the `crsctl set ipmi binaryloc` command must be run on all nodes where the binary location has changed.

Examples

To set the `ipmiutil` binary location on a node:

```
crsctl set ipmi binaryloc /usr/bin/ipmiutil
```

crsctl get css ipmiaddr

Retrieves the address stored in the Oracle Local Registry of the local Intelligent Platform Management Interface (IPMI) device.

Syntax

```
crsctl get css ipmiaddr
```

Usage Notes

- Run the command under the user account used to install Oracle Clusterware.
- This command only obtains the IP address stored in the Oracle Local Registry. It may not be the IP address actually used by IPMI.

Use either `ipmiutil` or `ipmitool` as root on the local server to obtain the IP address used by the IPMI device.

crsctl set css ipmiaddr

Stores the address of the local Intelligent Platform Management Interface (IPMI) device in the Oracle Local Registry.

Syntax

```
crsctl set css ipmiaddr ip_address
```

Usage Notes

- Run the command under the user account used to install Oracle Clusterware
- Obtain the IP address used by the IPMI device using either `ipmiutil` or `ipmitool` as `root` on the local server
- Oracle Clusterware stores the IP address for IPMI in the configuration store, and distributes the address as required
- This command only stores the IPMI IP address on the server from which you run it
- This command fails if another server cannot access IPMI at the supplied address

Example

To store the IPMI IP address on a local server and distribute it to other cluster nodes:

```
$ crsctl set css ipmiaddr 192.0.2.244
```

crsctl set css ipmiadmin

Stores the login credentials of an Intelligent Platform Management Interface (IPMI) administrator in the Oracle Local Registry.

Syntax

```
crsctl set css ipmiadmin ipmi_administrator_name
```

Usage Notes

- This command must be run under the user account that installed Oracle Clusterware.
- When prompted, provide the new password to associate with the new administrator account name. Oracle Clusterware stores the name and password for the local IPMI in the configuration store, and distributes the new credentials as required.
- This command only modifies the IPMI administrator on the server from which you run it.
- This command fails if another server cannot access the local IPMI at the supplied address.

Example

To modify the IPMI administrator `scott`:

```
$ crsctl set css ipmiadmin scott
```

crsctl query css ipmiconfig

Determines whether Oracle Clusterware on the local server has been configured to use IPMI for failure isolation.

Note that this command detects the presence of configuration data, but cannot not validate its correctness.

Syntax

```
crsctl query css ipmiconfig
```

Usage Notes

- This command attempts to locate and access the IPMI configuration stored in the Oracle Cluster Registry (OCR) and should be executed under the account used to install Oracle Clusterware, or an authorization failure may be reported.
- An authorization failure may not result when executed under another account, if the registry contains no IPMI configuration data.

Example

This command returns output similar to the following:

```
CRS-4236: Oracle Clusterware configured to use IPMI
```

Or:

```
CRS-4237: Oracle Clusterware is not fully configured to use IPMI
```

crsctl unset css ipmiconfig

Clears all previously stored IPMI configuration (login credentials and IP address) from the Oracle Local Registry.

Prerequisites

This is appropriate when deconfiguring IPMI in your cluster or if IPMI configuration was previously stored by the wrong user.

Syntax

```
crsctl unset css ipmiconfig
```

Usage Notes

- This command must be run under the user account originally used to configure IPMI or by a privileged user.
- This command only clears the IPMI configuration on the server on which you run it.
- If Oracle Clusterware was able to access and use the configuration data to be deleted by this command, then it will continue to do so until you restart Oracle Clusterware.

Example

To clear the IPMI configuration data from the Oracle Local Registry and restart Oracle Clusterware to prevent further use of IPMI, log in as `root` or a member of the Administrator's group on Windows and run the following commands:

```
# crsctl unset css ipmiconfig
# crsctl stop crs
# crsctl start crs
```

crsctl query css ipmidevice

Determines the presence of the Intelligent Platform Management Interface (IPMI) driver on the local system.

Syntax

```
crsctl query css ipmidevice
```

Usage Notes

- This command performs a pre-check during IPMI installation, and is normally issued only by the installer.
- This command performs a perfunctory check and a success return does not guarantee that the IPMI hardware is fully configured for use.
- There are no special privileges required to run this command.

Example

This command returns output similar to the following:

```
CRS-4231: IPMI device and/or driver found
```

Or:

```
CRS-4218: Unable to access an IPMI device on this system
```


crsctl get css noautorestart

Lists the problems for which automatic restart of Cluster Synchronization Services is disabled.

Syntax

```
crsctl get css noautorestart
```

Usage Notes

- This command has no parameters.
- CRSCTL returns one of three possible values:
 - all: Auto restart is disabled for any problem.
 - interconnect: Auto restart is disabled when an interconnect problem occurs.
 - vf: Auto restart is disabled when a voting files problem occurs.

crsctl set css noautorestart

Stops the automatic restart of Cluster Synchronization Services when a specific problem occurs.

Syntax

```
crsctl set css noautorestart {all | interconnect | vf | none}
```

Usage Notes

Specify one of the following values:

- all: Disables automatic restart when any problem occurs.
- interconnect: Disables automatic restart when an interconnect problem occurs.
- vf: Disables automatic restart when a voting files problem occurs.
- none: Enables Cluster Synchronization Services automatic restart.

crsctl delete css votedisk

Removes a voting file from the Oracle Clusterware configuration.

Syntax

```
crsctl delete css votedisk {voting_disk_GUID [...] | vdisk [...] |  
+diskgroup}
```

Parameters

Table E-56 crsctl delete css votedisk Command Parameters

Parameter	Description
<i>voting_disk_GUID</i> [...]	Specify the file universal identifier (GUID) of the voting file you want to remove. Specify multiple GUIDs in a space-delimited list.
<i>vdisk</i> [...]	Alternatively, specify the path of the voting file you want to remove. Specify multiple voting file paths in a space-delimited list.
<i>+diskgroup</i>	Alternatively, specify the name of an Oracle ASM disk group that contains voting files you want to remove. You can only use parameter when Oracle Clusterware is in exclusive mode.

Usage Notes

- You can specify one or more GUIDs of voting files you want to remove, one or more paths to voting files you want to remove, or the name of an Oracle ASM disk group that contains voting files you want to remove.
- You can obtain the GUIDs of each current voting file by running the `crsctl query css votedisk` command

Example

To remove a voting file using a GUID:

```
$ crsctl delete css votedisk 26f7271ca8b34fd0bfcfdc2031805581e
```

crsctl query css votedisk

Displays the voting files used by Cluster Synchronization Services, the status of the voting files, and the location of the disks, whether they are stored in Oracle ASM or elsewhere.

Syntax

```
crsctl query css votedisk
```

Example

This command returns output similar to the following:

```
## STATE      File Universal Id                      File Name Disk group
--  -----  -
1. ONLINE    296641fd201f4f3fbf3452156d3b5881 (/ocfs2/host09_vd3) []
2. ONLINE    8c4a552bdd9a4fd9bf93e444223146f2 (/netapp/ocrvf/newvd) []
3. ONLINE    8afeee6ae3ed4fe6bfbb556996ca4da5 (/ocfs2/host09_vd1) []
Located 3 voting file(s).
```

crsctl check ctss

Checks the status of the Cluster Time Synchronization services.

Syntax

```
crsctl check ctss
```

Example

This command returns output similar to the following:

```
CRS-4700: The Cluster Time Synchronization Service is in Observer mode.
```

Or:

```
CRS-4701: The Cluster Time Synchronization Service is in Active mode.  
CRS-4702: Offset from the reference node (in msec): 100
```

crsctl discover dhcp

Sends DHCP discover packets on the network at the specified port. If DHCP servers are present on the network, then they respond to the discovery message and the command succeeds.

Syntax

```
crsctl discover dhcp -clientid clientid [-port port] [-subnet subnet]
```

Parameters

Table E-57 crsctl discover dhcp Command Parameters

Parameter	Description
-clientid <i>clientid</i>	Specify the client ID for which you want to attempt discovery. Obtain the client ID by running the crsctl get clientid dhcp command.
-port <i>port</i>	Specify the port to which CRSCTL sends the discovery packets.
-subnet <i>subnet</i>	Optionally, you can specify the subnet on which DHCP discover packets are sent.

Usage Notes

You must run this command as `root`.

Example

This command returns output similar to the following:

```
CRS-10009: DHCP server returned server: 192.168.53.232,
loan address : 192.168.29.221/255.255.252.0, lease time: 43200
```

crsctl get clientid dhcp

Displays the client ID that the Oracle Clusterware agent uses to obtain the IP addresses from the DHCP server for configured cluster resources. The VIP type is required.

Syntax

```
crsctl get clientid dhcp -cluname cluster_name -viptype vip_type
[-vip vip_res_name] [-n node_name]
```

Parameters**Table E-58 crsctl get clientid dhcp Command Parameters**

Parameter	Description
-cluname <i>cluster_name</i>	Specify the name of the cluster where the cluster resources are configured.
-viptype <i>vip_type</i>	Specify the type of the VIP resource for which you want to display client IDs: HOSTVIP, SCANVIP, or APPVIP.
-vip <i>vip_res_name</i>	Specify the name of the VIP resource. This parameter is required if you specify the APPVIP VIP type.
-n <i>node_name</i>	Specify the name of the node for which you want to obtain the client ID. This parameter is required if you specify the HOSTVIP VIP type.

Example

This command returns output similar to the following:

```
CRS-10007: stmjk0462clr-stmjk01-vip
```

crsctl release dhcp

Sends a DHCP lease release request to a specific client ID, and sends release packets on the network to a specific port.

Syntax

```
crsctl release dhcp -clientid clientid [-port port]
```

Parameters

Table E-59 crsctl release dhcp Command Parameters

Parameter	Description
-clientid <i>clientid</i>	Specify the client ID for which you want to attempt release. Obtain the client ID by running the <code>crsctl get clientid</code> command.
-port <i>port</i>	The port to which CRSCTL sends the release packets. If you do not specify a port, then CRSCTL uses the default value 67.

Example

This command returns output similar to the following:

```
CRS-10012: released DHCP server lease for client ID stmjk0462clr-stmjk01-
vip
on port 67
```

crsctl request dhcp

Sends DHCP request packets on the network to the specified port. If the DHCP server has an IP address it can provide, then it responds with the IP address for the client ID.

Syntax

```
crsctl request dhcp -clientid clientid [-port port]
```

Parameters

Table E-60 crsctl request dhcp Command Parameters

Parameter	Description
-clientid <i>clientid</i>	Specify the client ID for which you want to attempt request. Obtain the client ID by running the <code>crsctl get clientid</code> command.
-port <i>port</i>	Optionally, you can specify the port to which CRSCTL sends the request packets. If you do not specify a port, then CRSCTL uses the default value 67.

Example

This command returns output similar to the following:

```
CRS-10009: DHCP server returned server: 192.168.53.232,
loan address : 192.168.29.228/255.255.252.0, lease time: 43200
```

crsctl replace discoverystring

Replaces the existing discovery string used to locate voting files.

Syntax

```
crsctl replace discoverystring "absolute_path[,...]"
```

Usage Notes

- Specify one or more comma-delimited absolute paths that match one or more voting file locations. You can use wild cards and a list of paths must be enclosed in double quotation marks ("").
- You must be `root`, the Oracle Clusterware installation owner, or a member of the Administrators group to run this command.
- You can run this command on any node in the cluster.
- If you store voting files in an Oracle ASM disk group, then you cannot change the discovery string.

Example

Assume the current discovery string is `/oracle/css1/*`. To also use voting files in the `/oracle/css2/` directory, replace the current discovery string using the following command:

```
# crsctl replace discoverystring "/oracle/css1/*,/oracle/css2/*"
```

crsctl query dns

Obtains a list of addresses returned by DNS lookup of the name with the specified DNS server.

Syntax

```
crsctl query dns -name name [-dnsserver DNS_server_address] [-type query_type]  
[-port port] [-attempts number_of_attempts] [-timeout timeout_in_seconds]  
[-v]
```

Parameters

Table E-61 crsctl query dns Command Parameters

Parameter	Description
-name <i>name</i>	Specify the fully-qualified domain name that you want to query.
-dnsserver <i>DNS_server_address</i>	Optionally, you can specify the address of the DNS server on which you want the domain name to be queried.
-type <i>query_type</i>	Optionally, you can select either IPv4 or IPv6 by specifying either A or AAAA, respectively, as a query type.
-port <i>port</i>	Optionally, you can specify the port on which the DNS server listens. If you do not specify a port, then it defaults to port 53.

Table E-61 (Cont.) crsctl query dns Command Parameters

Parameter	Description
-attempts <i>number_of_attempts</i>	Optionally, you can specify the number of retry attempts.
-timeout <i>timeout_in_seconds</i>	Optionally, you can specify the timeout length in seconds.
-v	Verbose output.

Example

The `crsctl query dns` command returns output similar to the following for a DNS server named `stmjk07-vip.stmjk0462.foo.com`:

```
CRS-10024: DNS server returned 192.168.29.250 for name
stmjk07-vip.stmjk0462.foo.com
```

crsctl start ip

Starts a given IP name or IP address on a specified interface with a specified subnet mask.

Run this command on the server on which you want to start the IP.

Syntax

```
crsctl start ip -A {IP_name | IP_address}/netmask/interface_name
```

Parameters

Table E-62 crsctl start ip Command Parameters

Parameter	Description
<i>IP_name</i> <i>IP_address</i>	Specify either a domain name or an IP address preceded by -A. If you do not specify a fully-qualified domain name, then CRSCTL uses a standard name search.
<i>/netmask/ interface_name</i>	Specify a subnet mask and an interface name for the IP to start, separated by a forward slash (/). This pair must also be preceded by a forward slash.

Example

To start an IP on the local server, run the command similar to the following:

```
$ crsctl start ip -A 192.168.29.220/255.255.252.0/eth0
```

crsctl status ip

Checks if a given IP address is up on the network.

Syntax

```
crsctl status ip -A {IP_name | IP_address}
```

Usage Notes

Specify either a domain name or an IP address preceded by `-A`. If you do not specify a fully-qualified domain name, then CRSCTL uses a standard name search.

Example

This command returns output similar to the following:

```
CRS-10003: IP address 192.168.29.220 could be reached from current node
```

crsctl stop ip

Stops a given IP name or IP address on a specified interface with a specified subnet mask.

Run this command on the server on which you want to stop the IP.

Syntax

```
crsctl stop ip -A {IP_name | IP_address}/interface_name
```

Parameters**Table E-63 crsctl stop ip Command Parameters**

Parameter	Description
<i>IP_name</i> <i>IP_address</i>	Specify either a domain name or an IP address preceded by <code>-A</code> . If you do not specify a fully-qualified domain name, then CRSCTL uses a standard name search.
<i>interface_name</i>	Specify an interface on which to stop the IP.

Example

To stop an IP on the local server, run the command similar to the following:

```
$ crsctl stop ip -A MyIP.domain.com/eth0
```

crsctl lsmodules

Lists the components of the modules that you can debug.

Syntax

```
crsctl lsmodules {mdns | gnpn | css | crf | crs | ctss | evm | gipc}
```


Usage Notes

You can obtain a list of components for any of the following modules:

- mdns: Multicast domain name server
- gppp: Grid Plug and Play service
- css: Cluster Synchronization Services
- crf: Cluster Health Monitor
- crs: Cluster Ready Services
- ctss: Cluster Time Synchronization Service
- evm: Event Manager
- gipc: Grid Interprocess Communication

Example

This command returns output similar to the following:

```
$ crsctl lsmodules evm
List EVMD Debug Module: CLSVER
List EVMD Debug Module: CLUCLS
List EVMD Debug Module: COMMCRS
List EVMD Debug Module: COMMNS
List EVMD Debug Module: CRSOCR
List EVMD Debug Module: CSSCLNT
List EVMD Debug Module: EVMAGENT
List EVMD Debug Module: EVMAPP
...
```

Related Topics

- [Dynamic Debugging Using crsctl set log](#)

crsctl create member_cluster_configuration

Creates the configuration deployment file for a new member cluster.

Syntax

```
crsctl create member_cluster_configuration member_cluster_name
  -file cluster_manifest_file -member_type database | application
  [-version member_cluster_version] [-domain_services [asm_storage local
  | direct | indirect] [rhp] [acfs]
```

Parameters

Table E-64 crsctl create member_cluster_configuration Command Parameters

Parameter	Description
<i>member_cluster_name</i>	Specify the name of the new member cluster.

Table E-64 (Cont.) crsctl create member_cluster_configuration Command Parameters

Parameter	Description
-file <i>cluster_manifest_file</i>	Specify the path of the cluster manifest file (including the .xml extension) you want to create.
-member_type database application	Specify the type of member cluster you want to create. If you choose <i>database</i> , then you can use the cluster to deploy Oracle databases. If you choose <i>application</i> , then you can use the cluster to only deploy applications.
-version <i>member_cluster_version</i>	Optionally, you can specify the five-digit version of Oracle Grid Infrastructure (such as 12.2.0.1.0) on the new member cluster, if it is different from that of the domain services cluster.
-domain_services [asm_storage local direct indirect] [rhp] [acfs]	<p>Optionally, you can specify services to be initially configured for this member cluster. If you do not use the <i>-domain_services</i> parameter, then CRSCTL only configures the Grid Infrastructure Management Repository and TFA services.</p> <p>Additionally, you can optionally specify the Oracle ASM storage access path for the database member clusters.</p> <p>local: Storage is local to the cluster</p> <p>direct or indirect: Choose direct or indirect access to storage provided on the domain services cluster</p> <p>Optionally, you can choose <i>rhp</i> to generate credentials and configuration for a Fleet Patching and Provisioning Client cluster.</p> <p>Optionally, you can choose <i>acfs</i> for an indirect-access Member Cluster to utilize Oracle ACFS Remote to provide storage to the Member Cluster.</p>

crsctl delete member_cluster_configuration

Deletes a member cluster configuration.

Syntax

```
crsctl delete member_cluster_configuration member_cluster_name [-f]
```

Usage Notes

You must specify the name of the member cluster you want to delete and you can optionally use *-f* to force the delete operation.

crsctl query member_cluster_configuration

Displays the configuration for a member cluster.

Syntax

```
crsctl query member_cluster_configuration [member_cluster_name]
```

Usage Notes

If you choose not to specify a particular member cluster by name, then CRSCTL displays the details of all the member clusters.

Example

This command returns output similar to the following:

```
$ crsctl query member_cluster_configuration

indirect-mc0      12.2.0.1.0 66cdb4efd2f77f7affe559cc56003a14 ASM,GIMR,TFA
indirect-mc1      12.2.0.1.0 2b85c6ac63434f30bfa6d074bb337621 ASM,GIMR,TFA
indirect-mc2      12.2.0.1.0 d883e65af9ffff1fbf67e922e99bd6e4 ASM,GIMR,TFA
indirect-mc3      12.2.0.1.0 60dcca80166aff0cffaf8e33213bfacb ASM,GIMR,TFA
indirect-mc4      12.2.0.1.0 78d64e1454505f3effda0136738f325e ASM,GIMR,TFA
```

crsctl delete node

Removes a node from the cluster.

Syntax

```
crsctl delete node -n node_name
```

Usage Notes

- You must be `root` to run this command
- The `node_name` parameter is required

Example

To delete the node named `node06` from the cluster, run the following command as `root`:

```
# crsctl delete node -n node06
```

crsctl get nodename

Obtains the name of the local node.

Syntax

```
crsctl get nodename
```

Example

This command returns output similar to the following:

```
node2
```

crsctl get node role

Obtains the configured node role of nodes in the cluster.

Syntax

```
crsctl get node role {config | status} [node node_name | -all]
```

Usage Notes

- Specify the `config` option to obtain the configured node role for a specific node.
- Specify the `status` option to obtain the current status of a specific node.
- You can specify a particular node for which to obtain role information or you can use `-all` to obtain information for all nodes in the cluster. If you do not specify a particular node, then CRSCTL returns information about the local node.

Examples

This command returns output similar to the following:

```
$ crsctl get node role config -all
Node 'mjkeenan_node5_0' configured role is 'hub'
Node 'mjkeenan_node5_1' configured role is 'hub'
Node 'mjkeenan_node5_2' configured role is 'hub'
Node 'mjkeenan_node5_3' configured role is 'hub'

$ crsctl get node role status -all
Node 'mjkeenan_node5_0' active role is 'hub'
Node 'mjkeenan_node5_2' active role is 'hub'
Node 'mjkeenan_node5_3' active role is 'hub'

$ crsctl get node role config -node mjkeenan_node5_1
Node 'mjkeenan_node5_1' configured role is 'hub'
```

crsctl add policy

Adds a configuration policy to the policy set.

Syntax

```
crsctl add policy policy_name -attr
"attr_name=attr_value[,attr_name=attr_value[, ...]]"
```

Parameters

Table E-65 crsctl add policy Command Parameters

Parameter	Description
<i>policy_name</i>	Specify a name for the policy you want to add.

Table E-65 (Cont.) crsctl add policy Command Parameters

Parameter	Description
<i>attr_name</i>	Specify a description for the policy using the DESCRIPTION policy attribute preceded by the <code>-attr</code> flag.
<i>attr_value</i>	A value for the DESCRIPTION policy attribute that describes the policy. Note: The <i>attr_name</i> and <i>attr_value</i> parameters must be enclosed in double quotation marks (") and separated by commas. For example: <code>-attr "DESCRIPTION=daytime"</code>

Usage Notes

- Adding a policy does not activate the policy
- The *policy_name* parameter is required
- Privileges necessary to run this command depend on the value of the ACL attribute of the policy set

Example

To add a policy:

```
$ crsctl add policy nightTime -attr "DESCRIPTION=nighttime"
```

crsctl delete policy

Deletes a configuration policy from the policy set.

Syntax

```
crsctl delete policy policy_name [policy_name [...]] [-i]
```

Usage Notes

- Specify a name for the policy or a space-delimited list of policy names you want to delete. The *policy_name* parameter is required.
- If you specify `-i`, then the command fails if Oracle Clusterware cannot process the request immediately.
- Privileges necessary to run this command depend on the value of the ACL attribute of the policy set.

Example

To delete a policy, run the following command as `root` or the Oracle Clusterware installation owner:

```
# crsctl delete policy nightTime
```

crsctl eval activate policy

Predicts the effects of activating a specific policy without making changes to the system.

This command may be useful to cluster administrators.

Syntax

```
crsctl eval activate policy policy_name [-f] [-admin [-l serverpools |
resources
| all] [-x] [-a]] [-explain]
```

Parameters

Table E-66 crsctl eval activate policy Command Paramaters

Parameter	Description
-f	Specify this parameter to evaluate what happens if you try to forcibly activate a policy.
-admin [-l serverpools resources all] [-x] [-a]	<p>You must specify <code>-admin</code> if you specify any combination of <code>-l</code>, <code>-x</code>, and <code>-a</code>.</p> <p>If you specify the <code>-l</code> parameter, then you can choose one of the following three output levels:</p> <ul style="list-style-type: none"> <code>serverpools</code>: Restricts the output to servers running in a server pool <code>resources</code>: Restricts the output to resources running on servers in a server pool <code>all</code>: Displays all available output <p>If you specify the <code>-x</code> parameter, then CRSCTL displays the differences.</p> <p>If you specify the <code>-a</code> parameter, then CRSCTL displays all resources.</p>
-explain	Specify this parameter to display reasoned command evaluation information.

Example

This command returns output similar to the following:

```
$ crsctl eval activate policy p2 -explain
```

```
Stage Group 1:
```

```
-----
```

Stage	Required	Action
1	E	Starting to evaluate activation of policy 'p2' with server pools (MIN_SIZE, MAX_SIZE) 'Free(0,-1),sp1(2,-1),sp2(4,-1)'.
	E	Looking at other server pools to see whether MIN_SIZE value 4 of server pool 'sp2' can be met.
	E	Scanning server pools with MIN_SIZE or fewer servers in ascending order of IMPORTANCE.
	E	Considering server pools (IMPORTANCE): sp1(0) for suitable servers.
	E	Considering server pool 'sp1' because its MIN_SIZE is 2 and it has 0 servers above MIN_SIZE.
	E	Relocating server 'mjkeenan_nodel_0' to server pool 'sp2'.
	E	Scanning server pools with MIN_SIZE or fewer servers in ascending order of IMPORTANCE.
	E	Considering server pools (IMPORTANCE): sp1(0) for suitable servers.
	E	Considering server pool 'sp1' because its MIN_SIZE is 2 and it has 0 servers above MIN_SIZE.
	E	Relocating server 'mjkeenan_nodel_1' to server pool 'sp2'.
2	Y	Resource 'cs2' (3/1) will be in state 'ONLINE INTERMEDIATE' on server 'mjkeenan_nodel_0'.
	Y	Resource 'cs2' (4/1) will be in state 'ONLINE INTERMEDIATE' on server 'mjkeenan_nodel_1'.
3	Y	Server 'mjkeenan_nodel_0' will be moved from pool 'sp1' to pool 'sp2'.
	Y	Server 'csivanan_nodel_1' will be moved from pool 'sp1' to pool 'sp2'.

Related Topics

- [How Server Pools Work](#)

crsctl modify policy

Modifies an existing configuration policy.

Syntax

```
crsctl modify policy policy_name -attr "attr_name=attr_value"
```

Parameters

Table E-67 crsctl modify policy Command Parameters

Parameter	Description
<i>policy_name</i>	Specify the name of the policy you want to modify.
<i>attr_name</i>	Specify a description for the policy using the DESCRIPTION policy attribute following the -attr flag.

Table E-67 (Cont.) crsctl modify policy Command Parameters

Parameter	Description
<code>attr_value</code>	<p>A value for the DESCRIPTION policy attribute that describes the policy.</p> <p>Note: The <code>attr_name</code> and <code>attr_value</code> parameters must be enclosed in double quotation marks (" ") and separated by commas. For example:</p> <pre>-attr "DESCRIPTION=daytime"</pre>

Usage Notes

- The `policy_name` parameter is required
- Privileges necessary to run this command depend on the value of the ACL attribute of the policy set

Example

To modify an existing policy, run the following command as root or the Oracle Clusterware installation owner:

```
# crsctl modify policy p1 -attr "DESCRIPTION=daytime"
```

crsctl status policy

Obtains the status and definition of a configuration policy.

Syntax

```
crsctl status policy [policy_name [...]] | -w "filter" | -active]
```

Parameters

Table E-68 crsctl status policy Command Parameters

Parameter	Description
<code>policy_name [...]</code>	Specify the name of the policy or a space-delimited list of policy names for which you want to view the status.
<code>-w "<i>filter</i>"</code>	<p>Alternatively, you can specify a policy filter preceded by the <code>-w</code> flag.</p> <p>See Also: "Filters" for more information</p>
<code>-active</code>	Alternatively, you can specify this parameter to display the status of the active policy.

Usage Notes

Privileges necessary to run this command depend on the value of the ACL attribute of the policy set.

crsctl create policyset

Creates a single policy set, in the form of a text file, that reflects the server pool configuration. After you create a policy set, you can copy the contents of the text file to create other policy sets.

Syntax

```
crsctl create policyset -file path_to_file
```

Usage Notes

Specify a path to where CRSCTL creates a file that you can edit and then send back using [crsctl modify policyset](#) to add, delete, or update policies.

Example

To create a policy set:

```
$ crsctl create policyset -file /tmp/ps
```

crsctl modify policyset

Modifies an existing policy set.

Syntax

```
crsctl modify policyset {-attr  
"attr_name=attr_value[,attr_name=attr_value[, ...]]"  
| -file file_name} [-ksp] [-f]
```

Parameters

Table E-69 crsctl modify policyset Command Parameters

Parameter	Description
<i>attr_name</i>	The name of a policy attribute you want to modify following the - <i>attr</i> flag. With this command, you can specify any of the following attributes: ACL LAST_ACTIVATED_POLICY SERVER_POOL_NAMES

Table E-69 (Cont.) crsctl modify policyset Command Parameters

Parameter	Description
<code>attr_value</code>	<p>A value for the policy attribute.</p> <p>Note: The <code>attr_name</code> and <code>attr_value</code> parameters must be enclosed in double quotation marks (" ") and separated by commas. For example:</p> <pre>-attr "ACL='owner:mjkeenan:rxw,pgrp:svrtech:rxw,other::r--', SERVER_POOL_NAMES=sp1 sp2 Free"</pre>
<code>-file file_name</code>	If you specify this parameter instead of <code>-attr</code> , then enter a name of a file that contains policy set definitions.
<code>-ksp</code>	If you specify this parameter, then CRSCTL keeps the server pools in the system, which means that they are independent and not managed by the policy set.
<code>-f</code>	Specify this parameter to force enable policy set modifications.

Usage Notes

- Privileges necessary to run this command depend on the value of the `ACL` attribute of the policy set
- You can only specify policy definitions using the `-file` parameter or by running the [crsctl modify policy](#) command

Example

To modify an existing policy set, run the following command as `root` or the Oracle Clusterware installation owner:

```
# crsctl modify policyset -file my_policy_set.def
```

crsctl status policyset

Obtains the current policies in the policy set, including the access control list, which governs who can modify the set, the last activated policy, and the configuration which is now in effect, which is known as the Current policy.

Syntax

```
crsctl status policyset [-file file_name]
```

Usage Notes

- You can specify this parameter to create a file that you can edit and then send back using [crsctl modify policyset](#) to add, delete, or update multiple policies. If you do not specify this optional parameter, then CRSCTL displays the Current configuration.

- Privileges necessary to run this command depend on the value of the `ACL` attribute of the policy set.

Example

This command returns output similar to the following:

```

ACL=owner:'mjkeenan:rwx,pgrp:g900:rwx,other::r--'
LAST_ACTIVATED_POLICY=DayTime
SERVER_POOL_NAMES=Free pool1 pool2 pool3
POLICY
NAME=DayTime
DESCRIPTION=Test policy
SERVERPOOL
  NAME=pool1
  IMPORTANCE=0
  MAX_SIZE=2
  MIN_SIZE=2
  SERVER_CATEGORY=
  SERVER_NAMES=
SERVERPOOL
  NAME=pool2
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
POLICY
NAME=NightTime
DESCRIPTION=Test policy
SERVERPOOL
  NAME=pool1
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool2
  IMPORTANCE=0
  MAX_SIZE=2
  MIN_SIZE=2
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
POLICY
NAME=Weekend

```

```
DESCRIPTION=Test policy
SERVERPOOL
  NAME=pool1
  IMPORTANCE=0
  MAX_SIZE=0
  MIN_SIZE=0
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool2
  IMPORTANCE=0
  MAX_SIZE=1
  MIN_SIZE=1
  SERVER_CATEGORY=
SERVERPOOL
  NAME=pool3
  IMPORTANCE=0
  MAX_SIZE=3
  MIN_SIZE=3
  SERVER_CATEGORY=
```

crsctl check resource

Initiates the check action inside the application-specific agent of a particular resource. Oracle Clusterware only provides output if something prevents the system from issuing the check request, such as a bad resource name.

Syntax

```
crsctl check resource {resource_name [...] | -w "filter"} [-n node_name]
  [-k cardinality_id] [-d degree_id]
```

Parameters

Table E-70 crsctl check resource Command Parameters

Parameter	Description
<i>resource_name</i> [...]	Specify a particular resource. You can check multiple resources by entering a space-delimited list of resource names.
-w " <i>filter</i> "	Specify a resource filter that Oracle Clusterware uses to limit the number of resources checked. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include: <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to check only resources of <i>cluster_resource</i> type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to check resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to check resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute <i>and</i> the name of the resource contains the number 2 <p>See Also: "Filters" for more information</p>

Table E-70 (Cont.) crsctl check resource Command Parameters

Parameter	Description
-n <i>node_name</i>	Check the resource instance on a specific node. If you do not specify the -n parameter, then Oracle Clusterware checks the resource instances only on the local server.
-k <i>cardinality_id</i>	Specify the resource cardinality ID.
-d <i>degree_id</i>	Specify the resource degree ID.

Usage Notes

- You must have read and execute permissions on the specified resources to check them.
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources.
- If this command is successful, it only means that a check was issued; it does not mean the CHECK action was completed.

Example

To initiate the check action inside the application-specific agent of a particular resource:

```
$ crsctl check resource appsvip
```

crsctl relocate resource

Relocates resources to another server in the cluster.

Syntax

```
crsctl relocate resource {resource_name [-k cid] | {resource_name | -all}
    -s source_server | -w "filter"} [-n destination_server] [-env
    "env1=val1,env2=val2,..."]
    [-i] [-f]
```

Parameters

Table E-71 crsctl relocate resource Command Parameters

Parameter	Description
<i>resource_name</i> [-k <i>cid</i>]	Specify the name of a resource you want to relocate. Optionally, you can also specify the resource cardinality ID. If you specify this parameter, then Oracle Clusterware relocates the resource instance that have the cardinality you specify.
<i>resource_name</i> -all -s <i>source_server</i>	Alternatively, you can specify one particular resource or all resources located on a particular server that you specify <i>from</i> which you want to relocate those resources.

Table E-71 (Cont.) crsctl relocate resource Command Parameters

Parameter	Description
-w " <i>filter</i> "	<p>Alternatively, you can specify a resource filter that Oracle Clusterware uses to limit the number of resources relocated. The filter must be enclosed in double quotation marks (" "). Examples of resource filters include:</p> <ul style="list-style-type: none"> "TYPE == cluster_resource": This filter limits Oracle Clusterware to relocate only resources of cluster_resource type "CHECK_INTERVAL > 10": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute "(CHECK_INTERVAL > 10) AND (NAME co 2)": This filter limits Oracle Clusterware to relocate resources that have a value greater than 10 for the CHECK_INTERVAL resource attribute <i>and</i> the name of the resource contains the number 2 <p>See Also: "Filters" for more information</p>
-n <i>destination_server</i>	Specify the name of the server to which you want relocate resources. If you do not specify a destination server, then Oracle Clusterware relocates the resources to the best server according to the attribute profile of each resource.
-env " <i>env1=val1,env2=val2</i> ,..."	You can optionally override one or more resource profile attribute values for this command. If you specify multiple environment name-value pairs, then you must separate each pair with a comma and enclose the entire list in double quotation marks (" ").
-i	If you specify -i, then the command returns an error if processing this command requires waiting for Oracle Clusterware to unlock the resource or its dependents. Sometimes, Oracle Clusterware locks resources or other objects to prevent commands from interfering with each other.
-f	Specify the -f parameter to force the relocating of the resource when it has other resources running that depend on it. Dependent resources are relocated or stopped when you use this parameter. Note: When you are relocating resources that have cardinality greater than 1, you must use either -k or -s to narrow down which resource instances are to be relocated.

Usage Notes

- Any one of the three following options is required to specify which resources you want to relocate:
 - You can specify one particular resource to relocate.
 - Or you can specify one particular or all the resources to relocate from a particular source server.
 - Thirdly, you can specify a resource filter that Oracle Clusterware uses to match resources to relocate.
- If a resource has a degree ID greater than 1, then Oracle Clusterware relocates all instances of the resource.

- You must have read and execute permissions on the specified resources to relocate them.
- Do not use this command for any resources with names that begin with *ora* because these resources are Oracle resources.

Example

To relocate one particular resource from one server to another:

```
# crsctl relocate resource myResource1 -s node1 -n node3
```

crsctl get resource use

Checks the current setting value of the `RESOURCE_USE_ENABLED` server configuration parameter.

Syntax

```
crsctl get resource use
```

Usage Notes

The possible values are 1 or 0. If the value for this attribute is 1, which is the default, then the server can be used for resource placement. If the value is 0, then Oracle Clusterware disallows starting server pool resources on the server. The server remains in the Free server pool.

Example

This command returns output similar to the following:

```
CRS-4966: Current resource use parameter value is 1
```

crsctl set resource use

Sets the value of the `RESOURCE_USE_ENABLED` server configuration parameter for the server on which you run this command.

Syntax

```
crsctl set resource use [1 | 0]
```

Usage Notes

- The possible values are 1 or 0. If you set the value for this attribute to 1, which is the default, then the server can be used for resource placement. If you set the value to 0, then Oracle Clusterware disallows starting server pool resources on the server. The server remains in the Free server pool.
- You must run this command as `root` or a cluster administrator, or an administrator on Windows systems.

Example

To set the value of the `RESOURCE_USE_ENABLED` server configuration parameter:

```
# crsctl set resource use 1
```

crsctl start rollingpatch

Transitions Oracle Clusterware and Oracle ASM into rolling patch mode. In this mode, the software tolerates nodes having different patch levels.

Syntax

```
crsctl start rollingpatch
```

Usage Notes

- This command queries the Oracle Clusterware rolling patch state and Oracle ASM cluster state. If either one is not in rolling patch mode, then it will use the appropriate method to transition Oracle Clusterware or Oracle ASM to rolling patch mode.
- If Oracle Clusterware and Oracle ASM are both in rolling patch mode when you run this command, then this command does nothing.
- The rolling patch mode is not persistent. If all the nodes in a cluster are shut down and restarted, then the cluster transitions out of rolling patch mode when it is restarted. Similarly, if Oracle Clusterware is stopped and then restarted on all nodes in the cluster, then the rolling patch mode is lost.
- This command does not transition Oracle ASM to rolling patch mode if issued within an Oracle ASM Client Cluster.

crsctl stop rollingpatch

Transitions Oracle Clusterware and Oracle ASM out of rolling patch mode. Once the software transitions out of rolling patch mode, it does not tolerate nodes having different patch levels.

Syntax

```
crsctl stop rollingpatch
```

Usage Notes

- This command queries the Oracle Clusterware rolling patch state and Oracle ASM cluster state. If either one is in rolling patch mode, then it will use the appropriate method to transition Oracle Clusterware or Oracle ASM out of rolling patch mode.
- This command verifies that all the nodes in the cluster have a consistent patch level, and returns an error otherwise.
- If neither Oracle Clusterware or Oracle ASM are in rolling patch mode when you issue this command, then this command does nothing.

- This command does not transition Oracle ASM out of rolling patch mode if issued within an Oracle ASM Client Cluster.

crsctl start rollingupgrade

Transitions Oracle Clusterware and Oracle ASM into rolling upgrade mode.

Syntax

```
crsctl start rollingupgrade version
```

Usage Notes

- This command queries the Oracle Clusterware rolling upgrade state and Oracle ASM cluster state. If either one is not in rolling upgrade mode, then it will use the appropriate method to transition Oracle Clusterware or Oracle ASM to rolling upgrade mode.
- If Oracle Clusterware and Oracle ASM are both in rolling upgrade mode when you run this command, then this command does nothing.
- The rolling upgrade mode is not persistent. If all the nodes in a cluster are shut down and restarted, then the cluster transitions out of rolling upgrade mode when it is restarted. Similarly, if Oracle Clusterware is stopped and then restarted on all nodes in the cluster, then the rolling upgrade mode is lost.
- This command does not transition Oracle ASM to rolling upgrade mode if run within an Oracle ASM Client Cluster.

crsctl eval add server

Simulates the addition of a server without making changes to the system.

This command may be useful to cluster administrators.

Syntax

```
crsctl eval add server server_name [-file file_path] | [-attr  
"attr_name=attr_value[,...]" ]  
[-admin [-l level [-x] [-a]] [-explain]
```

Parameters

Table E-72 crsctl eval add server Command Parameters

Parameter	Description
<i>server_name</i>	The name of the server you want to add.
-file <i>file_path</i>	Fully-qualified path to a file containing server attributes.

Table E-72 (Cont.) crsctl eval add server Command Parameters

Parameter	Description
-attr "attr_name=attr_value[,...]"	<p>Alternatively, you can specify the name of a server attribute you want to modify preceded by the <code>-attr</code> flag, followed by a value for the server attribute.</p> <p>Note: The <code>attr_name=attr_value</code> name-value pairs must be enclosed in double quotation marks (" "). Multiple name-value pairs must be separated by commas. For example:</p> <pre>-attr "MAX_SIZE=30,IMPORTANCE=25"</pre>
-admin [-l level] [-x] [-a]	<p>If you specify this parameter, then CRSCTL displays output for the cluster administrator.</p> <p>If you specify the <code>-l</code> parameter, then you can choose one of the following three output levels:</p> <ul style="list-style-type: none"> <code>serverpools</code>: Restricts the output to servers running in a server pool <code>resources</code>: Restricts the output to resources running on servers in a server pool <code>all</code>: Displays all available output <p>If you specify the <code>-x</code> parameter, then CRSCTL displays the differences.</p> <p>If you specify the <code>-a</code> parameter, then CRSCTL displays all resources.</p> <p>Note: To specify either the <code>-l</code>, <code>-x</code>, or <code>-a</code> parameters, or any combination of the three, you must specify the <code>-admin</code> parameter.</p>
-explain	Specify this parameter to display reasoned command evaluation information.

Example

This command returns output similar to the following:

```
$ crsctl eval add server scott_node5_0
```

```
Stage Group 1:
-----
Stage   Required   Action
-----
1       E   Looking for a suitable server pool for server 'scott_node5_0'.
        E   Scanning server pools with MIN_SIZE or more servers in
        E   descending order of IMPORTANCE.
        E   Considering server pools (IMPORTANCE): Free(0) for relocating
        E   servers.
        E   Relocating server 'scott_node5_0' to server pool 'Free'.
        Y   Server 'scott_node5_0' will be added to pools 'Free'.
-----
```

Related Topics

- [Server Category Attributes](#)
You define servers into named categories, and assign attributes that define servers as members of that category.

crsctl eval delete server

Predicts the effects of deleting a server without making changes to the system.

This command may be useful to cluster administrators.

Syntax

```
crsctl eval delete server server_name [-admin [-l level] [-x] [-a]] [-f] [-explain]
```

Parameters

Table E-73 crsctl eval delete server Command Parameters

Parameter	Description
<i>server_name</i>	Specify the name of the server you want to evaluate before deleting.
-admin [-l <i>level</i>] [-x] [-a]	<p>If you specify this parameter, then CRSCTL displays output for the cluster administrator.</p> <p>If you specify the -l parameter, then you can choose one of the following three output levels:</p> <ul style="list-style-type: none"> • <i>serverpools</i>: Restricts the output to servers running in a server pool • <i>resources</i>: Restricts the output to resources running on servers in a server pool • <i>all</i>: Displays all available output <p>If you specify the -x parameter, then CRSCTL displays the differences.</p> <p>If you specify the -a parameter, then CRSCTL displays all resources.</p> <p>Note: To specify either the -l, -x, or -a parameters, or any combination of the three, you must specify the -admin parameter.</p>
-explain	Specify this parameter to display reasoned command evaluation information.

Examples

This command returns output similar to the following:

```
$ crsctl eval delete server scott_node1_0 -explain
```

```
Stage Group 1:
```

```
-----  
Stage Number      Required          Action  
-----
```

```

1      Y  Resource 'cs1' (1/1) will be in state 'OFFLINE'.
      E  Server 'scott_nodel_0' is removed from server pool 'sp1'.
      E  Server pool 'sp1' is below the MIN_SIZE value of 2 with 1
      servers.
      E  Looking at other server pools to see whether MIN_SIZE value 2 of
      server pool 'sp1' can be met.
      E  Scanning server pools with more than MIN_SIZE servers in
      ascending order of IMPORTANCE.
      E  Considering server pools (IMPORTANCE): sp2(4) for suitable
      servers.
      E  Considering server pool 'sp2' because its MIN_SIZE is 1 and it
      has 1 servers above MIN_SIZE.
      E  Relocating server 'scott_nodel_2' to server pool 'sp1'.
      Y  Server 'scott_nodel_2' will be moved from pool 'sp2' to pool
      'sp1'.
      Y  Server 'scott_nodel_0' will be removed from pools 'sp1'.
      Y  Resource 'cs2' (1/1) will be in state 'OFFLINE'.
2      Y  Resource 'cs1' (1/1) will be in state 'ONLINE|INTERMEDIATE' on
      server 'scott_nodel_2'.
-----

```

Related Topics

- [How Server Pools Work](#)

crsctl eval relocate server

Predicts the effects of relocating a server to a different server pool without making changes to the system.

This command might be useful for a cluster administrator.

Syntax

```
crsctl eval relocate server server_name -to server_pool_name [-f]
      [-admin [-l level] [-x] [-a]]
```

Parameters

Table E-74 crsctl eval relocate server Parameters

Parameter	Description
<i>server_name</i>	The name of the server you want to relocate. You can provide a space-delimited list of servers to evaluate relocating multiple servers.
-to	Specify the name of the server pool to which you want relocate the server.
-f	If you specify this parameter, then CRSTL predicts the effects of forcibly relocating a server.

Table E-74 (Cont.) crsctl eval relocate server Parameters

Parameter	Description
-admin [-l level] [-x] [-a]	<p>If you specify this parameter, then CRSCTL displays output for the cluster administrator.</p> <p>If you specify the -l parameter, then you can choose one of the following three output levels:</p> <ul style="list-style-type: none"> serverpools: Restricts the output to servers running in a server pool resources: Restricts the output to resources running on servers in a server pool all: Displays all available output <p>If you specify the -x parameter, then CRSCTL displays the differences.</p> <p>If you specify the -a parameter, then CRSCTL displays all resources.</p> <p>Note: To specify either the -l, -x, or -a parameters, or any combination of the three, you must specify the -admin parameter.</p>

Example

This command returns output similar to the following:

```
$ crsctl eval relocate server scott-node4-0 -to sp2

Stage Group 1:
-----
----
Stage   Required Action
-----
-----
1       Y Server 'scott-node4-0' will be moved from pool 'sp1' to pool
'sp2'.
2       Y Resource 'cs2' (2/1) will be in state 'ONLINE|INTERMEDIATE' on
server 'scott-node4-0'.
        Y Resource 'cs1' (2/1) will be in state 'ONLINE|INTERMEDIATE' on
server 'scott-node4-0'.
-----
-----
```

crsctl modify server

Modifies the configuration of a particular server.

Syntax

```
crsctl modify server server_name -attr "attr_name=attr_value[,...]"
```

Parameters

Table E-75 crsctl modify server Command Parameters

Parameter	Description
<i>server_name</i>	Specify the name of the server you want to modify.
-attr "attr_name=attr_value"	Specify the name of a server attribute you want to modify preceded by the -attr flag, followed by a value for the server attribute. Note: The <i>attr_name=attr_value</i> name-value pairs must be enclosed in double quotation marks (" "). Multiple name-value pairs must be separated by commas. For example: -attr "CHECK_INTERVAL=30,START_TIMEOUT=25"

Usage Notes

If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' '). For example:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```

Example

To modify the configuration of a specific server, enter a command similar to the following:

```
$ crsctl modify server email_server_2 -attr "WORKLOAD_VALUE=1"
```

crsctl relocate server

Relocates a server to a different server pool.

Syntax

```
crsctl relocate server server_name [...] -c server_pool_name [-i] [-f]
```

Parameters

Table E-76 crsctl relocate server Command Parameters

Parameter	Description
<i>server_name</i> [...]	Specify the name of the server you want to relocate. You can provide a space-delimited list of servers to relocate multiple servers.
-c <i>server_pool_name</i>	Specify the name of the server pool to which you want relocate the servers.

Table E-76 (Cont.) crsctl relocate server Command Parameters

Parameter	Description
-i	If you specify -i, then the command fails if Oracle Clusterware cannot process the request immediately.
-f	<p>If you specify the -f parameter, then Oracle Clusterware stops resources running on the servers in another server pool and relocates that server into the server pool you specified.</p> <p>If you do not specify the -f parameter, then Oracle Clusterware checks for resources that must be stopped on the servers that are being relocated. If it finds any, then Oracle Clusterware rejects the crsctl relocate server command.</p> <p>Note: If the number of servers in the server pool is not above the value of the MIN_SIZE server pool attribute, then the force parameter will have no affect because CRSCTL will not violate the configuration.</p>

Usage Notes

The *server_name* and -c *server_pool_name* parameters are required.

Example

To move the *node6* and *node7* servers into the *sp1* server pool without disrupting any active resources on those nodes, use the following command:

```
$ crsctl relocate server node6 node7 -c sp1
```

crsctl status server

Obtains the status and configuration information of one or more particular servers.

Syntax

```
crsctl status server {server_name [...] | -w "filter"} [-g | -p | -v | -f]
|
  [-category category_name | -w "filter"]
```

Parameters

Table E-77 crsctl status server Command Parameters

Parameter	Description
<i>server_name</i> [...]	Specify one or more space-delimited server names.
-w " <i>filter</i> "	<p>Alternatively, specify a filter to determine which servers are displayed. The filter must be enclosed in double quotation marks (" "). Values that contain parentheses or spaces must be enclosed in single quotation marks (' '). For example, "STATE = ONLINE" limits the display to servers that are online.</p> <p>See Also: "Filters" for more information</p>

Table E-77 (Cont.) crsctl status server Command Parameters

Parameter	Description
-g -p -v -f	<p>You can specify one of the following parameters when Oracle Clusterware checks the status of specific servers:</p> <ul style="list-style-type: none"> • -g: Use this parameter to check if the specified servers are registered • -p: Use this parameter to display static configuration of the specified servers • -v: Use this parameter to display the run-time configuration of the specified servers • -f: Use this parameter to display the full configuration of the specified servers
-category <i>category_name</i>	<p>Optionally, you can specify a particular category of servers for which to obtain status.</p>
-w " <i>filter</i> "	<p>Alternatively, specify a filter to determine which categories are displayed. The filter must be enclosed in double quotation marks (" "). Values that contain parentheses or spaces must be enclosed in single quotation marks (' '). For example, "STATE = ONLINE" limits the display to servers that are online.</p> <p>See Also: "Filters" for more information</p>

Examples

Example E-7 Example 1

This command returns output similar to the following:

```
NAME=node1
STATE=ONLINE

NAME=node2
STATE=ONLINE
```

Example E-8 Example 2

The full configuration of a specific server is similar to the following:

```
NAME=node2
MEMORY_SIZE=72626
CPU_COUNT=12
CPU_CLOCK_RATE=1711
CPU_HYPERTHREADING=0
CPU_EQUIVALENCY=1000
DEPLOYMENT=other
CONFIGURED_CSS_ROLE=hub
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=ora.pool1
```



```
STATE_DETAILS=  
ACTIVE_CSS_ROLE=hub
```

crsctl get server css_critical

Obtains the current configured value for the `CSS_CRITICAL` server attribute

Syntax

```
crsctl get server css_critical
```

Usage Notes

This command has no parameters.

crsctl set server css_critical

Sets the configuration value for the `CSS_CRITICAL` server attribute.

Syntax

```
crsctl set server css_critical {yes | no}
```

Usage Notes

Choose `yes` to set the server as `CSS_CRITICAL`.

crsctl get server label

Checks the current setting value of the `SERVER_LABEL` server attribute.

Syntax

```
crsctl get server label
```

Example

This command returns output similar to the following:

```
CRS-4972: Current SERVER_LABEL parameter value is hubserver
```

crsctl set server label

Sets the configuration value of the `SERVER_LABEL` server configuration attribute for the server on which you run this command.

Syntax

```
crsctl set server label value
```

Usage Notes

- Specify a value for the server. This value can reflect a physical location, such as `building_A`, or some other identifying characteristic of the server, such as `hubserver`.
- You must restart the Oracle Clusterware technology stack on the node before any changes you make take effect.

crsctl add serverpool

Adds a server pool that is for hosting non-database resources (such as application servers) to Oracle Clusterware.

Syntax

```
crsctl add serverpool server_pool_name {-file file_path |
-attrib "attr_name=attr_value[,...]" } [-i] [-f]
```

Parameters

Table E-78 crsctl add serverpool Command Parameters

Parameter	Description
<code>server_pool_name</code>	A short, descriptive name for the server pool. A server pool name has a 254 character limit and can contain any platform-supported characters except the exclamation point (!), the tilde (~), and spaces. A server pool name cannot begin with a period nor with <code>ora</code> .
<code>-file file_path</code>	Fully-qualified path to an attribute file to define the server pool.
<code>-attr "attr_name=attr_value[,...]"</code>	Alternatively, specify the name of a server pool attribute for the server pool you are adding preceded by the <code>-attr</code> flag, followed by a value for the server pool attribute. Note: The <code>attr_name=attr_value</code> name-value pairs must be enclosed in double quotation marks (" "). Multiple name-value pairs must be separated by commas. For example: <code>-attr "MAX_SIZE=30,IMPORTANCE=3"</code> The available attribute names include: <ul style="list-style-type: none"> • IMPORTANCE • MIN_SIZE • MAX_SIZE • SERVER_NAMES • PARENT_POOLS • EXCLUSIVE_POOLS • ACL • SERVER_CATEGORY
<code>-i</code>	If you specify <code>-i</code> , then the command fails if Oracle Clusterware cannot process the request immediately.

Table E-78 (Cont.) crsctl add serverpool Command Parameters

Parameter	Description
-f	If you specify the <code>-f</code> parameter, then Oracle Clusterware stops resources running on a server in another server pool and relocates that server into the server pool you are adding. If you do not specify the <code>-f</code> parameter, then Oracle Clusterware checks whether the creation of the server pool results in stopping any resources on a server in another server pool that is going to give up a server to the server pool you are adding. If so, then Oracle Clusterware rejects the <code>crsctl add serverpool</code> command.

Usage Notes

- The `server_pool_name` parameter is required.
- If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' ').
- Do not use this command for any server pools with names that begin with `ora` because these server pools are Oracle server pools.
- Running this command may result in Oracle Clusterware relocating other servers between server pools to comply with the new configuration.
- You must run this command as `root` or a cluster administrator.
- Use the `crsctl add serverpool` command to create server pools that host non-database resources. To create server pools that host Oracle databases, use the SRVCTL command utility.

 **See Also:**

Oracle Real Application Clusters Administration and Deployment Guide for information about using the SRVCTL command utility to create server pools

Examples

To add a server pool named `testsp` with a maximum size of 5 servers, run the following command as `root` or the Oracle Clusterware installation owner:

```
# crsctl add serverpool testsp -attr "MAX_SIZE=5"
```

Create the `sp1_attr` file with the attribute values for the `sp1` server pool, each on its own line, as shown in the following example:

```
IMPORTANCE=1
MIN_SIZE=1
MAX_SIZE=2
SERVER_NAMES=node3 node4 node5
PARENT_POOLS=Generic
```

```
EXCLUSIVE_POOLS=testsp  
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--
```

Use the following command to create the `sp1` server pool using the `sp1_attr` file as input:

```
$ crsctl add serverpool sp1 -file /tmp/sp1_attr
```

crsctl delete serverpool

Removes a server pool from the Oracle Clusterware configuration.

Syntax

```
crsctl delete serverpool server_pool_name [server_pool_name [...]] [-i]
```

See Also:

["How Server Pools Work"](#) for more information about server pools and server pool attributes

Usage Notes

- The `server_pool_name` parameter is required.
 - Use spaces to delimit multiple server pool names.
- If you specify `-i`, then the command fails if Oracle Clusterware cannot process the request immediately.
- Do not use this command for any server pools with names that begin with `ora` because these server pools are Oracle server pools.
- While you can use this command in either environment, it is only useful in the Oracle RAC environment.

Example

To delete a server pool, run the following command as `root` or the Oracle Clusterware installation owner:

```
# crsctl delete serverpool sp1
```

crsctl eval add serverpool

Predicts the effects of adding a server pool without making changes to the system.

This command may be useful to cluster administrators.

Syntax

```
crsctl eval add serverpool server_pool_name [-file file_path | -attr
"attr_name=attr_value[,...]"]
    [-admin [-l level [-x] [-a]] [-explain] [-f]
```

Parameters

Table E-79 crsctl eval add serverpool Command Parameters

Parameter	Description
<i>server_pool_name</i>	Specify a short, descriptive name for the server pool.
-file <i>file_path</i>	Specify a fully-qualified path to an attribute file to define the server pool.
-attr " <i>attr_name=attr_value</i> [,. ..]"	Alternatively, specify the name of a server pool attribute for the server pool you are adding preceded by the -attr flag, followed by a value for the server pool attribute. Note: The <i>attr_name=attr_value</i> name-value pairs must be enclosed in double quotation marks (" "). Multiple name-value pairs must be separated by commas. For example: -attr "MAX_SIZE=30,IMPORTANCE=3"
-admin [-l <i>level</i>] [-x] [-a]	Server pool attributes include: <ul style="list-style-type: none"> • IMPORTANCE • MIN_SIZE • MAX_SIZE • SERVER_NAMES • PARENT_POOLS • EXCLUSIVE_POOLS • ACL If you specify this parameter, then CRSCTL displays output for the cluster administrator. If you specify the -l parameter, then you can choose one of the following three output levels: <ul style="list-style-type: none"> • serverpools: Restricts the output to servers running in a server pool • resources: Restricts the output to resources running on servers in a server pool • all: Displays all available output If you specify the -x parameter, then CRSCTL displays the differences. If you specify the -a parameter, then CRSCTL displays all resources. Note: To specify either the -l, -x, or -a parameters, or any combination of the three, you must specify the -admin parameter.
-explain	Specify this parameter to display reasoned command evaluation information.
-f	If you specify this parameter, then CRSCTL predicts the effects of forcibly adding a server pool.

Example

The following example evaluates adding a server pool named `sp1`:

```
$ crsctl eval add serverpool sp1 -explain
```

```
Stage Group 1:
```

```
-----  
-  
Stage Required Action  
-----  
-  
  
1      E   Starting to look for server pools that will donate servers to  
be  
      E   reallocated to the new pool 'sp1'.  
      E   Scanning server pools with more than MIN_SIZE servers in  
      E   ascending order of IMPORTANCE.  
      E   Considering server pools (IMPORTANCE): Free(0) for suitable  
      E   servers.  
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it  
      E   has 4 servers above MIN_SIZE.  
      E   Relocating server 'scott_nodel_0' to server pool 'sp1'.  
      E   Scanning server pools with more than MIN_SIZE servers in  
      E   ascending order of IMPORTANCE.  
      E   Considering server pools (IMPORTANCE): Free(0) for suitable  
      E   servers.  
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it  
      E   has 3 servers above MIN_SIZE.  
      E   Relocating server 'scott_nodel_1' to server pool 'sp1'.  
      E   Scanning server pools with more than MIN_SIZE servers in  
      E   ascending order of IMPORTANCE.  
      E   Considering server pools (IMPORTANCE): Free(0) for suitable  
      E   servers.  
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it  
      E   has 2 servers above MIN_SIZE.  
      E   Relocating server 'scott_nodel_2' to server pool 'sp1'.  
      E   Scanning server pools with more than MIN_SIZE servers in  
      E   ascending order of IMPORTANCE.  
      E   Considering server pools (IMPORTANCE): Free(0) for suitable  
      E   servers.  
      E   Considering server pool 'Free' because its MIN_SIZE is 0 and it  
      E   has 1 servers above MIN_SIZE.  
      E   Relocating server 'scott_nodel_3' to server pool 'sp1'.  
      E   Scanning server pools with MIN_SIZE or fewer servers in  
      E   ascending order of IMPORTANCE.  
      Y   Server 'scott_nodel_2' will be moved from pool 'Free' to pool  
      Y   'sp1'.  
      Y   Server 'scott_nodel_0' will be moved from pool 'Free' to pool  
      Y   'sp1'.  
      Y   Server 'scott_nodel_1' will be moved from pool 'Free' to pool  
      Y   'sp1'.  
      Y   Server 'scott_nodel_3' will be moved from pool 'Free' to pool  
      Y   'sp1'.
```


Related Topics

- [How Server Pools Work](#)

crsctl eval delete serverpool

Simulates the deletion of a server pool without making changes to the system.

This command may be useful to cluster administrators.

Syntax

```
crsctl eval delete serverpool server_pool_name [-admin [-l level] [-x] [-a]] [-explain]
```

Parameters

Table E-80 crsctl eval delete serverpool Paramaters

Parameter	Description
<i>server_pool_name</i>	The name of the server pool you want to delete.
-admin [-l <i>level</i>] [-x] [-a]	<p>If you specify this parameter, then CRSCTL displays output for the cluster administrator.</p> <p>If you specify the -l parameter, then you can choose one of the following three output levels:</p> <ul style="list-style-type: none"> • <i>serverpools</i>: Restricts the output to servers running in a server pool • <i>resources</i>: Restricts the output to resources running on servers in a server pool • <i>all</i>: Displays all available output <p>If you specify the -x parameter, then CRSCTL displays the differences.</p> <p>If you specify the -a parameter, then CRSCTL displays all resources.</p> <p>Note:To specify either the -l, -x, or -a parameters, or any combination of the three, you must specify the -admin parameter.</p>
-explain	Specify this parameter to display reasoned command evaluation information.

Examples

Consider the following server pool configuration:

```
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=

NAME=sp1
ACTIVE_SERVERS=scott-node2-0 scott-node2-3

NAME=sp2
ACTIVE_SERVERS=scott-node2-1
```

The following command evaluates the effect of deleting server pool `sp1`:

```
$ crsctl eval delete sp sp1

Stage Group 1:
-----
-----
Stage Required Action
-----
-----

  1 Y Server 'scott-node2-0' will be moved from pool 'sp1' to pool
      'sp2'.
      Y Server 'scott-node2-3' will be moved from pool 'sp1' to pool
      'sp2'.

-----
-----
```

Adding the `-explain` parameter to the preceding command returns reasoned command evaluation similar to the following:

```
$ crsctl eval delete sp sp1 -explain

Stage Group 1:
-----
-----
Stage Required Action
-----
-----

  1 E Starting to look for server pools to reallocate servers from
      pool 'sp1'.
      E Looking for a suitable server pool for server
      'scott-node2-0'.
      E Scanning server pools with fewer than MIN_SIZE servers in
      descending order of IMPORTANCE.
```



```

E Considering server pools (IMPORTANCE): sp2(1) for relocating
  servers.
E Relocating server 'scott-node2-0' to server pool 'sp2'.
E Looking for a suitable server pool for server
  'scott-node2-3'.
E Scanning server pools with fewer than MIN_SIZE servers in
  descending order of IMPORTANCE.
E Considering server pools (IMPORTANCE): sp2(1) for relocating
  servers.
E Relocating server 'scott-node2-3' to server pool 'sp2'.
2 Y Server 'scott-node2-0' will be moved from pool 'sp1' to pool
  'sp2'.
  Y Server 'scott-node2-3' will be moved from pool 'sp1' to pool
    'sp2'.

```


Related Topics

- [How Server Pools Work](#)

crsctl eval modify serverpool

Use the `crsctl eval modify serverpool` command to predict the effects of modifying a server pool without making changes to the system. This command may be useful to cluster administrators.

Syntax

```

crsctl eval modify serverpool server_pool_name {-file file_path
| -attr "attr_name=attr_value [,attr_name=attr_value[, ...]]"}
[-f] [-admin [-l level [-x] [-a]] [-explain]

```

Parameters

Table E-81 crsctl eval modify serverpool Parameters

Parameter	Description
<i>server_pool_name</i>	The name of the server pool you want to modify.
-file <i>file_path</i>	Fully-qualified path to an attribute file to define the server pool you want to modify.

Table E-81 (Cont.) crsctl eval modify serverpool Parameters

Parameter	Description
<i>attribute_name</i>	The name of a server pool attribute that you want to modify that Oracle Clusterware uses to manage the server pool preceded by the <code>-attr</code> flag. The available attribute names include: <ul style="list-style-type: none"> • IMPORTANCE • MIN_SIZE • MAX_SIZE • SERVER_NAMES • PARENT_POOLS • EXCLUSIVE_POOLS • ACL
<i>attribute_value</i>	A value for the server pool attribute you want to modify. Note: The <i>attribute_name</i> and <i>attribute_value</i> parameters must be enclosed in double quotation marks (") and separated by commas. For example: <code>-attr "MAX_SIZE=30,IMPORTANCE=3"</code>
<code>-f</code>	If you specify this parameter, then CRSTL predicts the effects of forcibly modifying a server pool.
<code>-admin [-l level] [-x] [-a]</code>	If you specify this parameter, then CRSTL displays output for the cluster administrator. If you specify the <code>-l</code> parameter, then you can choose one of the following three output levels: <ul style="list-style-type: none"> • <code>serverpools</code>: Restricts the output to servers running in a server pool • <code>resources</code>: Restricts the output to resources running on servers in a server pool • <code>all</code>: Displays all available output If you specify the <code>-x</code> parameter, then CRSTL displays the differences. If you specify the <code>-a</code> parameter, then CRSTL displays all resources. Note: To specify either the <code>-l</code> , <code>-x</code> , or <code>-a</code> parameters, or any combination of the three, you must specify the <code>-admin</code> parameter.
<code>-explain</code>	Specify this parameter to display reasoned command evaluation information.

Usage Notes

- The *server_pool_name* parameter is required
- If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' '). For example:

`"START_DEPENDENCIES='hard(res1,res2,res3)'"`
- Running this command may result in Oracle Clusterware relocating other servers between server pools to comply with the new configuration
- Do not use this command for any server pools with names that begin with *ora* because these server pools are Oracle server pools

- While you can use this command in either environment, it is only useful in the Oracle RAC environment

Example

For the following examples, consider two server pools, `sp1` and `sp2`. Server pool `sp1` is configured, as follows:

```
ACTIVE_SERVERS=mjkeenan-node2-0 mjkeenan-node2-3
```

Server pool `sp2` is configured, as follows:

```
ACTIVE_SERVERS=mjkeenan-node2-1
```

Currently, `sp1` has more importance than `sp2`. The following command evaluates what would happen if you modified the `IMPORTANCE` and `MAX_SIZE` attributes for `sp2`:

```
$ crsctl eval modify sp sp2 -attr "MAX_SIZE=3,IMPORTANCE=5"
```

```
Stage Group 1:
```

```
-----
-----
Stage Required Action
-----
-----
```

```
      1 Y Server 'mjkeenan-node2-0' will be moved from pool 'sp1' to pool
          'sp2'.
          Y Server 'mjkeenan-node2-3' will be moved from pool 'sp1' to
pool
          'sp2'.
```

```
-----
-----
```

Adding the `-explain` parameter to the preceding command returns reasoned command evaluation similar to the following:

```
$ crsctl eval modify sp sp2 -attr "MAX_SIZE=3,IMPORTANCE=5" -explain
```

```
Stage Group 1:
```

```
-----
-----
Stage Required Action
-----
-----
```

```
      1 E Starting to look at server pools due to a configuration change
          in the pool 'sp2'.
          E Looking at other server pools to see whether MIN_SIZE value 3
of
          server pool 'sp2' can be met.
          E Scanning server pools with MIN_SIZE or fewer servers in
```

```

        ascending order of IMPORTANCE.
E Considering server pools (IMPORTANCE): sp1(3) for suitable
  servers.
E Considering server pool 'sp1' because its MIN_SIZE is 2 and it
  has 0 servers above MIN_SIZE.
E Relocating server 'mjkeenan-node2-0' to server pool 'sp2'.
E Scanning server pools with MIN_SIZE or fewer servers in
  ascending order of IMPORTANCE.
E Considering server pools (IMPORTANCE): sp1(3) for suitable
  servers.

E Considering server pool 'sp1' because its MIN_SIZE is 2 and it
  has 0 servers above MIN_SIZE.
E Relocating server 'mjkeenan-node2-3' to server pool 'sp2'.
Y Server 'mjkeenan-node2-0' will be moved from pool 'sp1' to
pool
        'sp2'.
Y Server 'mjkeenan-node2-3' will be moved from pool 'sp1' to
pool
        'sp2'.
-----
-----

```

Related Topics

- [How Server Pools Work](#)

crsctl getperm serverpool

Retrieves permissions for a particular server pool.

Syntax

```
crsctl getperm serverpool server_pool_name [-u user_name | -g group_name]
```



See Also:

"[How Server Pools Work](#)" for more information about server pools and server pool attributes

Parameters

Table E-82 crsctl getperm serverpool Command Parameters

Parameter	Description
<i>server_pool_name</i>	Specify the name of the server pool for which you want to retrieve permissions.
-u <i>user_name</i>	If you specify -u, then Oracle Clusterware retrieves permissions for a particular user.

Table E-82 (Cont.) crsctl getperm serverpool Command Parameters

Parameter	Description
<code>-g group_name</code>	If you specify <code>-g</code> , then Oracle Clusterware retrieves permissions for a particular group.

Usage Notes

- The `server_pool_name` parameter is required
- Do not use this command for any server pools with names that begin with `ora` because these server pools are Oracle server pools
- While you can use this command in either environment, it is only useful in the Oracle RAC environment

Example

This command returns output similar to the following:

```
$ crsctl getperm serverpool sp1

NAME: sp1
owner:root:rwX,pgrp:root:r-x,other::r--
```

crsctl modify serverpool

Modifies an existing server pool.

Syntax

```
crsctl modify serverpool server_pool_name -attr
"attr_name=attr_value[, ...]"
[-policy policyName | -all_policies] [-i] [-f]
```

Parameters

Table E-83 crsctl modify serverpool Command Parameters

Parameter	Description
<code>server_pool_name</code>	Specify the name of the server pool you want to modify.

Table E-83 (Cont.) crsctl modify serverpool Command Parameters

Parameter	Description
-attr "attr_name=attr_value[, ...]"	<p>Specify the name of a server pool attribute that you want to modify preceded by the <code>-attr</code> flag, followed by a value for the server pool attribute.</p> <p>Note: The <code>attr_name=attr_value</code> name-value pairs must be enclosed in double quotation marks (" "). Multiple name-value pairs must be separated by commas. For example:</p> <pre>-attr "MAX_SIZE=30,IMPORTANCE=3"</pre> <p>The available attribute names include:</p> <ul style="list-style-type: none"> • IMPORTANCE • MIN_SIZE • MAX_SIZE • SERVER_NAMES • PARENT_POOLS • EXCLUSIVE_POOLS • ACL • SERVER_CATEGORY
-policy <i>policyName</i> -all_policies	Specify a particular policy or all policies for which you want to modify the server pool definition.
-i	If you specify <code>-i</code> , then the command fails if Oracle Clusterware cannot process the request immediately.
-f	<p>If you specify the <code>-f</code> parameter, then Oracle Clusterware stops resources running on a server in another server pool and relocates that server into the server pool you are modifying.</p> <p>If you do not specify the <code>-f</code> parameter, then Oracle Clusterware checks whether the creation of the server pool results in stopping any resources on a server in another server pool that is going to give up a server to the server pool you are modifying. If so, then Oracle Clusterware rejects the <code>crsctl modify serverpool</code> command.</p>



See Also:

"[How Server Pools Work](#)" for more information about server pools and server pool attributes

Usage Notes

- The `server_pool_name` parameter is required.
- If an attribute value for an attribute name-value pair contains commas, then the value must be enclosed in single quotation marks (' '). For example:

```
"START_DEPENDENCIES='hard(res1,res2,res3)'"
```

- Running this command may result in Oracle Clusterware relocating other servers between server pools to comply with the new configuration.
- Do not use this command for any server pools with names that begin with *ora* because these server pools are Oracle server pools.
- While you can use this command in either environment, it is only useful in the Oracle RAC environment.

Example

To modify an existing server pool, run the following command as `root` or the Oracle Clusterware installation owner:

```
# crsctl modify serverpool sp1 -attr "MAX_SIZE=7"
```

crsctl setperm serverpool

Sets permissions for a particular server pool.

Syntax

```
crsctl setperm serverpool server_pool_name {-u acl_string | -x acl_string |  
-o user_name | -g group_name}
```

Parameters

Table E-84 crsctl setperm serverpool Command Parameters

Parameter	Description
<i>server_pool_name</i>	Specify the name of the server pool for which you want to set permissions.

Table E-84 (Cont.) crsctl setperm serverpool Command Parameters

Parameter	Description
{-u -x -o -g}	<p>You can specify only one of the following parameters for a server pool:</p> <ul style="list-style-type: none"> -u <i>acl_string</i>: You can update the access control list (ACL) for a server pool -x <i>acl_string</i>: You can delete the ACL for a server pool -o <i>user_name</i>: You can change the owner of a server pool by entering a user name -g <i>group_name</i>: You can change the primary group of a server pool by entering a group name <p>Specify a user, group, or other ACL string, as follows:</p> <pre>user:user_name[:readPermwritePermexecPerm] group:group_name[:readPermwritePermexecPerm] other[::readPermwritePermexecPerm]</pre> <ul style="list-style-type: none"> user: User ACL group: Group ACL other: Other ACL readPerm: Read permission for the server pool; the letter r grants a user, group, or other read permission, the minus sign (-) denies read permission writePerm: Write permission for the server pool; the letter w grants a user, group, or other write permission, the minus sign (-) denies write permission execPerm: Execute permission for the server pool; the letter x grants a user, group, or other execute permission, the minus sign (-) denies execute permission

Usage Notes

- The *server_pool_name* parameter is required
- Do not use this command for any server pools with names that begin with *ora* because these server pools are Oracle server pools
- While you can use this command in either environment, it is only useful in the Oracle RAC environment

Example

To grant read, write, and execute permissions on a server pool for user Jane Doe:

```
$ crsctl setperm serverpool sp3 -u user:jane.doe:rwX
```

crsctl status serverpool

Retrieves the status and configuration information of one or more particular server pools.

Syntax

```
crsctl status serverpool [server_pool_name [...] | -w "filter"] [-p | -v | -f]
```

```
crsctl status serverpool {[server_pool_name [...]} -g
```

Parameters

Table E-85 crsctl status serverpool Command Parameters

Parameter	Description
<i>server_pool_name</i> [...] -g	Specify one or more space-delimited server pool names to identify specific server pools. -g: Use this parameter to check if the specified server pools are registered Note: You <i>cannot</i> use the -g parameter with any other parameters after you specify the server pool names.
-w " <i>filter</i> "	Alternative to specifying a server pool by name, you can use this parameter to specify a filter, such as MIN_SIZE > 3, surrounded by double quotation marks (" "). Use this parameter to identify server pools by a particular characteristic. See Also: " Filters " for more information
-p -v -f	You can optionally specify one of the following parameters: <ul style="list-style-type: none"> -p: Use this parameter to display static configuration of the specified server pools -v: Use this parameter to display the run-time configuration of the specified server pools -f: Use this parameter to display the full configuration of the specified server pools

Usage Notes

- The *server_pool_name* parameter or a filter is required
- Do not use this command for any server pools with names that begin with *ora* because these server pools are Oracle server pools
- While you can use this command in either environment, it is only useful in the Oracle RAC environment

Examples

Example E-9 Example 1

To display the full configuration of the server pool *sp1*:

```
$ crsctl status serverpool sp1 -f
```

```
NAME=sp1
IMPORTANCE=1
MIN_SIZE=0
MAX_SIZE=-1
```

```
SERVER_NAMES=node3 node4 node5
PARENT_POOLS=Generic
EXCLUSIVE_POOLS=
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--
SERVER_CATEGORY=ora.hub.category
ACTIVE_SERVERS=node3 node4
```

Example E-10 Example 2

To display all the server pools and the servers associated with them, use the following command:

```
$ crsctl status serverpool
```

```
NAME=Free
ACTIVE_SERVERS=

NAME=Generic
ACTIVE_SERVERS=node1 node2

NAME=ora.orcl
ACTIVE_SERVERS=node1 node2

NAME=sp1
ACTIVE_SERVERS=node3 node4
```

Example E-11 Example 3

To find a server pool that meets certain criteria, use the following command:

```
$ crsctl status serverpool -w "MAX_SIZE > 1"
```

```
NAME=sp2
ACTIVE_SERVERS=node3 node4
```

crsctl query socket udp

Verifies that a daemon can listen on a specified address and port.

Syntax

```
crsctl query socket udp [-address address] [-port port]
```

Parameters

Table E-86 crsctl query socket udp Command Parameters

Parameter	Description
-address <i>address</i>	Specify the IP address on which the socket is to be created. If you do not specify an address, then CRSCTL assumes the local host as the default.

Table E-86 (Cont.) crsctl query socket udp Command Parameters

Parameter	Description
<code>-port port</code>	Specify the port on which the socket is to be created. If you do not specify a port, then CRSCTL assumes 53 as the default.

Usage Notes

You must run this command as `root` to verify port numbers less than 1024.

Examples

The command returns output similar to the following:

```
$ crsctl query socket udp
```

```
CRS-10030: could not verify if port 53 on local node is in use
```

```
# crsctl query socket udp
```

```
CRS-10026: successfully created socket on port 53 on local node
```

The first of the preceding two commands was not run as `root`, and in both commands no port was specified, so CRSCTL assumed the default, 53, which is less than 1024. This condition necessitates running the command as `root`.

```
$ crsctl query socket udp -port 1023
```

```
CRS-10030: could not verify if port 1023 on local node is in use
```

```
# crsctl query socket udp -port 1023
```

```
CRS-10026: successfully created socket on port 1023 on local node
```

Similar to the first two examples, the first of the preceding two commands was not run as `root`, and, although a port number was specified, it is still less than 1024, which requires `root` privileges to run the command.

In this last example, a port number greater than 1024 is specified, so there is no need to run the command as `root`:

```
$ crsctl query socket udp -port 1028
```

```
CRS-10026: successfully created socket on port 1028 on local node
```

crsctl start testdns

Starts a test DNS server that will listen on a specified IP address and port. The test DNS server does not respond to incoming packets but does display the packets it receives. Typically, use this command to check if domain forwarding is set up correctly for the GNS domain.

Syntax

```
crsctl start testdns [-address address [-port port]] [-once] [-v]
```

Parameters

Table E-87 crsctl start testdns Command Parameters

Parameter	Description
-address <i>address</i>	Specify a server address in the form <i>IP_address/netmask[/interface_name]</i> .
-port <i>port</i>	The port on which the server listens. If you do not specify a port, then it defaults to port 53.
-once	Specify this parameter to indicate that the DNS server should exit after it receives one DNS query packet.
-v	Verbose output.

Example

To start a test DNS server on the local server, run the command similar to the following:

```
$ crsctl start testdns -address 192.168.29.218 -port 63 -v
```

crsctl status testdns

Queries the test DNS server running on a specified address and local host name.

Syntax

```
crsctl status testdns [-server DNS_server_address] [-port port] [-v]
```

Parameters

Table E-88 crsctl status testdns Command Parameters

Parameter	Description
-server <i>DNS_server_address</i>	Optionally, you can specify the DNS server address for which you want to check the status.
-port <i>port</i>	Optionally, you can specify the port on which the DNS server listens. If you do not specify a port, then it defaults to port 53.
-v	Verbose output.

Example

This command returns output similar to the following:

```
CRS-10024: DNS server returned 192.168.28.74 for name
stmjk07-vip.stmjk0462.foo.com
```

crsctl stop testdns

Stops a test DNS server.

Syntax

```
crsctl stop testdns [-address address] [-port port] [-v]
```

Parameters

Table E-89 crsctl stop testdns Command Parameters

Parameter	Description
-address <i>address</i>	Optionally, you can specify the server address for which you started the test DNS server in the form <i>IP_address/netmask[/interface_name]</i> .
-port <i>port</i>	Optionally, you can specify the port on which the DNS server listens. If you do not specify a port, then it defaults to port 53.
-v	Verbose output.

Example

This command returns output similar to the following:

```
CRS-10032: successfully stopped the DNS listening daemon running on port
53 on
local node
```

crsctl replace votedisk

Moves or replaces the existing voting files.

This command creates voting files in the specified locations, either in Oracle ASM or some other storage option. Oracle Clusterware copies existing voting file information into the new locations and removes the voting files from the former locations.

Syntax

```
crsctl replace votedisk [+asm_disk_group | path_to_voting_disk [...]]
```

Parameters

Table E-90 crsctl replace votedisk Command Parameters

Parameter	Description
<code>+asm_disk_group</code>	Specify the Oracle ASM disk group (preceded by a plus sign (+)) where you want to locate the voting file.
<code>path_to_voting_disk</code> [...]	Alternatively, you can specify a space-delimited list of voting file paths for voting files that reside outside of Oracle ASM.

Usage Notes

- You must be `root`, the Oracle Clusterware installation owner, or a member of the Administrators group to run this command.
- Specify to replace a voting file in either an Oracle ASM disk group or in some other storage device.
- You can run this command on any node in the cluster.

Examples

Example E-12 Example 1

To replace a voting file that is located within Oracle ASM:

```
# crsctl replace votedisk +diskgroup1
```

Example E-13 Example 2

To replace a voting file that is located on a shared file system:

```
# crsctl replace votedisk /mnt/nfs/disk1 /mnt/nfs/disk2
```

Oracle Restart Environment CRSCTL Commands

You can only use these commands in the Oracle Restart environment.

The commands listed in this section control Oracle High Availability Services. These commands manage the Oracle High Availability Services stack in the Oracle Restart environment, which consists of the Oracle High Availability Services daemon (`ohasd`), Oracle ASM (if installed), and Cluster Synchronization Services (if Oracle ASM is installed). These commands only affect the local server on which you run them.

 **Note:**

Oracle does not support using `crs_*` commands in an Oracle Restart environment.

Each server in the cluster is in one of two possible states:

- The whole stack is up, which means that Oracle High Availability Services is active
- The whole stack is down, which means that Oracle High Availability Services is inactive

crsctl check has

Checks the status of ohasd.

Syntax

```
crsctl check has
```

Example

This command returns output similar to the following:

```
CRS-4638: Oracle High Availability Services is online
```

crsctl config has

Displays the automatic startup configuration of the Oracle High Availability Services stack on the server.

Syntax

```
crsctl config has
```

Usage Notes

This command has no parameters.

Example

This command returns output similar to the following:

```
CRS-4622 Oracle High Availability Services autostart is enabled.
```

crsctl disable has

Disables automatic startup of the Oracle High Availability Services stack when the server boots up.

Syntax

```
crsctl disable has
```

Example

This command returns output similar to the following:

```
CRS-4621 Oracle High Availability Services autostart is disabled.
```

crsctl enable has

Enables automatic startup of the Oracle High Availability Services stack when the server boots up.

Syntax

```
crsctl enable has
```

Example

This command returns output similar to the following:

```
CRS-4622 Oracle High Availability Services autostart is enabled.
```

crsctl query has releaseversion

Displays the release version of the Oracle Clusterware software that is stored in the binaries on the local node.

Syntax

```
crsctl query has releaseversion
```

Example

This command returns output similar to the following:

```
Oracle High Availability Services release version on the local node is  
[12.2.0.1.0]
```

crsctl query has softwareversion

Displays the software version on the local node.

Syntax

```
crsctl query has softwareversion [server_name]
```

Usage Notes

If you do not provide a server name, then Oracle Clusterware displays the version of Oracle Clusterware running on the local server.

Example

This command returns output similar to the following:

```
Oracle High Availability Services version on the local node is [12.2.0.1.0]
```


crsctl start has

Starts Oracle High Availability Services on the local server.

Syntax

```
crsctl start has [-nowait] [-noautostart] [-wait]
```

Usage Notes

- Use the `-nowait` parameter to skip waiting for Oracle High Availability Services to start.
- Use the `-noautorestart` parameter to start only Oracle High Availability Services.
- Use the `-wait` parameter to wait for Oracle High Availability Services to start.

Example

To start Oracle High Availability Services on the local server:

```
# crsctl start has
```

crsctl stop has

Stops Oracle High Availability Services on the local server. This command attempts to gracefully stop resources managed by Oracle Clusterware while attempting to stop Oracle High Availability Services.

Syntax

```
crsctl stop has [-f]
```

Usage Notes

If any resources that Oracle Clusterware manages are still running after you run the `crsctl stop has` command, then the command fails. Use the `-f` parameter to unconditionally stop all resources and stop Oracle High Availability Services.

Example

To stop Oracle High Availability Services on the local server:

```
# crsctl stop has
```

Troubleshooting and Diagnostic Output

You can use `crsctl set log` commands as the `root` user to enable dynamic debugging for Cluster Ready Services (CRS), Cluster Synchronization Services (CSS), and the Event Manager (EVM), and the clusterware subcomponents.

You can dynamically change debugging levels using `crsctl debug` commands. Debugging information remains in the Oracle Cluster Registry (OCR) for use during the next startup. You can also enable debugging for resources.

This section includes the following topics:

- [Dynamic Debugging Using `crsctl set log`](#)
- [Component Level Debugging](#)

Dynamic Debugging Using `crsctl set log`

Setting log levels for Oracle Clusterware.

Syntax

```
crsctl set log module_name "component1=level1,..."
```

Usage Notes

- You can set log levels for various components of the following modules:

- `mdns`: Multicast domain name server
- `gppp`: Grid Plug and Play service
- `css`: Cluster Synchronization Services
- `crf`: Cluster Health Monitor
- `crs`: Cluster Ready Services
- `ctss`: Cluster Time Synchronization Service
- `evm`: Event Manager
- `gipc`: Grid Interprocess Communication

Use the `crsctl lsmodules` command to obtain a list of components for each module.

- You can specify multiple component name/log level pairs in a comma-delimited list enclosed in double quotation marks ("").
- Separate component name/log level pairs with an equals sign (=) in Oracle Clusterware 11g release 2 (11.2.0.3), and later. Previous Oracle Clusterware versions used a colon (:).
- The `log_level` is a number from 1 to 5 that sets the log level for the component or resource, where 1 is the least amount of log output and 5 provides the most detailed log output. The default log level is 2.

Example

To set log levels for the CRSRTI and CRSCOMM components of the CRS module:

```
$ crsctl set log crs "CRSRTI=1,CRSCOMM=2"
```

Component Level Debugging

You can use `crsctl set log` and `crsctl set trace` commands as the root user to enable dynamic debugging for the various Oracle Clusterware modules.

Enabling Debugging for Oracle Clusterware Modules

You can enable debugging for Oracle Clusterware modules and their components, and for resources, by setting environment variables or by running the `crsctl set log` command.

Use the following syntax to enable debugging for various Oracle Clusterware modules:

```
crsctl set {log | trace} module_name "component:debugging_level[,...]"
```

Run the `crsctl set` command as the `root` user, and supply the following information:

- *module_name*: Specify the name of one of the following modules:
 - mdns: Multicast domain name server
 - gpnv: Grid Plug and Play service
 - css: Cluster Synchronization Services
 - crf: Cluster Health Monitor
 - crs: Cluster Ready Services
 - ctss: Cluster Time Synchronization Service
 - evm: Event Manager
 - gipc: Grid Interprocess Communication
- *component*: Specify the name of a component for one of the modules. See [Table E-91](#) for a list of all of the components.
- *debugging_level*: Specify a number from 1 to 5 to indicate the level of detail you want the debug command to return, where 1 is the least amount of debugging output and 5 provides the most detailed debugging output. The default debugging level is 2.

The following examples show how to enable debugging for the various modules:

- To enable debugging for Oracle Clusterware:

```
# crsctl set log crs "CRSRTI:1,CRSCOMM:2"
```

- To enable debugging for OCR:

```
# crsctl set log crs "CRSRTI:1,CRSCOMM:2,OCRSRV:4"
```

- To enable debugging for EVM:

```
# crsctl set log evm "EVMCOMM:1"
```

- To enable debugging for resources

```
# crsctl set log res "resname:1"
```

To obtain a list of components that can be used for debugging, run the `crsctl lsmodules` command, with the following syntax:

```
$ crsctl lsmodules {mdns | gpnv | css | crf | crs | ctss | evm | gipc}
```

 **Note:**

You do not have to be the `root` user to run the `crsctl lsmodules` command.

Table E-91 shows the components for the CRS, CSS, and EVM modules, respectively. Note that some component names are common between the CRS, EVM, and CSS daemons and may be enabled on that specific daemon. For example, `COMMNS` is the NS layer and because each daemon uses the NS layer, you can enable this specific module component on any of the daemons to get specific debugging information.

Table E-91 Components for the CRS, CSS, and EVM Modules

CRS Components ¹	CSS Components ²	EVM Components ³
CRSUI	CSSD	EVM
CRSCOMM	COMMCRS	EVMDMAIN
CRSRTI	COMMNS	EVMCOMM
CRSMAIN		EVMEVT
CRSPLACE		EVMAPP
CRSAPP		EVMAGENT
CRSRES		CRSOCR
CRSCOMM		CLUCLS
CRSOCR		CSSCLNT
CRSTIMER		COMMCRS
CRSEVT		COMMNS
CRSD		
CLUCLS		
CSSCLNT		
COMMCRS		
COMMNS		

¹ Obtain the list of CRS components using the `crsctl lsmodules crs` command.

² Obtain the list of CSS components using the `crsctl lsmodules css` command.

³ Obtain the list of EVM components using the `crsctl lsmodules evm` command.

For example, use the following command to set debugging levels on specific cluster nodes, include the `-nodelist` keyword and the names of the nodes, as follows:

```
crsctl set log crs "CRSRTI:1,CRSCOMM:2" -nodelist node1,node2
```

Table E-92 describes the Cluster Synchronization Services modules.

Table E-92 Cluster Synchronization Services (CSS) Modules and Functions

Module	Description
CSS	CSS client component
CSSD	CSS daemon component

[Table E-93](#) describes the function of each communication (COMM) module.

Table E-93 Communication (COMM) Modules and Functions

Module	Description
COMMCRS	Clusterware communication layer
COMMNS	NS communication layer

[Table E-94](#) describes the functions performed by each CRS module.

Table E-94 Oracle Clusterware (CRS) Modules and Functions

Module	Descriptions
CRSUI	User interface module
CRSCOMM	Communication module
CRSRTI	Resource management module
CRSMAIN	Main module/driver
CRSPLACE	CRS placement module
CRSAPP	CRS application
CRSRES	CRS resources
CRSOCR	Oracle Cluster Registry interface
CRSTIMER	Various timers related to CRS
CRSEVT	CRS EVM/event interface module
CRSD	CRS daemon

Using the `crsctl set log crs` command, you can debug the OCR components listed in [Table E-95](#). The components listed in [Table E-95](#) can also be used for the Oracle Local Registry (OLR) except for OCRMAS and OCRASM. You can also use them for OCR and OLR clients, except for OCRMAS and OCRSRV. Some OCR and OLR clients are OCRCONFIG, OCRDUMP, and so on.

Table E-95 Oracle Cluster Registry (OCR) Component Names

Module	Description
OCRAPI	OCR abstraction component
OCRCLI	OCR client component
OCRSRV	OCR server component
OCRMAS	OCR master thread component
OCRMSG	OCR message component
OCRCAC	OCR cache component
OCRRAW	OCR raw device component
OCRUTL	OCR util component
OCROSD	OCR operating system dependent (OSD) layer
OCRASM	OCR ASM component

Table E-96 describes the OCR tool modules.

Table E-96 OCRCONFIG Modules and Functions

Module	Description
OCRCONFIG	OCRCONFIG component for configuring OCR
OCRDUMP	OCRDUMP component that lists the Oracle Cluster Registry contents
OCRCHECK	OCRCHECK component that verifies all of the configured OCRs

Enabling Debugging for Oracle Clusterware Resources

You can enable debugging for Oracle Clusterware resources by running the `crsctl set log` command, using the following syntax:

```
crsctl set log res "resource_name=debugging_level"
```

Run the `crsctl set log` command as the `root` user, and supply the following information:

- *resource_name*: The name of the resource to debug.
- *debugging_level*: A number from 1 to 5 to indicate the level of detail you want the debug command to return, where 1 is the least amount of debugging output and 5 provides the most detailed debugging output.

Note:

After you resolve an issue that required additional debugging, you can reset the debug level to default, as follows:

```
# crsctl set log crs resource_name=default
```

To obtain a list of resources that can be used for debugging, run the `crsctl status resource` command.

Example

To generate a debugging log for the VIP resource on `node1`, issue the following command:

```
# crsctl set log res "ora.node1.vip:1"
```

F

Fleet Patching and Provisioning Control (RHPCTL) Command Reference

Use the Fleet Patching and Provisioning Control (RHPCTL) utility to manage Fleet Patching and Provisioning in your cluster.

This appendix contains reference information for Fleet Patching and Provisioning commands, including utility usage information and a comprehensive listing of the RHPCTL commands.

RHPCTL Overview

RHPCTL is a command-line utility with which you perform Fleet Patching and Provisioning operations and manage Fleet Patching and Provisioning Servers and Clients. RHPCTL uses the following syntax:

```
rhctl command object [parameters]
```

In RHPCTL syntax:

- *command* is a verb such as `add`, `delete`, or `query`
- *object* (also known as a noun) is the target or object on which RHPCTL performs the command, such as `client` or `image`.
- *parameters* extend the use of a preceding command combination to include additional parameters for the command. Specify parameters as `-keyword value`. If the *value* field contains a comma-delimited list, then do not use spaces between the items in the list.

You can use RHPCTL commands to perform several Fleet Patching and Provisioning operations, including:

- Fleet Patching and Provisioning Client operations, such as creating a Fleet Patching and Provisioning Client configuration.
- Role operations, such as adding and deleting roles, and granting and revoking roles for users.
- Site operations, such as obtaining configuration information for Fleet Patching and Provisioning Servers.
- Image operations, such as adding, deleting, and importing images.
- Image series operations, such as adding and deleting image series.
- Working copy operations, such as adding and deleting working copies.

Using RHPCTL Help

To see help for all RHPCTL commands, from the command line enter:

```
rhctl -help
```

To see the command syntax and a list of parameters for each RHPCTL command, from the command line enter:

```
rhpcctl command (or verb) object (or noun) -help
```

RHPCTL Command Reference

This section describes RHPCTL command usage information, and lists and describes RHPCTL commands.

Note:

A multitenant container database is the only supported architecture in Oracle Database 20c. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

rhpcctl delete audit

Deletes the Fleet Patching and Provisioning audit records.

Syntax

```
rhpcctl delete audit [-to timestamp]
```

Usage Notes

Optionally, you can specify a date up to which audit records will be deleted, in the format *YYYY-MM-DD*. Otherwise, this command deletes all audit records.

rhpcctl modify audit

Modifies the maximum number of audit records to store.

Syntax

```
rhpcctl modify audit -maxrecord number
```

Usage Notes

Specify the maximum number of audit records to store.

rhpcctl query audit

Displays the Fleet Patching and Provisioning audit records.

Syntax

```

rhpctl query audit [[-operation {add | delete | modify | grant | revoke |
move | verify | discover
| upgrade | allow | disallow | deleteimage | insertimage | promote |
addnode | deletenode | register | unregister | export | import | query
| subscribe | unsubscribe}]
[-entity {client | role | audit | image | imagetype | useraction |
series | workingcopy | database | server | user | audit | imagetype |
useraction}]
| [-user user_name] [-client cluster_name] | [-from timestamp -to
timestamp]
| -before timestamp | -since timestamp | -first number | -last number]
| -record record_id | -config]

```

Parameters

Table F-1 rhpctl query audit Command Parameters

Parameter	Description
-operation {add delete modify grant revoke move verify discover upgrade allow disallow deleteimage insertimage promote addnode deletenode register unregister export import query subscribe unsubscribe}	Specify the type of operation for which you want an audit query.
-entity {client role image series workingcopy database server user audit imagetype useraction}	Specify the entity for which you want an audit query.
-user <i>user_name</i>	Optionally, you can choose to run a query audit on a particular user who performed Fleet Patching and Provisioning operations.
-client <i>cluster_name</i>	Optionally, you can choose to run a query audit on a particular client cluster where Fleet Patching and Provisioning operations were performed.
-from <i>timestamp</i> -to <i>timestamp</i>	Optionally, you can specify a time interval for which to run an audit query. Timestamps must be in the format <i>YYYY-MM-DD</i> .
-before <i>timestamp</i>	Optionally, you can specify a time before which to run an audit query. Timestamp must be in the format <i>YYYY-MM-DD</i> .

Table F-1 (Cont.) rhpctl query audit Command Parameters

Parameter	Description
<code>-since timestamp</code>	Optionally, you can specify a time after which to run an audit query. Timestamp must be in the format <code>YYYY-MM-DD</code> .
<code>-first number</code>	Optionally, you can specify a number of the first audit records for a given time.
<code>-last number</code>	Optionally, you can specify a number of the last audit records for a given time.
<code>-record record_id</code>	Optionally, you can specify a particular audit record ID.
<code>-config</code>	You can choose this parameter to show the maximum record configuration.

rhpctl add client

Adds a Fleet Patching and Provisioning Client to the Fleet Patching and Provisioning Server configuration.

Syntax

```
rhpctl add client -client cluster_name [-toclientdata path] [-targetnode
node_name
  {-sudouser sudo_user_name -sudopath sudo_binary_location | -root |
  -cred cred_name} | -auth plugin_name [-arg1 name1:value1...]]
  [-maproles role=user_name[,role=user_name[,...]]]
  [-version version]
```

Parameters

Table F-2 rhpctl add client Command Parameters

Parameter	Description
<code>-client client_name</code>	Specify the name of the cluster in which you want to create the client.
<code>-toclientdata path</code>	Optionally, you can specify the path to the XML file that is created by the Fleet Patching and Provisioning Server (specific to the client cluster), which contains the information the client needs to configure its connection to the server.
<code>-targetnode node_name</code>	Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client.
<code>-sudouser sudo_user_name -sudopath sudo_binary_location -root -cred cred_name</code>	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>sudo</code> or <code>root</code> to access the remote node. If you choose <code>sudo</code> , then you must use the <code>-sudouser</code> parameter and specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.

Table F-2 (Cont.) rhpctl add client Command Parameters

Parameter	Description
<code>-auth <i>plugin_name</i> [-arg1 <i>name1:value1...</i>]</code>	Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
<code>-maproles <i>role=user_name[,...]</i></code>	You can specify either built-in roles or roles that you have defined, and you can assign multiple uses to each role. Use commas to separate multiple roles and users.
<code>-version <i>version</i></code>	Optionally, you can specify the version of the credentials file format, such as 18.0.0.0.0.

Usage Notes

- Only clusters running Oracle Grid Infrastructure 12c release 2 (12.2) or later can be configured and added as Fleet Patching and Provisioning Clients. Clusters running earlier versions of Oracle Grid Infrastructure, and servers running no Oracle Grid Infrastructure, can be managed directly by the Fleet Patching and Provisioning Server.
- You can only run this command on the Fleet Patching and Provisioning Server.

Examples

To add a client to the Fleet Patching and Provisioning Server:

```
$ rhpctl add client -client ClientCluster3 -toclientdata Grid_home/RHPserver/info
```

rhpctl delete client

Deletes a specific Fleet Patching and Provisioning Client from the configuration.

Syntax

```
rhpctl delete client -client cluster_name [-force]
```

Usage Notes

- Specify the name of the client cluster that you want to delete from the configuration.
- You must stop the Fleet Patching and Provisioning Client before you run this command or use the `-force` option.

Example

To delete the Fleet Patching and Provisioning Client `ClientCluster3`:

```
$ rhpctl delete client -client ClientCluster3
```

rhpctl discover client

Validates the input provided and discovers parameters on the given nodes, and generates a response file that you can use for configuring Oracle Clusterware. After completing this command, use [page F-10](#) to validate the response file and prepare the target nodes for Oracle Clusterware deployment.

Syntax

```
rhpctl discover client -image image_name -generatepath response_file_path
  {-responsefile response_file_name | -clusternodes node_list -client
cluster_name
  -oraclehome oracle_home_path} {-root | -sudouser sudo_username
  -sudopath sudo_binary_path | -cred cred_name | -auth plugin_name
  [-arg1 name1:value1...] [-user gi_user_name]
  [-scan scan_name]
```

Parameters

Table F-3 rhpctl discover client Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of the Oracle Grid Infrastructure Gold Image which the resulting response file will support.
-generatepath <i>response_file_path</i>	Specify a file path where the response file that RHPCTL generates will be copied. The RHPCTL command generates name of the response file and displays it while the command is running.
-responsefile <i>response_file_name</i>	If you have a partially complete response file and you want it to be completed with reference to the target nodes, then specify the response file name using this parameter. Note: The response file must include the node list, client name, and Oracle home path.
-clusternodes <i>node_list</i>	Specify a comma-delimited list of nodes on which you plan to provision Oracle Clusterware (using the resulting response file) in the following format: <i>node_name:node_vip[:node_role]</i> [, <i>node_name:node_vip[:node_role]</i> ...]
-client <i>cluster_name</i>	Specify the name of the target cluster to be probed.
-oraclehome <i>oracle_home_path</i>	Specify the location of the Oracle home.
-root -sudouser <i>sudo_username</i> - sudopath <i>sudo_binary_path</i> - cred <i>cred_name</i> - auth <i>plugin_name</i> [- arg1 <i>name1:value1...</i>]	You must choose either <code>sudo</code> or <code>root</code> to access the remote nodes. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
-user <i>gi_user_name</i>	Specify the name of the Oracle Grid Infrastructure installation user.

Table F-3 (Cont.) rhpctl discover client Command Parameters

Parameter	Description
<code>-scan <i>scan_name</i></code>	Specify the SCAN name.

rhpctl export client

Exports data from the repository on the Fleet Patching and Provisioning Server to a client data file.

Syntax

```
rhpctl export client -client cluster_name -clientdata file_path
```

Parameters

Table F-4 rhpctl export client Command Parameters

Parameter	Description
<code>-client <i>cluster_name</i></code>	Specify the name of the client cluster that you want to export.
<code>-clientdata <i>file_path</i></code>	Specify the path to the location of the client data file.

Usage Notes

You can only run this command on the Fleet Patching and Provisioning Server.

Example

To export repository data from a Fleet Patching and Provisioning Client named `mjk9394` to a client data file, `/tmp/mjk9394.xml`:

```
$ rhpctl export client -client mjk9394 -clientdata /tmp/mjk9394.xml
```

rhpctl modify client

Modifies a Fleet Patching and Provisioning Client.

Syntax

```
rhpctl modify client -client cluster_name [-enabled {TRUE | FALSE}]
  [-maproles role=user_name[+user_name...]
  [,role=user_name[+user_name...],...]] [-password]]
```

Parameters

Table F-5 rhpctl modify client Command Parameters

Parameter	Description
-client <i>cluster_name</i>	Specify the name of the client cluster that you want to modify.
-enabled {TRUE FALSE}	Specify whether the client is enabled.
-maproles <i>role=user_name[+user_name...][,...]</i>	You can modify either built-in roles or roles that you have defined, and you can assign multiple uses to each role. When you use the <code>-maproles</code> parameter, use a plus sign (+) to map more than one user to a specific role. Separate additional role/user pairs with commas.
-password	Optionally, you can specify a password to recreate the Fleet Patching and Provisioning Client credentials.

Example

To disable a Fleet Patching and Provisioning Client named `RHPClient001`:

```
$ rhpctl modify client -client RHPClient001 -enabled FALSE
```

rhpctl query client

Displays the configuration information of a specific Fleet Patching and Provisioning Client cluster.

Syntax

```
rhpctl query client [-client cluster_name[-detail [-node node_name] [-displayhtml]]]
  [-rhpserver rhps_regex]
```

Parameters

Table F-6 rhpctl query client Command Parameters

Parameter	Description
-client <i>cluster_name</i>	Specify the name of the client cluster in which the Fleet Patching and Provisioning Client resides.
-detail	Generate cluster information (HTML) for the cluster.

Table F-6 (Cont.) rhpctl query client Command Parameters

Parameter	Description
<code>-node <i>node_name</i></code>	Specify the name of a node in a remote cluster.
<code>-displayhtml</code>	Display the HTML cluster information for the cluster.
<code>-rhpserver <i>rhps_regex</i></code>	Specify a regular expression to match the cluster name of the servers where the operation must be executed.

Usage Notes

Specify the name of the client cluster in which the Fleet Patching and Provisioning Client resides for which you want to display the configuration information.

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles  
role=user_name[+user_name...][,role=user_name[+user_name...]]...
```

For information about granting roles with RHPCTL, refer to [rhpctl grant role](#)

Example

This command displays output similar to the following:

```
/rhpctl query client -client mbcluster-13  
Site: mbcluster-13  
Fleet Patching and Provisioning Client Version: 12.2.0.1.0  
Enabled: true  
Host from which RHPC last registered: rhpserver01.foo.com  
Port number last registered by RHPC: 8896  
RHP Enabled: true  
Standalone: false  
Managed: true
```

rhpctl update client

Updates an image on the Fleet Patching and Provisioning Client.

Syntax

```
rhctl update client -image image_name {-targetnode node_name
| -batches 'node_name'} -root
```

Parameters

Table F-7 rhctl update client Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of the image that you want to update.
-targetnode <i>node_name</i>	Specify the name of the node on which you want to update the Fleet Patching and Provisioning Client.
-batches ' <i>node_name</i> '	Alternative to specifying a target node, you can specify batches of nodes. Note: If you use this parameter for Oracle Database Appliance nodes, then run the command twice, in succession, specifying any one Oracle Database Appliance node for the first run, and another Oracle Database Appliance node for the second run.
-root	You must specify this parameter if you use either the -targetnode or -batches parameters.

Usage Notes

You can only run this command from a Fleet Patching and Provisioning Server.

Examples

The following example uses the -targetnode parameter:

```
$ rhctl update client -image ODA1 -targetnode rac07box1 -root
```

The two following examples use the -batches parameter:

```
$ rhctl update client -image ODA1 -batches '(rac07box1)' -root
$ rhctl update client -image ODA1 -batches '(rac07box2)' -root
```

rhctl verify client

Validates the input provided and creates or completes and verifies the values in a response file that you can use to configure Oracle Clusterware.

Syntax

```
rhctl verify client -image image_name -responsefile response_file_name
[-clusternodes node_list] {-root | -sudouser sudo_username -sudopath
sudo_binary_path | -cred cred_name} | -auth plugin_name [-arg1
name1:value1...]
[-user gi_user_name] [-client cluster_name] [-scan scan_name]
[-oraclehome oracle_home_path] [-ignorewarn] [-fixup [-setupSSH]]
```


Parameters

Table F-8 rhpctl verify client Command Parameters

Parameter	Description
<code>-image <i>image_name</i></code>	Specify the name of the image.
<code>-responsefile <i>response_file_name</i></code>	Specify a response file to be used to provision Oracle Grid Infrastructure.
<code>-clusternodes <i>node_list</i></code>	Specify a comma-delimited list of nodes on which Oracle Clusterware will be provisioned in the following format: <i>node_name:node_vip[:node_role]</i> [, <i>node_name:node_vip[:node_role]</i> ...]
<code>-root -sudouser <i>sudo_username</i> - sudopath <i>sudo_binary_path</i> - cred <i>cred_name</i> - auth <i>plugin_name</i> [- arg1 <i>name1:value1...</i>]</code>	You must choose either <code>sudo</code> or <code>root</code> to access the remote nodes. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
<code>-user <i>gi_user_name</i></code>	Specify the name of the Oracle Grid Infrastructure installation user.
<code>-client <i>cluster_name</i></code>	Specify the name of the cluster you want to verify.
<code>-scan <i>scan_name</i></code>	Specify the SCAN name.
<code>-oraclehome <i>oracle_home_path</i></code>	Specify the location of the Oracle home.
<code>-ignorewarn</code>	Use this parameter to ignore warnings during validation.
<code>-fixup [-setupSSH]</code>	Use this parameter to run a fixup script, which automatically applies changes to the nodes to satisfy changes that CVU recommends. Optionally, you can use the <code>-setupSSH</code> parameter to set up passwordless SSH user equivalence on the remote nodes for the provisioning user.

rhpctl add credentials

Adds credentials to the Oracle Cluster Registry (OCR).

Syntax

```
rhpctl add credentials -cred cred_name {-root | -sudouser sudo_user_name  
-sudopath sudo_binary_location}
```

Parameters

Table F-9 rhpctl add credentials Command Parameters

Parameter	Description
<code>-cred cred_name</code>	Specify a credential name to associate the user and password credentials to access a remote node.
<code>-root</code> <code>-sudouser sudo_user_name - sudopath sudo_binary_location</code>	You must choose either to provide <code>root</code> access to access a remote node or a <code>sudo</code> user name and path to the <code>sudo</code> binary to perform super user operations.

rhpctl delete credentials

Deletes credentials from the Oracle Cluster Registry (OCR).

Syntax

```
rhpctl delete credentials -cred cred_name
```

Usage Notes

Specify only the name of the credentials you want to delete.

rhpctl add database

Creates a database using a specific working copy.

Syntax

```
rhpctl add database -workingcopy workingcopy_name -dbname unique_db_name
[-datafileDestination datafileDestination_path] {-node node_list | -
serverpool pool_name
[-pqpool pool_name | -newpqpool pool_name -pqcardinality cardinality]
| -newpool pool_name -cardinality cardinality [-pqpool pool_name
| -newpqpool pool_name -pqcardinality cardinality]} [-dbtype
{RACONENODE | RAC | SINGLE}] [-dbtemplate file_path |
image_name:relative_file_path]
[-cdb] [-pdbName pdb_prefix [-numberOfPDBs pdb_count]]
[{-sudouser user_name -sudopath sudo_binary_path | -root | -cred
cred_name
| -auth plugin_name [-arg1 name1:value1...]}] [-targetnode node_name]
[-useractiondata user_action_data] [-schedule timer_value] [-eval]
```

Parameters

Table F-10 rhpctl add database Command Parameters

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify the name of an existing working copy for the database that you want to add.
-dbname <i>unique_db_name</i>	Specify the unique name of the database (DB_UNIQUE_NAME without DB_DOMAIN) that you are adding.
-datafileDestination <i>datafileDestination_path</i>	Specify the data file destination location or the name of the Oracle Automatic Storage Management (Oracle ASM) disk group. Note: You cannot specify a disk group for Oracle Database versions before Oracle Database 11g release 2 (11.2).
-node <i>node_list</i>	Specify a node or comma-delimited list of several nodes on which to create the database.
-serverpool <i>server_pool_name</i>	Specify the name of an existing server pool.
-pqpool <i>server_pool_name</i>	Specify the name of an existing server pool. Note: This parameter is only applicable in an Oracle Flex Cluster environment and refers to server pools (either already defined, as in this case, or to be created when you use the -newpqpool parameter) running on non-Hub Nodes.
-newpqpool <i>server_pool_name</i>	Optionally, you can create a new server pool to be used for parallel queries. Specify a name for the new server pool. Note: This parameter is only applicable in an Oracle Flex Cluster environment because it refers to server pools running on non-Hub Nodes.
-pqcardinality <i>cardinality</i>	If you create a new server pool, then you must specify a cardinality value for the server pool. Note: This parameter is only applicable in an Oracle Flex Cluster environment.
-newpool <i>server_pool_name</i>	Optionally, you can create a new server pool. Specify a name for the new server pool.
-cardinality <i>cardinality</i>	If you create a new server pool, then you must specify a cardinality value for the server pool.
-dbtype {RACONENODE RAC SINGLE}	Specify whether the database is Oracle RAC One Node, Oracle RAC, or a nonclustered database.
-dbtemplate <i>file_path</i> <i>image_name:relative_file_path</i>	Specify the absolute file path to a database template or the relative path to the image home directory on a Fleet Patching and Provisioning Server.
-cdb	Optionally, use this parameter to create a database as a container database.
-pdbName <i>pdb_prefix</i>	If you are creating one or more pluggable databases, then specify a pluggable database name prefix.
-numberOfPDBs <i>pdb_count</i>	Specify the number of pluggable databases you want to create.

Table F-10 (Cont.) rhpctl add database Command Parameters

Parameter	Description
<code>-sudouser user_name -sudopath sudo_binary_path -root -cred cred_name -auth plugin_name [-arg1 name1:value1...]</code>	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>sudo</code> or <code>root</code> to access the remote node. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
<code>-targetnode node_name</code>	Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client.
<code>-useractiondata user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-schedule timer_value</code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05
<code>-eval</code>	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.

Usage Notes

If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Examples

To create a database on a working copy named `prodhome`:

```
$ rhpctl add database -workingcopy prodhome -dbname proddb -
datafileDestination /acfs/proddata -serverpool prodpool1 -dbtype RAC
```

**Note:**

You can create multiple databases on a working copy.

rhpctl addnode database

Adds instances to an administrator-managed Oracle RAC database.

Syntax

```
rhpctl addnode database -workingcopy workingcopy_name
  -dbname unique_db_name -node node_list
  [-root | -cred cred_name | -sudouser sudo_user_name
```

```
-sudopath sudo_binary_location |
-auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
[-useractiondata user_action_data] [-eval] [-schedule timer_value]
```

Parameters

Table F-11 rhpctl addnode database Command Parameters

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify the name of a working copy.
-dbname <i>unique_db_name</i>	Specify the unique name of the database (DB_UNIQUE_NAME without DB_DOMAIN) that you are adding.
-node <i>node_list</i>	Specify a node or comma-delimited list of several nodes on which to create the database.
-root -cred <i>cred_name</i> - sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_location</i> -auth <i>plugin_name</i> <i>plugin_args</i>	If you choose to use the -targetnode parameter, then you must choose either root, a credential name, sudo, or an authentication plugin to access the remote node. Choose -root to perform super user operations as root. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a sudo user by specifying a sudo user name and the path to the sudo binary, or to use an authentication plugin to access the remote node.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the useractiondata parameter of the user action script.
-eval	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.
-schedule <i>timer_value</i>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05

Usage Notes

- If the specified working copy is not installed on the nodes in the node list, then you must first run `rhpctl addnode workingcopy`.
- If the working copy is on a Fleet Patching and Provisioning Client or on the Fleet Patching and Provisioning Server, then credentials are not required. This is true whether you run the command on the Server or the Client. Credentials are required when you run the command on the Server and the working copy is on a target that is not a Fleet Patching and Provisioning Client.
- If you choose to use the -schedule parameter, then you must run this command on the Fleet Patching and Provisioning Server.

rhpcctl addpdb database

Adds a pluggable database to an existing container database on a working copy. After you create a working copy of a gold image and provision that working copy to a target, and create a database as a multitenant container database, you can add a pluggable database to the container database on the working copy.

Syntax

```
rhpcctl addpdb database -workingcopy workingcopy_name -cdbName cdb_name
  -pdbName new_pdb_name [-pdbDatafileDestination
pdb_datafile_destination_path]
  [-root | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]
  | -sudouser user_name -sudopath sudo_binary_path] [-targetnode
node_name]
  [-useractiondata user_action_data] [-schedule timer_value]
```

Parameters

Table F-12 rhpcctl addpdb database Command Parameters

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify the name of an existing working copy for the pluggable database that you want to add.
-cdbName <i>cdb_name</i>	Specify the name of the multitenant container database to which you want to add the pluggable database.
-pdbName <i>new_pdb_name</i>	Specify a name for the pluggable database you are adding.
- pdbDatafileDestinati on <i>pdb_datafile_destina tion</i>	Optionally, you can specify the path to the data file destination location for the pluggable database.
-root -cred <i>cred_name</i> -auth <i>plugin_name</i> [-arg1 <i>name1:value1...</i>] - sudouser <i>user_name</i> - sudopath <i>sudo_binary_path</i>	<p>If you choose to use the -targetnode parameter, then you must choose either sudo or root to access the remote node.</p> <p>If you choose sudo, then you must specify a user name to run super-user operations, and a path to the location of the sudo binary.</p> <p>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.</p> <p>Alternative to -sudouser, -root, or -cred, you can use -auth to specify an authentication plugin to access a remote node.</p>
-targetnode <i>node_name</i>	Optionally, you can specify the name of a node in the cluster on which you want to run this operation.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the useractiondata parameter of the user action script.

Table F-12 (Cont.) rhpctl addpdb database Command Parameters

Parameter	Description
<code>-schedule</code> <code>timer_value</code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05

Usage Notes

The working copy can be on Fleet Patching and Provisioning Server, a Fleet Patching and Provisioning Client, or a non-Fleet Patching and Provisioning Client target.

Example

The following example creates a pluggable database called `pdb183` on a container database called `raccdb183` on a working copy called `wc_db183`:

```
$ rhpctl addpdb database -workingcopy wc_db183 -cdbName raccdb183 -pdbName
pdb183
```

rhpctl deletepdb database

Deletes a pluggable database to an existing container database on a working copy.

Syntax

```
rhpctl deletepdb database -workingcopy workingcopy_name -cdbName cdb_name
-pdbName pdb_name
[-root | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]
| -sudouser user_name -sudopath sudo_binary_path] [-targetnode
node_name]
[-useractiondata user_action_data] [-schedule timer_value]
```

Parameters**Table F-13 rhpctl deletepdb database Command Parameters**

Parameter	Description
<code>-workingcopy</code> <code>workingcopy_name</code>	Specify the name of an existing working copy for the pluggable database that you want to delete.
<code>-cdbName</code> <code>cdb_name</code>	Specify the name of the multitenant container database from which you want to delete the pluggable database.
<code>-pdbName</code> <code>pdb_name</code>	Specify the name of the pluggable database you want to delete.

Table F-13 (Cont.) rhpctl deletepdb database Command Parameters

Parameter	Description
<code>-root</code> <code>-cred</code> <code>cred_name</code> <code>-auth</code> <code>plugin_name</code> [-arg1 <code>name1:value1...</code>] <code>-</code> <code>sudouser user_name</code> - <code>sudopath</code> <code>sudo_binary_path</code>	<p>If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>sudo</code> or <code>root</code> to access the remote node.</p> <p>If you choose <code>sudo</code>, then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary.</p> <p>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.</p> <p>Alternative to <code>-sudouser</code>, <code>-root</code>, or <code>-cred</code>, you can use <code>-auth</code> to specify an authentication plugin to access a remote node.</p>
<code>-targetnode</code> <code>node_name</code>	Optionally, you can specify the name of a node in the cluster on which you want to run this operation.
<code>-useractiondata</code> <code>user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-schedule</code> <code>timer_value</code>	<p>Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:</p> <p>2018-07-25T19:13:17+05</p>

Usage Notes

The working copy can be on Fleet Patching and Provisioning Server, a Fleet Patching and Provisioning Client, or a non-Fleet Patching and Provisioning Client target.

Examples

The following example deletes a pluggable database called `pdb183` from a container database called `raccdb183` on a working copy called `wc_db183`:

```
$ rhpctl deletepdb database -workingcopy wc_db183 -cdbName raccdb183 -
pdbName pdb183
```

rhpctl delete database

Deletes a database that was created on a working copy.

Note:

If the database is hosted on a working copy that is on the Fleet Patching and Provisioning Server or on a Fleet Patching and Provisioning Client, then credentials are not required. This is true whether you run the command on the Server or the Client. Credentials are required when you run the command on the Server and the working copy is on a target that is not a Fleet Patching and Provisioning Client.

Syntax

```

rhpctl delete database -workingcopy workingcopy_name -dbname unique_db_name
  {-sudouser sudo_user_name -sudopath sudo_binary_path | -root | -cred
cred_name
  | -auth plugin_name [-arg1 name1:value1...]} [-targetnode node_name]
  [-useractiondata user_action_data]

```

Parameters

Table F-14 rhpctl delete database Command Parameters

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify a name for the working copy for the database that you want to delete.
-dbname <i>unique_db_name</i>	Specify the unique name of the database (DB_UNIQUE_NAME without DB_DOMAIN) that you are deleting.
-sudouser <i>user_name</i> -sudopath <i>sudo_binary_path</i> - root -cred <i>cred_name</i> -auth <i>plugin_name</i> [-arg1 <i>name1:value1...</i>]	<p>If you choose to use the -targetnode parameter, then you must choose either sudo or root to access the remote node.</p> <p>If you choose sudo, then you must specify a user name to run super-user operations, and a path to the location of the sudo binary.</p> <p>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.</p> <p>Alternative to -sudouser, -root, or -cred, you can use -auth to specify an authentication plugin to access a remote node.</p>
-targetnode <i>node_name</i>	Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the useractiondata parameter of the user action script.

rhpctl deletenode database

Deletes an instance, which contracts an administrator-managed Oracle RAC database.

Syntax

```

rhpctl deletenode database -workingcopy working_copy_name -dbname
unique_db_name
  -node node_list {-root | -sudouser sudo_username -sudopath
sudo_binary_path
  | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]} [-force]
  [-failover] [-drain_timeout timeout] [-stopoption stop_option]
  [-useractiondata user_action_data] [-schedule timer_value] [-eval]

```

Parameters

Table F-15 rhpctl deletenode database Command Parameters

Parameter	Description
<code>-workingcopy</code> <code>working_copy_name</code>	Specify the name of a working copy.
<code>-dbname</code> <code>unique_db_name</code>	Specify the unique name of the database (DB_UNIQUE_NAME without DB_DOMAIN) that you are deleting.
<code>-node node_list</code>	Specify a node or comma-delimited list of several nodes from which to delete the database instance.
<code>-root</code> <code>-sudouser</code> <code>sudo_username</code> - <code>sudopath</code> <code>sudo_binary_path</code> - <code>cred cred_name</code> - <code>auth plugin_name</code> [- <code>arg1</code> <code>name1:value1...</code>]	Choose either <code>sudo</code> or <code>root</code> to access the remote nodes. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
<code>-force</code>	Use <code>-force</code> to remove the instance after forcibly stopping the instance.
<code>-failover</code>	Optionally, you can use this parameter to attempt to have services running on the instance that want to delete fail over to another instance.
<code>-drain_timeout</code> <code>timeout</code>	Optionally, you can use <code>-drain_timeout</code> to specify the time, in seconds, allowed for resource draining to be completed. Accepted values are an empty string (""), 0, or any positive integer. The default value is an empty string, which means that this parameter is not set. If it is set to 0, then draining occurs, immediately. The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted.
<code>-stopoption</code> <code>stop_option</code>	Optionally, you can specify a stop option for the database. Options include: ABORT, IMMEDIATE, NORMAL, TRANSACTIONAL, and TRANSACTIONAL_LOCAL.
<code>-useractiondata</code> <code>user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-schedule</code> <code>timer_value</code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05
<code>-eval</code>	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.

Usage Notes

- If the working copy is on a Fleet Patching and Provisioning Client or on the Fleet Patching and Provisioning Server, then credentials are not required. This is true whether you run the command on the Server or the Client. Credentials are required when you run the command on the Server and the working copy is on a target that is not a Fleet Patching and Provisioning Client.
- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

rhpcctl move database

Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.

Syntax

```
rhpcctl move database -patchedwc workingcopy_name {{-sourcewc workingcopy_name |
  -sourcehome Oracle_home_path [-oraclebase Oracle_base_path]
  [-client cluster_name]}
  [-dbname db_name_list | -excludedblist db_name_list]
  [-nonrolling | -forcerolling | -batches list_of_batches | -smartmove [-saf availability]
  [-separate]] [-eval] [schedule timer_value] [-ignorewcpatches] [-keepplacement]
  [-disconnect [-noreplay]] [-drain_timeout time] [-stopoption stop_option]
  [-nodatapatch] [-targetnode node_name] [-notify [-cc user_list]] |
  -continue [-skip] | -revert | -abort}
  [-root | -cred cred_name | -sudouser sudo_user_name
  -sudopath sudo_binary_location |
  -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
  [-useractiondata user_action_data] [-schedule timer_value]
```

Parameters

Table F-16 rhpcctl move database Command Parameters

Parameter	Description
<code>-patchedwc</code> <i>workingcopy_name</i>	Specify the name of the working copy to where you want to move the database.
<code>-sourcewc</code> <i>workingcopy_name</i>	Specify the name of the working copy from which the database is to be moved.
<code>-sourcehome</code> <i>Oracle_home_path</i>	Alternatively, you can specify the source Oracle home path.
<code>-oraclebase</code> <i>Oracle_base_path</i>	Specify the ORACLE_BASE path for provisioning the Oracle database home (required only for ORACLEDBSOFTWARE image type).
<code>-client</code> <i>cluster_name</i>	Specify the name of the client cluster.

Table F-16 (Cont.) rhpctl move database Command Parameters

Parameter	Description
-dbname <i>db_name_list</i>	Specify the unique names of the databases (DB_UNIQUE_NAME without DB_DOMAIN) that you want to move to the patched working copy. Note: If you are moving a non-clustered (single-instance) database, then, for the value of the -dbname parameter, you must specify the SID of the database instead of the database name.
-excludedblist <i>db_name_list</i>	Alternative to using the -dbname parameter, you can use the -excludedblist parameter to patch all databases except specific databases.
-nonrolling - forcerolling -batches <i>list_of_batches</i> - smartmove [-saf <i>availability</i>] [- separate]	Optionally, you can choose one of the three following methods to move a database: <ul style="list-style-type: none"> • Use the -nonrolling parameter to move the database in a non-rolling mode. By default, databases move in a rolling mode. • Use the -forcerolling parameter to force the Oracle home to move in rolling mode. • Use the -batches parameter to specify a comma-delimited list of batches of nodes (where each batch is a comma-delimited list of node names enclosed in parentheses) enclosed in double quotation marks (" ") in the format: "(nA,nB,...),(...,nY,nZ)". • Alternatively, use the -smartmove parameter. Use the -saf <i>availability</i> parameter to specify a service availability factor, which is the minimum percentage of instances on which a service must remain running during the move. <p>Use the -separate parameter to process batches separately. When you use this parameter, the move command returns after each batch. The move operation for the first batch must specify the source home and other parameters that apply to all batches (such as -nonrolling and -keepplacement). Control subsequent batches using the -continue, -skip, and -abort parameters.</p>
-eval	Use the -eval parameter to print auto-generated batches of nodes and sequence of moves without actually performing the move operation.
-ignorewcpatches	Optionally, you can use this parameter to ignore if a patched working copy is missing some patches which are present in the source path or working copy.
-keepplacement	Use this parameter to ensure that services of administrator-managed Oracle RAC or Oracle RAC One Node databases are running on the same instances before and after the move operation.
-disconnect [-noreplay]	Optionally, use the -disconnect parameter to disconnect all sessions before stopping or relocating services. If you choose to use -disconnect, then you can choose to use the -noreplay parameter to disable session replay during disconnection.
-drain_timeout <i>timeout</i>	Specify the time, in seconds, allowed for resource draining to be completed. Accepted values are an empty string (""), 0, or any positive integer. The default value is an empty string, which means that this parameter is not set. If it is set to 0, then draining occurs, immediately. The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted.
-stopoption <i>stop_option</i>	Optionally, you can specify a stop option for the database. Options include: ABORT, IMMEDIATE, NORMAL, TRANSACTIONAL, and TRANSACTIONAL_LOCAL.

Table F-16 (Cont.) rhpctl move database Command Parameters

Parameter	Description
<code>-nodatapatch</code>	Use this parameter to indicate not to run <code>datapatch</code> for databases you are moving.
<code>-targetnode node_name</code>	Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client.
<code>-notify [-cc user_list]</code>	Optionally, you can supply a list of users to whom email notifications of the move will be sent, in addition to the owner of the working copy.
<code>-continue [-skip]</code>	If a batch-mode <code>rhpctl move database</code> command fails at any point, then, after correcting the cause of the error, you can rerun the command with the <code>-continue</code> parameter to attempt to patch the failed batch. If you want to skip the failed batch and continue with the next batch, use the <code>-continue</code> and <code>-skip</code> parameters together. If you attempt to skip over the last batch, then the move operation is terminated.
<code>-revert</code>	If a batch-mode or non-batch-mode <code>rhpctl move database</code> command fails, then you can rerun the command with the <code>-revert</code> parameter to undo the changes that have been made, and return the configuration to its initial state.
<code>-abort</code>	If a batch-mode or non-batch-mode <code>rhpctl move database</code> command fails, then you can rerun the command with the <code>-abort</code> parameter to terminate the patching process and leave the cluster in its current state.
<code>-root -cred cred_name -sudouser sudo_user_name -sudopath sudo_binary_location -auth plugin_name plugin_args</code>	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>root</code> , a credential name, <code>sudo</code> , or an authentication plugin to access the remote node. Choose <code>-root</code> to perform super user operations as <code>root</code> . Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a <code>sudo</code> user by specifying a <code>sudo</code> user name and the path to the <code>sudo</code> binary, or to use an authentication plugin to access the remote node.
<code>-useractiondata user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-schedule timer_value</code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format. For example: <code>2018-07-25T19:13:17+05</code>

Usage Notes

- You can obtain context sensitive help for specific use cases for the `rhpctl move database` command, as follows:

```
$ rhpctl move database -help [EXISTING_PATCHEDWC | NEW_PATCHEDWC |
SRCHOME
| SINGLEINSTANCEDB | ROLLING | NONROLLING | BATCHES | SMARTMOVE]
```

- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Examples

To move all the databases running from one working copy to another in a rolling fashion:

```
$ rhpctl move database -sourcewc prodHomeV1 -patchedwc prodHomeV2 -client prodcluster
```

In the preceding example, the patched working copy, `prodHomeV2`, must exist.

To move all databases running on a non-managed Oracle home at `/u01/app/product/12.1.0/dbhome` to a working copy named `myDB12Home1`:

```
$ rhpctl move database -sourcehome /u01/app/product/12.1.0/dbhome -oraclebase /u01/app/product/12.1.0/obase -patchedwc myDB12Home1
```

To move a database named `SampleDB` from a working copy named `myDB12Home1` to a working copy named `myDB12Home1patched` (any other databases running on `myDB12Home1` are not affected by this move):

```
$ rhpctl move database -sourcewc myDB12Home1 -patchedwc myDB12Home1patched -dbname SampleDB
```

To move all databases running on a working copy named `myDB12Home1` to a working copy named `myDB12Home1patched`:

```
$ rhpctl move database -sourcewc myDB12Home1 -patchedwc myDB12Home1patched
```

To move a non-clustered (single-instance) database with a SID of `SID101` running on a patched working copy named `myDB12Home1patched`:

```
$ rhpctl move database -patchedwc myDB12Home1patched -sourcehome /u01/app/oracle/product/12.2.0/db_1 -targetnode vm02 -dbname SID101 -sudouser oracle -sudopath /usr/bin/sudo
```

The preceding examples are the basic form of the command. You can also move groups of databases in batches. The batch operations also support management of session connections and recovery options.

rhpctl movepdb database

Moves one or more pluggable databases from a source, single-instance container database, to a destination, single-instance container database.

The source and destination single-instance container databases can be running in a provisioned database working copy. The working copy can be on the Fleet Patching and Provisioning Server, a Fleet Patching and Provisioning Client, or a non-Fleet Patching and Provisioning Client target, which is a target without a Fleet Patching and Provisioning Client configured and running. The destination single-instance container database can be at a higher patch level, which facilitates patching of a pluggable database to a higher patch level.

Syntax

```

rhpctl movepdb database -sourcecdb source_cdb_name -destcdb
destination_cdb_name
  {-pdbname pdb_name_list | -excludepdblist pdb_name_list}
  [-root | -cred cred_name | -sudouser sudo_user_name
  -sudopath sudo_binary_location |
  -auth plugin_name [-arg1 name1:value1
  [-arg2 name2:value2 ...]]] [-client client_name | -targetnode
node_name]
  [-useractiondata user_action_data] [-schedule timer_value]

```

Parameters

Table F-17 rhpctl movepdb database Command Parameters

Parameter	Description
-sourcecdb <i>source_cdb_name</i>	Specify the name of the Oracle Multitenant container database from which you want to move the pluggable database.
-destcdb <i>destination_cdb_name</i>	Specify the name of the multitenant container database to which you want to move the pluggable database.
-pdbname <i>pdb_name_list</i>	Specify a comma-separated list of names of pluggable databases that you want to move.
-excludepdblist <i>pdb_name_list</i>	Specify a list of pluggable databases that you want to excluded from the move operation.
-root -cred <i>cred_name</i> -sudouser <i>sudo_user_name</i> -sudopath <i>sudo_binary_location</i> -auth <i>plugin_name</i> <i>plugin_args</i>	<p>If you choose to use the -targetnode parameter, then you must choose either root, a credential name, sudo, or an authentication plugin to access the remote node.</p> <p>Choose -root to perform super user operations as root. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a sudo user by specifying a sudo user name and the path to the sudo binary, or to use an authentication plugin to access the remote node.</p>
-client <i>client_name</i> -targetnode <i>node_name</i>	Optionally, you can specify either the name of the client cluster or the node on which the operation is to be run.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the useractiondata parameter of the user action script.

Table F-17 (Cont.) rhpctl movepdb database Command Parameters

Parameter	Description
<code>-schedule timer_value NOW</code>	<p>Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:</p> <p>2019-01-07T19:13:17+05</p> <p>You can also specify <code>NOW</code> to schedule the operation, immediately.</p>

Usage Notes

You can only use this command if both the source and destination single-instance container databases are on the same node.

Examples

To move a pluggable database from a source single-instance container database to a destination single-instance container database:

```
rhpctl movepdb database -sourcecdb srccdb -pdbname pdb1,pdb2,pdb3 -destcdb
dstcdb
```

rhpctl upgrade database

Upgrades a database to the version of the destination working copy.

Syntax

```
rhpctl upgrade database [-sourcewc source_workingcopy_name | -sourcehome
oracle_home_path
  [-oraclebase Oracle_base_path] [{-client cluster_name | -targetnode
node_name}]]
  [-root | -cred cred_name | -sudouser sudo_username -sudopath
path_to_sudo_binary
  | -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
  -destwc destination_workingcopy_name [-image image_name [-path
where_path]]
  -dbname unique_db_name [-useractiondata user_action_data] [-eval [-
preupg]
  [-schedule timer_value]
  [-ignoremissingpatches patch_name1[,patch_name2...]]
  [-dbuaargs dbua_arguments]
  [-autoupg [-upgtimezone { YES | NO }] [-runutlrp { YES | NO }] [-fra
db_recovery_file_dest] -ignoregroupcheck ]
```


Parameters

Table F-18 rhpctl upgrade database Command Parameters

Parameter	Description
<code>-sourcewc</code> <code>source_workingcopy_name</code>	Specify the name of the source working copy from which you want to upgrade the database.
<code>-sourcehome</code> <code>oracle_home_path</code>	Alternative to specifying the name of the source working copy, you can specify the path to the source Oracle home.
<code>-oraclebase</code> <code>oraclebase_path</code>	If you use the <code>-sourcehome</code> parameter, then you can, optionally, specify a different ORACLE_BASE from the source Home.
<code>-client cluster_name</code> <code>-targetnode node_name</code>	Specify either the name of the client cluster or the name of a node in a remote cluster with no Fleet Patching and Provisioning Client on which to provision a working copy.
<code>-root</code> <code>-cred cred_name</code> <code>-sudouser</code> <code>sudo_user_name</code> <code>-sudopath</code> <code>sudo_binary_location</code> <code>-auth plugin_name</code> <code>plugin_args</code>	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>root</code> , a credential name, <code>sudo</code> , or an authentication plugin to access the remote node. Choose <code>-root</code> to perform super user operations as <code>root</code> . Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a <code>sudo</code> user by specifying a <code>sudo</code> user name and the path to the <code>sudo</code> binary, or to use an authentication plugin to access the remote node.
<code>-destwc</code> <code>destination_workingcopy_name</code> [<code>-image image_name</code> [<code>-path where_path</code>]]	Specify the name of the destination working copy to which the database is to be upgraded. If the destination working copy does not exist, then specify the gold image from which to create it, and optionally, the path to where to provision the working copy.
<code>-dbname unique_db_name</code>	Specify the name of the database you are upgrading.
<code>-useractiondata</code> <code>user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-eval</code>	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.
<code>-schedule timer_value</code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05
<code>-ignoremissingpatches</code>	Executes the move and/or upgrade although the specified patches, which are present in the source path or working copy, could be missing from the destination path or working copy.
<code>-dbuaargs dbua_arguments</code>	Species a double-quoted string containing optional arguments for DBUA.
<code>-autoupg</code>	Executes the upgrade database operation using the AutoUpgrade toolkit.
<code>-upgtimezone { YES NO }</code>	Enables or disables time zone upgrade as part of the AutoUpgrade process. Default is YES.
<code>-runutlrp { YES NO }</code>	Enables or disables the recompilation of invalid objects as part of the AutoUpgrade process. Default is YES.
<code>-fra</code> <code>db_recovery_file_dest</code>	Specifies the location of the flash recovery area.
<code>-ignoregroupcheck</code>	Skips the group check except for OSDBA and OSASM during the upgrade database process.

Usage Notes

If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Example

The following example upgrades a database, `testy`, from Oracle Database 11g, which is on working copy `db112mbc143` to Oracle Database 12c, which is on working copy `db12102mbc143`, both of which reside on the remote node `bposvr141`:

```
$ rhpctl upgrade database -dbname testy -sourcewc db112mbc143 -destwc
db12102mbc143 -root -targetnode bposvr141
```

rhpctl zdtupgrade database

Enables zero downtime database upgrade for Oracle RAC and Oracle RAC One Node databases.

Syntax

```
rhpctl zdtupgrade database -dbname unique_db_name -destwc
destination_workingcopy_name
  [-sourcewc source_workingcopy_name | -sourcehome oracle_home_path]
  -ggsrwc golden_gate_source_workingcopy_name
  -ggdstwc golden_gate_dest_workingcopy_name
  [-clonedatadg diskgroup_name [-cloneredodg diskgroup_name]
  [-clonerecodg diskgroup_name]]
  [-rmanlocation backup_location]
  [-targetnode node_name
  {-root | -cred cred_name | -sudouser sudo_user_name
  -sudopath sudo_binary_location |
  -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]}]
  [-useractiondata user_action_data]
```

Parameters

Table F-19 rhpctl zdtupgrade database Command Parameters

Parameter	Description
<code>-dbname</code> <i>unique_db_name</i>	Specify the unique name of the database that you want to upgrade.
<code>-destwc</code> <i>destination_workingc</i> <i>opy_name</i>	Specify the name of the destination working copy to which the database is to be upgraded.
<code>-sourcewc</code> <i>source_workingcopy_n</i> <i>ame</i>	Optionally, you can specify the name of the source working copy from which you want to upgrade the database.
<code>-sourcehome</code> <i>oracle_home_path</i>	Alternative to specifying the name of the source working copy, you can specify the path to the source Oracle home.

Table F-19 (Cont.) rhpctl zdtupgrade database Command Parameters

Parameter	Description
<code>-ggsrwc</code> <code>golden_gate_source_workingcopy_name</code>	Specify the name of the Oracle GoldenGate source working copy.
<code>-ggdstwc</code> <code>golden_gate_dest_workingcopy_name</code>	Specify the name of the Oracle GoldenGate destination working copy.
<code>-clonedatadg</code> <code>diskgroup_name</code>	Optionally, you can specify the name of an Oracle ASM disk group to use as a data file location for the cloned database.
<code>-cloneredodg</code> <code>diskgroup_name</code>	Optionally, you can specify the name of an Oracle ASM disk group to use as a redo log location for the clones database.
<code>-clonerecodg</code> <code>diskgroup_name</code>	Optionally, you can specify the name of an Oracle ASM disk group to use as a recovery area for the cloned database.
<code>-rmanlocation</code> <code>backup_location</code>	Optionally, you can specify the source RMAN backup location.
<code>-targetnode</code> <code>node_name</code>	Optionally, you can specify the name of a node in a remote cluster with no Fleet Patching and Provisioning Client on which to provision a working copy.
<code>-root</code> <code>-cred cred_name</code> <code>-sudouser sudo_user_name</code> <code>-sudopath sudo_binary_location</code> <code>-auth plugin_name plugin_args</code>	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>root</code> , a credential name, <code>sudo</code> , or an authentication plugin to access the remote node. Choose <code>-root</code> to perform super user operations as <code>root</code> . Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a <code>sudo</code> user by specifying a <code>sudo</code> user name and the path to the <code>sudo</code> binary, or to use an authentication plugin to access the remote node.
<code>-useractiondata</code> <code>user_action_data</code>	Optionally, you can use this parameter to specify a value to be passed to the <code>useractiondata</code> parameter of a <code>useraction</code> script

rhpctl addnode gihome

Adds one or more nodes to an Oracle Grid Infrastructure installation.

Syntax

```

rhpctl addnode gihome {-workingcopy workingcopy_name | -client
cluster_name}
    -newnodes node_name:node_vip[:node_role]
[,node_name:node_vip[:node_role]...]
    {-root | -cred cred_name | -sudouser sudo_user_name
    -sudopath sudo_binary_location |
    -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]}
    [-targetnode node_name] [-force] [-setupssh] [-useractiondata
user_action_data]
    [-eval] [-schedule timer_value]

```

Parameters

Table F-20 rhpctl addnode gihome Command Parameters

Parameter	Description
<code>-workingcopy</code> <code>workingcopy_name</code>	Specify the name of the working copy of the active Oracle Grid Infrastructure home that you want to install and configure on the specified node.
<code>-client cluster_name</code>	Alternatively, you can specify the name of the client cluster to which to add cluster nodes.
<code>-node node_list</code>	Specify a comma-delimited list of nodes on which Oracle Clusterware will be provisioned in the following format: <code>node_name:node_vip[:node_role]</code> <code>[,node_name:node_vip[:node_role]...]</code>
<code>-root</code> <code>-cred cred_name</code> <code>-sudouser sudo_user_name</code> - <code>-sudopath sudo_binary_location</code> <code>-auth plugin_name plugin_args</code>	You must choose either <code>root</code> , a credential name, <code>sudo</code> , or an authentication plugin to access the remote node. Choose <code>-root</code> to perform super user operations as <code>root</code> . Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a <code>sudo</code> user by specifying a <code>sudo</code> user name and the path to the <code>sudo</code> binary, or to use an authentication plugin to access the remote node.
<code>-targetnode node_name</code>	Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client.
<code>-force</code>	Optionally, you can use this parameter to forcibly add nodes ignoring any previously failed add-node operation.
<code>-setupssh</code>	Sets up passwordless SSH user equivalence on the remote nodes for the provisioning user.
<code>-useractiondata user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-eval</code>	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.
<code>-schedule timer_value</code>	Optionally, you can schedule a time to run this command in ISO-8601 format. For example: 2018-01-21T19:13:17+05.

Usage Notes

- You can specify the target for the operation using the working copy name or, if the target is a Fleet Patching and Provisioning Client, then using the client cluster name.
- You must provide either `root` credentials, a credential name, a `sudo` user, or an authentication plugin.
- A target node is required if the target cluster is an Oracle Clusterware 11g release 2 (11.2) or 12c release 1 (12.1) cluster and must be the node name of an existing cluster node.

rhpcctl deletenode gihome

Removes one or more nodes from an Oracle Grid Infrastructure installation.

Syntax

```
rhpcctl deletenode gihome {-workingcopy workingcopy_name | -client
cluster_name}
  -node node_list {-root | -sudouser sudo_username -sudopath
sudo_binary_path
  -cred cred_name | -auth plugin_name [-arg1 name1:value1...]}
  [-targetnode node_name] [-useractiondata user_action_data] [-eval]
```

Parameters

Table F-21 rhpcctl deletenode gihome Command Parameters

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify the name of a working copy of the Oracle Grid Infrastructure home that you want to remove from the specified node.
-client <i>cluster_name</i>	Alternatively, you can specify the name of the client cluster from which to remove cluster nodes.
-node <i>node_list</i>	Specify a comma-delimited list of node names from which to delete Oracle Grid Infrastructure.
-root -sudouser <i>sudo_username</i> - sudopath <i>sudo_binary_path</i> - cred <i>cred_name</i> - auth <i>plugin_name</i> [- arg1 <i>name1:value1...</i>]	You must choose either <code>sudo</code> or <code>root</code> to access the remote nodes. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
-targetnode <i>node_name</i>	Name of a node in a remote cluster with no Fleet Patching and Provisioning Client.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
-eval	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.

Usage Notes

- You can specify the target for the operation using the working copy name or, if the target is a Fleet Patching and Provisioning Client, then using the client cluster name.
- You must provide either `root` credentials or a `sudo` user.

- A target node is required if the target cluster is an Oracle Clusterware 11g release 2 (11.2) or 12c release 1 (12.1) cluster and must be the node name of an existing cluster node.

rhpcctl move gihome

Moves the Oracle Grid Infrastructure software stack from one home to another.

Syntax

```
rhpcctl move gihome {-destwc destination_workingcopy_name |
  -desthome destination_oracle_home_path}
  {{-sourcewc source_workingcopy_name | -sourcehome oracle_home_path}
  [-targetnode target_node_name] [-ignorewcpatches] [-nonrolling]
  [-keepplacement] [-auto -dbhomes mapping_of_Oracle_homes [-dblist
  db_name_list
  | -excludedblist db_name_list] [-nodatapatch] [-disconnect]
  [-stopoption stop_option] [-drain_timeout timeout]]
  [-batches list_of_batches | -smartmove [-saf availability]]
  [-schedule timer_value] | -continue | -revert | -abort}
  [-root | -cred cred_name | -sudouser sudo_username -sudopath
  path_to_sudo_binary
  | -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]] [-
  cleanpids]
  [-useractiondata user_action_data] [-schedule timer_value] [-eval]
```

Parameters

Table F-22 rhpcctl move gihome Command Parameters

Parameter	Description
-destwc <i>destination_workingcopy_name</i>	Specify the name of the destination working copy to which you want to move Oracle Grid Infrastructure.
-desthome <i>destination_oracle_home_path</i>	Alternative to specifying the name of the destination working copy, you can specify the path to the destination for the move of an Oracle home when you are moving the Oracle Grid Infrastructure home.
-sourcewc <i>working_copy_name</i>	If you want to move Oracle Grid Infrastructure from a working copy, then specify the name of the source working copy from which you want to move the Grid home.
-sourcehome <i>oracle_home_path</i>	If you are moving Oracle Grid Infrastructure from an unmanaged (not provisioned by Fleet Patching and Provisioning) Oracle home, then specify the path to the Oracle home from which you want to move Oracle Grid Infrastructure.
-targetnode <i>target_node_name</i>	Name of a node in a remote cluster with no Fleet Patching and Provisioning Client.
-ignorewcpatches	Use this parameter to ignore if the patched working copy is missing some patches which are present in the source path or working copy.
-nonrolling	Use this parameter to move the Oracle home in a non-rolling fashion.

Table F-22 (Cont.) rhpctl move gihome Command Parameters

Parameter	Description
-keepplacement	Specify this parameter to ensure that services of administrator-managed Oracle RAC or Oracle RAC One Node databases are running on the same instances before and after the move operation.
-auto -dbhomes <i>mapping_of_Oracle_homes</i>	Specify this parameter to automatically patch databases when you patch Oracle Grid Infrastructure.
-dblist <i>db_name_list</i>	Optionally, you can specify a list of databases on which you want to perform the patching operation.
-excludedblist <i>db_name_list</i>	Optionally, you can exclude specific databases from the patching operation.
-nodatapatch	Optionally, you can use this parameter to indicate not to run datapatch for databases being moved.
-disconnect	Optionally, you can use this parameter to disconnect all sessions before stopping or relocating services.
-stopoption <i>stop_option</i>	Optionally, you can choose one of the following stop options for the database: ABORT, IMMEDIATE, NORMAL, TRANSACTIONAL, or TRANSACTIONAL_LOCAL.
-drain_timeout <i>session_drain_time</i>	Optionally, you can use this parameter to specify a service drain timeout, in seconds.
-batches <i>list_of_batches</i>	Optionally, you can specify a comma-delimited list of batches of nodes (where each batch is a comma-delimited list of node names enclosed in parentheses) enclosed in double quotation marks (" ") in the format: "(nA,nB,...),(...nY,nZ)".
-smartmove [-saf <i>availability</i>	Alternatively, you can use the -smartmove parameter to auto-generate a list of batches of nodes and move databases by restarting instances after each batch. Optionally, you can use the -saf parameter to specify the service availability factor, which is the minimum percentage of instances on which a service must remain running during the move.
-schedule <i>timer_value</i>	Optionally, you can schedule a time to run this command in ISO-8601 format. For example: 2018-01-21T19:13:17+05.
-continue	Use this parameter to continue restarting the Oracle Clusterware stack on the next batch of nodes.
-revert	Use this parameter to revert back to before the move operation.
-abort	Use this parameter to abort an ongoing move operation.
-root -cred <i>cred_name</i> - sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_location</i> -auth <i>plugin_name</i> <i>plugin_args</i>	If you choose to use the -targetnode parameter, then you must choose either root, a credential name, sudo, or an authentication plugin to access the remote node. Choose -root to perform super user operations as root. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a sudo user by specifying a sudo user name and the path to the sudo binary, or to use an authentication plugin to access the remote node.

Table F-22 (Cont.) rhpctl move gihome Command Parameters

Parameter	Description
-cleanpids	When using a persistent home path for both the source and destination working copies, specify -cleanpids to ensure processes are stopped completely on the source home.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the useractiondata parameter of the user action script.
-schedule <i>timer_value</i>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05
-eval	Use this parameter to evaluate the rhpctl move gihome command and print automatically generated batches of nodes and the sequence of moves without actually running the command.

Usage Notes

If you choose to use the -schedule parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Example

Assume there is a target cluster running Oracle Grid Infrastructure 12c release 1 (12.1.0.2) from a working copy named grid12102wcpy, and one of the nodes in the cluster is named bposvr141. After provisioning the patched working copy, called grid12102PSU (using the -softwareonly parameter with the rhpctl add workingcopy command), move the Grid home to the patched working copy, as follows:

```
$ rhpctl move gihome -sourcewc grid12102wcpy -destwc grid12102PSU -root -
targetnode bposvr141
```

rhpctl upgrade gihome

Upgrades the Oracle Grid Infrastructure from a source working copy or source home path to a destination working copy.

Syntax

```
rhpctl upgrade gihome {-sourcewc source_workingcopy_name | -sourcehome
oracle_home_path
-targetnode target_node_name} -destwc destination_workingcopy_name
{-root | -sudouser sudo_user_name -sudopath sudo_binary_location]
-cred cred_name | -auth plugin_name [-arg1 name1:value1...]
[-ignoreprereq]
```


Parameters

Table F-23 rhpctl upgrade gihome Command Parameters

Parameter	Description
<code>-sourcewc</code> <code>source_workingcopy_name</code>	Specify the name of the source working copy from which the Oracle Grid Infrastructure home needs to be upgraded.
<code>-sourcehome</code> <code>oracle_home_path</code>	Alternative to specifying the name of the source working copy, you can specify the path to the unmanaged Oracle Grid Infrastructure home.
<code>-targetnode</code> <code>target_node_name</code>	In addition to specifying the source Oracle Grid Infrastructure home, you must also specify a node that is in a remote cluster that has no Fleet Patching and Provisioning Client.
<code>-destwc</code> <code>destination_workingcopy_name</code>	Specify the name of the destination working copy to which the Oracle Grid Infrastructure home is to be upgraded.
<code>-root</code> <code>-sudouser</code> <code>sudo_username</code> - <code>sudo_path</code> <code>sudo_binary_path</code> - <code>cred cred_name</code> - <code>auth plugin_name</code> [- <code>arg1</code> <code>name1:value1...</code>]	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>sudo</code> or <code>root</code> to access the remote node. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
<code>-ignoreprereq</code>	Use this parameter to ignore the CVU prerequisite checks.

rhpctl add image

Use the `rhpctl add image` command to create an image from an existing working copy and add it to the list of existing images on the Fleet Patching and Provisioning Server configuration.

Syntax

```
rhpctl add image -image image_name -workingcopy working_copy_name
  [-imagetype image_type] [-series series_name] [-state {TESTABLE |
  RESTRICTED | PUBLISHED}]
```

Parameters

Table F-24 rhpctl add image Command Parameters

Command Option	Description
<code>-image <i>image_name</i></code>	Specify the name of the image that you want to add.

Table F-24 (Cont.) rhpctl add image Command Parameters

Command Option	Description
<code>-workingcopy</code> <code>working_copy_name</code>	Specify the name of the working copy from which to create the image.
<code>-imagetype</code> <code>image_type</code>	Specify the software type. <code>ORACLEDBSOFTWARE</code> (default) for Oracle Database software, <code>ORACLEGISoftware</code> for Oracle Grid Infrastructure software, <code>ORACLEGGSoftware</code> for Oracle GoldenGate software, <code>LINUXOS</code> for Linux operating system ISO, or <code>SOFTWARE</code> for all other software. If you use custom image types, then specify the name of your image type.
<code>-series series_name</code>	If you want to add an image to an image series, then specify the name of an image series.
<code>-state {TESTABLE RESTRICTED PUBLISHED}</code>	Specify the state of the image.

Usage Notes



See Also:

[Patching Oracle Database Software](#) for details about how to use this command in the workflow for creating patched Oracle Database software homes

Example

An example of this command is:

```
$ rhpctl add image -image DB12201_PATCH -workingcopy
temp_wcpy_db12201_patch
```

rhpctl allow image

Allows access to an image by a user or a role.

Syntax

```
rhpctl allow image -image image_name {-user user_name [-client
cluster_name]
| -role role_name}
```

Parameters

Table F-25 rhpctl allow image Command Parameters

Parameter	Description
<code>-image <i>image_name</i></code>	Specify the name of the image to which you want to allow access.
<code>-user <i>user_name</i> [-client <i>cluster_name</i> -role <i>role_name</i>]</code>	Specify the either of the following: <ul style="list-style-type: none"> A user for which you want to allow access to the image and, optionally, the cluster name of the client cluster with the user. The role for which you want to allow access to the image.

Examples

To allow access to an image named PRODIMAGE:

```
$ rhpctl allow image -image PRODIMAGE -user mjk -client GHC1
```

rhpctl delete image

Deletes a specific image.

Syntax

```
rhpctl delete image -image image_name [-schedule timer_value]
```

Usage Notes

- Specify the name of the image you want to delete
- Optionally, you can use the `-schedule` parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:

```
2018-07-25T19:13:17+05
```

If you choose to use this parameter, then you must run this command on the Fleet Patching and Provisioning Server.

- This command will fail if the image belongs to one or more series
- This command will fail if there are any provisioned working copies based on this image

Example

The following example deletes an image named PRODIMAGEV0:

```
$ rhpctl delete image -image PRODIMAGEV0
```

rhpctl deploy image

Deploys an image to a specific node in a client cluster.

Syntax

```
rhpctl deploy image -image image_name [-targetnode node_name
  {-sudouser sudo_user_name -sudopath sudo_binary_path | -root}]
```

Parameters

Table F-26 rhpctl deploy image Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of the image you want to deploy.
-targetnode <i>node_name</i>	Optionally, you can specify the name of a node to which you want to deploy the image. This parameter is required if the node hosting the home is not a Fleet Patching and Provisioning Client.
-sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_path</i> - root]	If you use the -targetnode parameter, then you must specify either sudo or root to perform super user operations.

Usage Notes

You can only run this command from a Fleet Patching and Provisioning Server.

Example

The following example deploys an Oracle Database Appliance image to a node:

```
$ rhpctl deploy image -image ODA1 -targetnode racgbox1 -root
```

rhpctl disallow image

Disallows access to an image by a user or a role.

Syntax

```
rhpctl disallow image -image image_name {-user user_name [-client  
client_name]  
| -role role_name}
```

Parameters

Table F-27 rhpctl disallow image Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of the image to which you want to disallow access.

Table F-27 (Cont.) rhpctl disallow image Command Parameters

Parameter	Description
<code>-user <i>user_name</i> [-client <i>client_name</i> -role <i>role_name</i>]</code>	Specify either of the following: <ul style="list-style-type: none"> A user for which you want to disallow access to the image and, optionally, the cluster name of the client cluster with the user. The role for which you want to disallow access to the image.

Examples

To disallow access to an image:

```
$ rhpctl disallow image -image PRODIMAGE -user mjk -client GHCL
```

rhpctl import image

Creates an image on the Fleet Patching and Provisioning Server.

Use the `rhpctl import image` command to create an image by copying the entire software contents from the specified path to the Fleet Patching and Provisioning Server.

Syntax

```
rhpctl import image -image image_name -path path
  [-imagetype image_type] [-version software_version] [-pathowner
user_name]
  [-state {TESTABLE | RESTRICTED | PUBLISHED}] [-client cluster_name]
  [-targetnode node_name [-sudouser sudo_user_name -sudopath
sudo_binary_path | -root]]
  [-useractiondata user_action_data] [-schedule timer_value]
```

Parameters**Table F-28 rhpctl import image Command Parameters**

Parameter	Description
<code>-image <i>image_name</i></code>	Specify the name of the image that you want to add.
<code>-path <i>path</i></code>	Specify the absolute path location of the software home that you want to import (for Oracle Database images, this is the <code>ORACLE_HOME</code>).
<code>-imagetype <i>image_type</i></code>	Specify the software type. Use <code>ORACLEDBSOFTWARE</code> (default) for Oracle database software, <code>ORACLELEGISoftware</code> for Oracle Grid Infrastructure software, <code>ODAPATCHSOFTWARE</code> , for engineered systems (Oracle Data Appliance), or <code>SOFTWARE</code> for all other software. For a custom image type, use the image type name.
<code>-version <i>software_version</i></code>	Optionally, you can specify the version of the software you are importing.

Table F-28 (Cont.) rhpctl import image Command Parameters

Parameter	Description
<code>-pathowner <i>user_name</i></code>	Specify the user with read access to the files and directories under the specified path. Note: This parameter is applicable only for non-Oracle database software homes.
<code>-state {TESTABLE RESTRICTED PUBLISHED}</code>	Specify whether the state of the image is testable, restricted, or published.
<code>-client <i>cluster_name</i></code>	Specify the name of the client cluster.
<code>-targetnode <i>node_name</i></code>	Specify the name of the node from which you want to import the image. This parameter is required if the node hosting the home is not an Fleet Patching and Provisioning Client.
<code>-sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_path</i> - root]</code>	If you use the <code>-targetnode</code> parameter, then you must specify either <code>sudo</code> or <code>root</code> to perform super user operations.
<code>-useractiondata <i>user_action_data</i></code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-schedule <i>timer_value</i></code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05

Usage Notes

- You can only run this command on a Fleet Patching and Provisioning Server.
- When you import an Oracle Database or Oracle Grid Infrastructure software home, the version of the home must be one of the versions that Fleet Patching and Provisioning supports for provisioning and patching.

Examples

The following example imports an image:

```
$ rhpctl import image -image PRODIMAGEV1 -path /u01/app/product/12.1.0/dbhome -pathowner orcl
```

The following example imports an engineered system image:

```
$ rhpctl import image -image ODA1 -imagetype ODAPATCHSOFTWARE -path /tmp/ODAPatchBundle -version 12.1.2.8.0
```

rhpctl instantiate image

Requests copies of gold images from a peer Fleet Patching and Provisioning Server.

Syntax

```
rhpcctl instantiate image -server server_cluster_name {-image image_name
| -series series_name | -imagetype image_type | -all}
```

Parameters

Table F-29 rhpcctl instantiate image Command Parameters

Parameter	Description
-server <i>server_cluster_name</i>	Specify a Fleet Patching and Provisioning Server cluster from which you want to request images.
-image <i>image_name</i> -series <i>series_name</i> -imagetype <i>image_type</i> -all	<p>You can request copies of gold images from a peer Fleet Patching and Provisioning Server, specifically, by image name, series name, or image type. Alternatively, you can use the -all parameter to request copies of all gold images from the peer Fleet Patching and Provisioning server.</p> <p>If you choose to request images by image type, then specify ORACLEDBSOFTWARE (default) for Oracle Database software, ORACLEGISoftware for Oracle Grid Infrastructure software, ORACLEGGSoftware for Oracle GoldenGate software, and SOFTWARE for all other software. For a custom image type, use the image type name.</p>

Usage Notes

- User actions associated with an image being copied are not themselves copied.
- Groups configuration of a gold image is replicated in copies sent to peers.
- Copies of gold images are in the PUBLISHED state.

rhpcctl modify image

Modifies the configuration details of an image.

Syntax

```
rhpcctl modify image -image image_name -imagetype image_type
```

Parameters

Table F-30 rhpcctl modify image Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of the image that you want to modify.
-imagetype <i>image_type</i>	You can modify the software type. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGISoftware for Oracle Grid Infrastructure software, or SOFTWARE for all other software. For a custom image type, use the image type name.

rhpcctl query image

Displays the configuration of an existing image.

Syntax

```
rhpcctl query image {[-image image_name [-dbtemplate]] | [[-imagetype
image_type]
  [-version version] [-platform platform]]]
  [-server server_cluster_name | -client client_name | -local] | -drift}
  [-rhpserver rhps_regex]
```

Parameters

Table F-31 rhpcctl query image Command Parameters

Parameter	Description
-image <i>image_name</i> [-dbtemplate]	Specify the name of the image you want to query. Optionally, you can use the -dbtemplate parameter to display template file names in the default template directory.
-imagetype <i>image_type</i>	Specify the software type. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGRIDSOFTWARE for Oracle Grid Infrastructure software, or SOFTWARE for all other software. For a custom image type, use the image type name.
-version <i>version</i>	Specify the version of the image software you are querying.
-platform <i>platform</i>	Specify the operating system platform to which the image corresponds.
-server <i>server_cluster_name</i>	Specify the name of the server cluster to which the image corresponds.
-client <i>client_name</i>	If this parameter is used with the -image option, the command checks if the specified client has the given image instantiated on it. Otherwise, the use of the parameter lists the images instantiated on the given client.
-local	List the images stored on the current client.
-drift	List the the bug fixes not included in the golden image.
-rhpserver <i>rhps_regex</i>	Specify a regular expression to match the cluster name of the servers where the operation must be executed.

Usage Notes

If you use the -version parameter, then the version must have five fields, such as 12.1.0.2.4.

If you use the -platform parameter, then you can use Linux_AMD64, Linux_S390, Linux_PPC, IBM_AIX_PPC64, HP_IA64, Linux_Itanium, Solaris_SPARC64, Linux_LOP, and Intel_Solaris_AMD64

When issuing a command for a peer server using the -rhpserver option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the rhpcctl grant role command to grant a required role to the peer

server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles  
role=user_name[+user_name...][,role=user_name[+user_name...]]...
```

For information about granting roles with RHPCTL, refer to [rhpctl grant role](#)

rhpctl promote image

Promotes an image.

Syntax

```
rhpctl promote image -image image_name -state {TESTABLE | RESTRICTED |  
PUBLISHED}
```

Parameters

Table F-32 rhpctl promote image Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of the image that you want to promote.
-state {TESTABLE RESTRICTED PUBLISHED}	Specify one of the following as the name of the state of the image: TESTABLE: RESTRICTED: PUBLISHED:

Example

To promote an image named PRODIMAGE:

```
$ rhpctl promote image -image PRODIMAGE -state RESTRICTED
```

rhpctl unstantiate image

Stops updates for previously requested images from a peer Fleet Patching and Provisioning Server.

Syntax

```
rhpctl unstantiate image -server server_cluster_name {-image image_name  
| -series series_name | -imagetype image_type | -all}
```

Parameters

Table F-33 rhpctl uninstantiate image Command Parameters

Parameter	Description
<code>-server</code> <code>server_cluster_name</code>	Specify a Fleet Patching and Provisioning Server cluster from which you want to stop updates.
<code>-image image_name</code> <code>-series series_name</code> <code>-imagetype</code> <code>image_type</code> <code>-all</code>	You can updates from a peer Fleet Patching and Provisioning Server, specifically, by image name, series name, or image type. Alternatively, you can use the <code>-all</code> parameter to stop updates from the peer Fleet Patching and Provisioning server. If you choose to stop updates by image type, then specify <code>ORACLEDBSOFTWARE</code> (default) for Oracle Database software, <code>ORACLEGISoftware</code> for Oracle Grid Infrastructure software, <code>ORACLEGGSoftware</code> for Oracle GoldenGate software, and <code>SOFTWARE</code> for all other software. For a custom image type, use the image type name.

rhpctl add imagetype

Configures a new image type and its associated user actions.

Syntax

```
rhpctl add imagetype -imagetype image_type -basetype {SOFTWARE |
ORACLEGISoftware | ORACLEDBSOFTWARE | ORACLEGGSoftware}
[-useractions user_action_list]
```

Parameters

Table F-34 rhpctl add imagetype Command Parameters

Parameter	Description
<code>-imagetype</code> <code>image_type</code>	Specify the name of the image type you are creating.
<code>-basetype {SOFTWARE</code> <code>ORACLEGISoftware</code> <code>ORACLEDBSOFTWARE</code> <code>ORACLEGGSoftware}</code>	Specify a base image type on which the image type you are creating is based. Use <code>ORACLEDBSOFTWARE</code> (default) for Oracle Database software, <code>ORACLEGISoftware</code> for Oracle Grid Infrastructure software, <code>ORACLEGGSoftware</code> for Oracle GoldenGate software, and <code>SOFTWARE</code> for all other software.
<code>-useractions</code> <code>user_action_list</code>	Specify a comma-delimited list of names of user actions

Example

To add a new image type:

```
rhpctl add imagetype -imagetype DB122_PATCH_TYPE -basetype ORACLEDBSOFTWARE
```

rhpcctl allow imagetype

Grants access to an image type to a user or a role.

Syntax

```
rhpcctl allow imagetype -imagetype image_type {-user user_name [-client  
cluster_name] | -role role_name}
```

Parameters

Table F-35 rhpcctl allow imagetype Command Parameters

Parameter	Description
<code>-imagetype <i>image_type</i></code>	Specify the name of the image type to which you are granting access. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGISoftware for Oracle Grid Infrastructure software, or SOFTWARE for all other software. For a custom image type, use the image type name.
<code>-user <i>user_name</i></code>	Specify an operating system user to whom you are granting access to the image type. Either this parameter or the <code>-role</code> parameter is required.
<code>-client <i>cluster_name</i></code>	Optionally, you can specify the name of the client cluster to which the operating system user belongs, if you choose to use the <code>-user</code> parameter.
<code>-role <i>role_name</i></code>	Alternative to the <code>-user</code> parameter, you can specify a particular role to which to grant access to the image.

rhpcctl delete imagetype

Deletes an existing image type.

Syntax

```
rhpcctl delete imagetype -imagetype image_type
```

Usage Notes

Specify an image type to delete. You cannot delete any of the built-in image types.

rhpcctl disallow imagetype

Revokes access to an image type from a user or a role.

Syntax

```
rhpcctl disallow imagetype -imagetype image_type {-user user_name [-client  
cluster_name] | -role role_name}
```

Parameters

Table F-36 rhpctl disallow imagetype Command Parameters

Parameter	Description
<code>-imagetype</code> <i>image_type</i>	Specify the name of the image type from which you are revoking access. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGISoftware for Oracle Grid Infrastructure software, or SOFTWARE for all other software. For a custom image type, use the image type name.
<code>-user</code> <i>user_name</i>	Specify an operating system user from whom you are revoking access to the image type. Either this parameter or the <code>-role</code> parameter is required.
<code>-client</code> <i>cluster_name</i>	Optionally, you can specify the name of the client cluster to which the operating system user belongs, if you choose to use the <code>-user</code> parameter.
<code>-role</code> <i>role_name</i>	Alternative to the <code>-user</code> parameter, you can specify a particular role from which to revoke access to the image.

rhpctl modify imagetype

Modifies the configuration of an image type.

Syntax

```
rhpctl modify imagetype -imagetype image_type -useractions user_action_list
```

Parameters

Table F-37 rhpctl modify imagetype Command Parameters

Parameter	Description
<code>-imagetype</code> <i>image_type</i>	Specify the name of the image type you want to modify. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGISoftware for Oracle Grid Infrastructure software, ORACLEGGSoftware for Oracle GoldenGate software, or SOFTWARE for all other software. For a custom image type, use the image type name.
<code>-useractions</code> <i>user_action_list</i>	Specify a comma-delimited list of names of user actions

rhpctl query imagetype

Displays the configuration of an image type.

Syntax

```
rhpctl query imagetype -imagetype image_type [-rhpserver rhps_regex]
```

Parameters

Table F-38 rhpctl query imagetype Command Parameters

Parameter	Description
<code>-imagetype <i>image_type</i></code>	Specify the name of the image type you want to query. Use <code>ORACLEDBSOFTWARE</code> (default) for Oracle database software, <code>ORACLEGISoftware</code> for Oracle Grid Infrastructure software, <code>ORACLEGGSoftware</code> for Oracle GoldenGate software, or <code>SOFTWARE</code> for all other software. For a custom image type, use the image type name.
<code>-rhpserver <i>rhps_regex</i></code>	Specify a regular expression to match the cluster name of the servers where the operation must be executed.

Usage Notes

Specify the name of the image type you want to query. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISoftware` for Oracle Grid Infrastructure software, or `SOFTWARE` for all other software. For a custom image type, use the image type name.

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles  
role=user_name[+user_name...][,role=user_name[+user_name...]]...
```

For information about granting roles with RHPCTL, refer to [rhpctl grant role](#)

rhpctl delete job

Deletes a specific scheduled job from the repository.

Syntax

```
rhpctl delete job [-jobid job_id] [-force]
```

Parameters

Table F-39 rhpctl delete job Command Parameters

Parameter	Description
<code>-jobid <i>job_id</i></code>	Optionally, you can specify the job ID value for the job you want to delete that you obtained while scheduling the job. If you choose not to use this parameter, then RHPCTL deletes all jobs.
<code>-force</code>	Use this parameter to forcibly delete a job.

Usage Notes

You must run this command on the Fleet Patching and Provisioning Server.

Example

To delete a job with a job ID of 1:

```
$ rhpctl delete job -jobid 1
```

rhpctl query job

Queries the current status of a scheduled job with a specific job ID.

Syntax

```
rhpctl query job [-jobid job_id] [-status {EXECUTED | TIMER_RUNNING  
| EXECUTING | UNKNOWN | TERMINATED }] [-client client_name] [-user  
user_name]  
[-since timer_value] [-summary]
```

Parameters

Table F-40 rhpctl query job Command Parameters

Parameter	Description
<code>-jobid <i>job_id</i></code>	Optionally, you can specify the job ID value for the job you want to query that you obtained while scheduling the job. If you choose not to use this parameter, then RHPCTL queries all jobs.
<code>-status {EXECUTED TIMER_RUNNING EXECUTING UNKNOWN TERMINATED }</code>	Optionally, you can specify any of the following states of a job that you want to query: <ul style="list-style-type: none"> EXECUTED: The job is complete. TIMER_RUNNING: The timer for the job is still running. EXECUTING: The timer for the job has expired and is running. UNKNOWN: There is an unexpected failure due to issues such as a target going down, nodes going down, or any resource failures. TERMINATED: There is an abrupt failure or the operation has stopped.

Table F-40 (Cont.) rhpctl query job Command Parameters

Parameter	Description
-client <i>client_name</i> k	Optionally, you can specify the name of a client cluster for which you want to query jobs.
-user <i>user_name</i>	Optionally, you can specify the user name of the user for whom a software home is being provisioned.
-since <i>timer_value</i>	Optionally, you can specify a date from which to query the jobs, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05
-summary	Optionally, you can use this parameter to return only job details.

Usage Notes

You must run this command on the Fleet Patching and Provisioning Server.

Example

To query a specified scheduled job:

```
$ rhpctl query job -jobid 1
```

This command returns output similar to the following:

```
Job ID: 1
User: fred
Client: fredlinux4
Scheduled job command: "rhpctl import image -image DB-Imagel -imagetype
ORACLEDBSOFTWARE -path /ade/fred_linux4/esw1 -schedule
2018-07-27T13:38:57Z"
Scheduled job execution start time: 2018-07-27T05:38:57-08. Equivalent
local time: 2018-07-27 05:38:57
Current status: EXECUTED
Result file path: "/scratch/rhp_storage/chkbase/scheduled/
job-1-2017-11-27-05:39:14.log"
Job execution start time: 2018-07-27 05:39:14
Job execution end time: 2018-07-27 05:43:09
Job execution elapsed time: 3 minutes 55 seconds

Result file "/scratch/rhp_storage/chkbase/scheduled/
job-1-2018-07-27-05:39:14.log" contents:
slc05amw.foo.com: Audit ID: 4
slc05amw.foo.com: Creating a new ACFS file system for image "DB-Imagel" ...
slc05amw.foo.com: Copying files...
slc05amw.foo.com: Copying home contents...
slc05amw.foo.com: Changing the home ownership to user fred...
slc05amw.foo.com: Changing the home ownership to user fred...
```

rhpcctl collect osconfig

Collects a backup of the operating system configuration for a cluster.

Syntax

```
rhpcctl collect osconfig -client cluster_name [-targetnode node_name
  {-sudouser sudo_user_name -sudopath sudo_binary_location | -root}]
```

Parameters

Table F-41 rhpcctl collect osconfig Command Parameters

Parameter	Description
-client <i>cluster_name</i>	Specify the name of the client cluster.
-targetnode <i>node_name</i>	Optionally, you can specify the name of a particular node in a cluster from which to collect configuration information.
-sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_path</i> - root]	If you use the -targetnode parameter, then you must specify either sudo or root to perform super user operations.

rhpcctl compare osconfig

Compares operating system configurations for a specific cluster.

Syntax

```
rhpcctl compare osconfig -client cluster_name -node node_name -id1  
identifier -id2 identifier
```

Parameters

Table F-42 rhpcctl compare osconfig Command Parameters

Parameter	Description
-client <i>cluster_name</i>	Specify the name of the client cluster in which you want to compare operating system configurations.
-node <i>node_name</i>	Specify the name of a node in a remote cluster.
-id1 <i>identifier</i>	Specify an identifier of an operating system configuration to be considered as a reference.
-id2 <i>identifier</i>	Specify an identifier of an operating system configuration to be compared.

rhpcctl disable osconfig

Disables a scheduled backup of the operating system configuration and gives the option to delete all collected configuration backups.

Syntax

```
rhpcctl disable osconfig [-client cluster_name] [-clean]
```

Usage Notes

- Optionally, you can specify a client cluster name on which you want to disable collection of operating system configuration information.
- Optionally, you can use the `-clean` parameter to delete all operating system configuration backups.

rhpcctl enable osconfig

Enable operating system configuration information collection for the client cluster.

Syntax

```
rhpcctl enable osconfig -client cluster_name [-retaincopies count]
  [-start timer_value] [-frequency collect_frequency] [-collectnow
  [-targetnode node_name {-sudouser sudo_user_name -sudopath
  sudo_binary_location
  | -root}]] [-force]
```

Parameters

Table F-43 rhpcctl enable osconfig Command Parameters

Parameter	Description
<code>-client <i>cluster_name</i></code>	Specify the name of the client cluster.
<code>-retaincopies <i>count</i></code>	Optionally, you can specify the number of scheduled backups you want to be maintained. The default value is 37.
<code>-start <i>timer_value</i></code>	Optionally, you can specify a start date and time to run configuration collection according to the following example: 2018-07-23T00:00:00-07
<code>-frequency <i>collect_frequency</i></code>	Optionally, you can specify the configuration collection interval in number of days.
<code>-collectnow</code>	Optionally, you can use this parameter to collect configuration information, immediately.
<code>-targetnode <i>node_name</i></code>	Optionally, you can specify the name of a particular node in a cluster from which to collect configuration information.

Table F-43 (Cont.) rhpctl enable osconfig Command Parameters

Parameter	Description
-sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_path</i> - root]	If you use the <code>-targetnode</code> parameter, then you must specify either <code>sudo</code> or <code>root</code> to perform super user operations.
-force	Optionally, you can use this parameter to forcibly modify the count for the <code>-retaincopies</code> parameter previously set.

rhpctl query osconfig

Provides historic operating system configuration collection information, such as the collection schedule, retention count, scheduled job for periodic collection, and collection data.

Syntax

```
rhpctl query osconfig -client client_name
```

Usage Notes

Provide the name of the client cluster that you want to query operating system configuration collection information.

Example

This command returns output similar to the following:

```
$ rhpctl query osconfig -client rhpdemocluster

OSConfig Enabled: true
Collection start time: "00:00:00"
Collection frequency: "1"
retaincopies count: "35"
OSConfig periodic Job ID: "38"
Collection storage path: "/scratch/rhp_storage/chkbase/osconfig/
rhpdemocluster"
Latest list of nodes for collections : "mjk00fwc"
OSConfig ID: "22" Collected on: "Jul 27, 2018 22:00:58 PM"
OSConfig ID: "21" Collected on: "Jul 26, 2018 22:00:47 PM"
OSConfig ID: "20" Collected on: "Jul 25, 2018 22:00:29 PM"
```

rhpctl query peerserver

Displays information for a registered peer Fleet Patching and Provisioning Server.

Syntax

```
rhpcctl query peerserver [-server server_cluster_name [-serverPolicy]]
```

Parameters

Table F-44 rhpcctl query peerserver Command Parameters

Parameter	Description
-server <i>server_cluster_name</i>	Optionally, you can specify the name of the Fleet Patching and Provisioning Server cluster for which you want to view the information.
-serverPolicy	Optionally, you can specify the image policy for the peer Fleet Patching and Provisioning Server for which you want to view the information.

rhpcctl add role

Creates roles and adds them to the list of existing roles on the Fleet Patching and Provisioning Server configuration.



See Also:

["Fleet Patching and Provisioning Roles"](#)

Syntax

```
rhpcctl add role -role role_name -hasRoles roles
```

Parameters

Table F-45 rhpcctl add role Command Parameters

Parameter	Description
-role <i>role_name</i>	Specify a name for the role that you want to create.

Table F-45 (Cont.) rhpctl add role Command Parameters

Parameter	Description
<code>-hasRoles roles</code>	Specify a comma-delimited list of roles to include with the new role. GH_AUDIT_ADMIN GH_USER_ADMIN GH_ROLE_ADMIN GH_SITE_ADMIN GH_SERIES_ADMIN GH_SERIES_CONTRIB GH_WC_ADMIN GH_WC_OPER GH_WC_USER GH_SUBSCRIBE_USER GH_SUBSCRIBE_ADMIN GH_IMG_ADMIN GH_IMG_USER GH_IMG_TESTABLE GH_IMG_RESTRICT GH_IMG_PUBLISH GH_IMG_VISIBILITY GH_IMGTYPE_ADMIN GH_IMGTYPE_ALLOW GH_IMGTYPE_OPER GH_SA GH_CA GH_OPER

Usage Notes

- You can only run this command on the Fleet Patching and Provisioning Server.
- You must be assigned the GH_ROLE_ADMIN role to run this command.

Example

To add a role on the Fleet Patching and Provisioning Server:

```
$ rhpctl add role -role hr_admin -hasRoles GH_WC_USER,GH_IMG_USER
```

rhpctl delete role

Deletes a role from the list of existing roles on the Fleet Patching and Provisioning Server configuration.

Syntax

```
rhpctl delete role -role role_name
```

Usage Notes

- Specify the name of the role that you want to delete
- You *cannot* delete any built-in roles
- You can only run this command on the Fleet Patching and Provisioning Server

Example

To delete a role from the Fleet Patching and Provisioning Server:

```
$ rhpctl delete role -role hr_admin
```

rhpctl grant role

Grants a role to a client user or to another role.

Syntax

```
rhpctl grant role {-role role_name {-user user_name [-client cluster_name]
| -grantee role_name}} | {{[-client cluster_name]
[-maproles role=user_name[+user_name...][,role=user_name[+user_name...]]
[,...]
```

Parameters

Table F-46 rhpctl grant role Command Parameters

Parameter	Description
<code>-role <i>role_name</i></code>	Specify the name of the role that you want to grant clients or users.
<code>-user <i>user_name</i> [-client <i>cluster_name</i>]</code>	Specify the name of a user. The user name that you specify must be in the form of <i>user@rhpcclient</i> , where <i>rhpcclient</i> is the name of the Fleet Patching and Provisioning Client. Optionally, you can specify the name of the client cluster to which the user belongs.
<code>-grantee <i>role_name</i></code>	Use this parameter to specify a role to which you want to grant another role.
<code>[-client <i>cluster_name</i>] -maproles <i>role=user_name</i>[+<i>user_name</i>...][,<i>role=user_name</i>[+<i>user_name</i>...]][,...]</code>	You can map either built-in roles or roles that you have defined to either users on a specific client cluster or to specific users. When you use the <code>-maproles</code> parameter, use a plus sign (+) to map more than one user to a specific role. Separate additional role/user pairs with commas.

Example

The following example grants a role, ABC, to four specific users.

```
$ rhpctl grant role -role ABC -maproles  
ABC=mjk@rhpc1+dc@rhpc1+aj@rhpc1+jc@rhpc1
```

rhpctl query role

Displays the configuration information of a specific role.

Syntax

```
rhpctl query role [-role role_name]
```

Usage Notes

- Specify the name of the role for which you want to display the configuration information
- You can only run this command on the Fleet Patching and Provisioning Server

Example

This command returns output similar to the following:

```
$ rhpctl query role -role GH_CA  
  
Role name: GH_CA  
Associated roles: GH_IMGTYPE_ADMIN, GH_IMGTYPE_ALLOW, GH_IMGTYPE_OPER,  
GH_IMG_ADMIN, GH_IMG_PUBLISH, GH_IMG_RESTRICT, GH_IMG_TESTABLE,  
GH_IMG_VISIBILITY, GH_SERIES_ADMIN, GH_SERIES_CONTRIB, GH_SUBSCRIBE_ADMIN,  
GH_WC_ADMIN  
Users with this role: rhpusr@rwsdcVM13
```

rhpctl revoke role

Revokes a role from a client user.

Syntax

```
rhpctl revoke role {-role role_name {-user user_name  
[-client cluster_name] | -grantee role_name}}  
| {{[-client cluster_name] -maproles role=user_name[+user_name...]  
[,role=user_name[+user_name...]]...}}
```

Parameters

Table F-47 rhpctl revoke role Command Parameters

Parameter	Description
<code>-role <i>role_name</i></code>	Specify the name of the role from which you want to revoke clients or users.
<code>-user <i>user_name</i> [-client <i>cluster_name</i>]</code>	Specify the name of a user and, optionally, a client cluster from which you want to revoke a role. The user name that you specify must be in the form of <code>user@rhpclient</code> , where <code>rhpclient</code> is the name of the Fleet Patching and Provisioning Client.
<code>-grantee <i>role_name</i></code>	Specify the grantee role name.
<code>[-client <i>client_name</i>] -maproles <i>role=user_name[+user_name...]</i></code>	You can map either built-in roles or roles that you have defined to specific users. Use a plus sign (+) to map more than one user to a specific role. Separate additional role/user pairs with commas. Optionally, you can also specify a client cluster.

rhpctl add series

Adds a series to the Fleet Patching and Provisioning Server configuration.

Syntax

```
rhpctl add series -series series_name [-image image_name]
```

Parameters

Table F-48 rhpctl add series Command Parameters

Parameter	Description
<code>-series <i>series_name</i></code>	Specify a name for the series that you want to add.
<code>-image <i>image_name</i></code>	Optionally, you can specify the name of a configured image. This image becomes the first in the series.

Example

To add a series:

```
$ rhpctl add series -series DB12_series
```

rhpctl delete series

Deletes a series from the Fleet Patching and Provisioning Server configuration.

Syntax

```
rhpctl delete series -series series_name [-force]
```

Usage Notes

- Specify the name of the series that you want to delete.
- Use `-force` to delete an image series even if the series includes images.
- Before deleting an image series, you must first remove all images from the series by using the `rhctl deleteimage series` command.
- This command *does not* delete images, only series.

Example

The following example deletes a series called PRODDBSERIES:

```
$ rhctl delete series -series PRODDBSERIES
```

rhctl deleteimage series

Deletes an image from a series.

Syntax

```
rhctl deleteimage series -series series_name -image image_name
```

Parameters

Table F-49 rhctl deleteimage series Command Parameters

Parameter	Description
<code>-series <i>series_name</i></code>	Specify the name of the series from which you want to delete an image.
<code>-image <i>image_name</i></code>	Specify the name of the image that you want to delete from a series.

Example

The following command deletes an image called PRODIMAGEV0 from a series called PRODDBSERIES:

```
$ rhctl deleteimage series -series PRODDBSERIES -image PRODIMAGEV0
```

rhctl insertimage series

Inserts an existing image into a series.

**Note:**

A single image can belong to one or more series.

Syntax

```
rhpcctl insertimage series -series series_name -image image_name
[-before image_name]
```

Parameters

Table F-50 rhpcctl insertimage series Command Parameters

Parameter	Description
-series <i>series_name</i>	Specify the name of the series into which you want to insert an image.
-image <i>image_name</i>	Specify the name of the image that you want to insert into a series.
-before <i>image_name</i>	Optionally, you can specify the name of an image before which you want to insert the new image.

Example

To insert an image into a series:

```
rhpcctl insertimage series -series DB12_series -image DB12102_PSU
```

rhpcctl query series

Displays the configuration of a series.

Syntax

```
rhpcctl query series [-series series_name | -image image_name]
[-server server_cluster_name] [-rhpserver rhps_regex]
```

Parameters

Table F-51 rhpcctl query series Command Parameters

Parameter	Description
-series <i>series_name</i>	Specify the name of the series for which you want to display the configuration.
-image <i>image_name</i>	Alternatively, you can specify the name of a configured image.
-server <i>server_cluster_name</i>	Specify the name of the server cluster to which the image corresponds.
-rhpserver <i>rhps_regex</i>	Specify a regular expression to match the cluster name of the servers where the operation must be executed.

Usage Notes

If you do not specify a series or an image by name, then CRSCTL returns information for all series.

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpcctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpcctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpcctl grant role -client cluster_name -maproles  
role=user_name[+user_name...][,role=user_name[+user_name...]....] ]
```

For information about granting roles with RHPCTL, refer to [rhpcctl grant role](#)

Example

This command returns output similar to the following:

```
$ rhpcctl query series  
  
Image series: DB12_series  
Image series: GRID_series  
Image series: DB112_series
```

rhpcctl subscribe series

Subscribes a specific user to an image series.

Syntax

```
rhpcctl subscribe series -series series_name [-user user_name [-client  
cluster_name]]
```

Parameters

Table F-52 rhpcctl subscribe series Command Parameters

Parameter	Description
<code>-series series_name</code>	Specify the image series to which you want to subscribe a user.
<code>-user user_name</code>	Specify an operating system user to whom you are subscribing the image series.
<code>-client cluster_name</code>	Optionally, you can specify the name of the client cluster to which the operating system user belongs.

rhpcctl unsubscribe series

Unsubscribes a user from an image series.

Syntax

```
rhpcctl unsubscribe series -series series_name [-user user_name [-client  
cluster_name]]
```

Parameters

Table F-53 rhpcctl unsubscribe series Command Parameters

Parameter	Description
-series <i>series_name</i>	Specify the image series from which you want to unsubscribe a user.
-user <i>user_name</i>	Specify an operating system user from whom you are unsubscribing the image series.
-client <i>cluster_name</i>	Optionally, you can specify the name of the client cluster to which the operating system user belongs.

rhpcctl export server

Exports data from the repository to a Fleet Patching and Provisioning Server data file.

Syntax

```
rhpcctl export server -server peer_server_name -serverdata file_path
```

Usage Notes

- Specify the name of a peer server cluster.
- Specify the path to the file containing the Fleet Patching and Provisioning Server data.

rhpcctl query server

Displays the configuration of a server.

Syntax

```
rhpcctl query server
```

Usage Notes

This command has no parameters.

Example

This command displays output similar to the following:

```
$ ./rhpctl query server
```

```
Fleet Patching and Provisioning Server (RHPS): rhps-myserver
Storage base path: /u01/app/RHPImages
Disk Groups: RHPDATA
Port number: 8896
```

rhpctl register server

Registers the specific Fleet Patching and Provisioning Server as a peer server.

Syntax

```
rhpctl register server -server server_cluster_name -serverdata file
  {-root | -cred cred_name | -sudouser sudo_username -sudopath
  path_to_sudo_binary
  | -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]}
```

Parameters

Table F-54 rhpctl register server Command Parameters

Parameter	Description
-server <i>server_cluster_name</i>	Specify the name of the Fleet Patching and Provisioning Server cluster that you want to register.
-serverdata <i>file</i>	Specify the path to the file containing the Fleet Patching and Provisioning Server data.
-root -cred <i>cred_name</i> - sudouser <i>sudo_user_name</i> - sudopath <i>sudo_binary_location</i> -auth <i>plugin_name</i> <i>plugin_args</i>	Choose -root to perform super user operations as root. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a sudo user by specifying a sudo user name and the path to the sudo binary, or to use an authentication plugin to access the remote node.

rhpctl unregister server

Unregisters a specific Fleet Patching and Provisioning Server as a peer server.

Syntax

```
rhpctl unregister server -server server_cluster_name
```

Usage Notes

Specify the name of the Fleet Patching and Provisioning Server you want to unregister as a peer.

rhpcctl delete user

Deletes a user from the Fleet Patching and Provisioning repository.

Syntax

```
rhpcctl delete user -user user_name [-client cluster_name]
```

Parameters

Table F-55 rhpcctl delete user Command Parameters

Parameter	Description
-user <i>user_name</i>	Specify the name of the user you want to delete from a Fleet Patching and Provisioning Client.
-client <i>cluster_name</i>	Optionally, you can specify the name of the client cluster from which you want to delete from a specific user.

Usage Notes

- You can delete non built-in users only if that user does not own any working copies.
- If the user created an image or image series, then you can still delete the user, but the creator of the image or image series is changed to *internal-user@GHS*.
- If the user was the owner of an image series, then you can delete the user, but the owner of the image series will be changed to *internal-user@GHS*. You can still use the affected image series as normal, such that you can still provision a working copy from the affected image series, and you can still insert or delete images from the affected image series.

Example

The following example deletes the user named `scott` on the server cluster from the Fleet Patching and Provisioning repository:

```
$ rhpcctl delete user -user scott
```

rhpcctl modify user

Modifies the email address of a specific user.

Syntax

```
rhpcctl modify user -user user_name -email email_address [-client client_name]
```

Parameters

Table F-56 rhpctl modify user Command Parameters

Parameter	Description
<code>-user user_name</code>	Specify an operating system user whose email address you want to modify.
<code>-email email_address</code>	Specify the email address of the operating system user in the RFC 822 format.
<code>-client client_name</code>	Optionally, you can specify the name of the client cluster to which the operating system user belongs.

rhpctl register user

Registers an email address for a specific user.

Syntax

```
rhpctl register user -user user_name -email email_address [-client
client_name]
```

Parameters

Table F-57 rhpctl register user Command Parameters

Parameter	Description
<code>-user user_name</code>	Specify an operating system user whose email address you want to register.
<code>-email email_address</code>	Specify the email address of the operating system user in the RFC 822 format.
<code>-client client_name</code>	Optionally, if you run the command on the Fleet Patching and Provisioning Server, then you can specify the name of the client cluster to which the operating system user belongs. Otherwise, the command applies to a user on the cluster (either the Fleet Patching and Provisioning Server or Client) where the command is run.

Example

An example of this command is:

```
$ rhpctl register user -user scott -email scott@company.com
```

rhpctl unregister user

Unregisters an email address for a specific user.

Syntax

```
rhpcctl unregister user -user user_name [-client client_name]
```

Parameters

Table F-58 rhpcctl unregister user Command Parameters

Parameter	Description
-user <i>user_name</i>	Specify an operating system user whose email address you want to unregister.
-client <i>client_name</i>	Optionally, you can specify the name of the client cluster to which the operating system user belongs.

rhpcctl add useraction

Configures a user action and its associated script and action file.

Syntax

```
rhpcctl add useraction -useraction user_action_name -actionscript
script_name
  [-actionfile file_name] [-pre | -post] [-optype option]
  [-onerror {ABORT | CONTINUE}] [-runscope {ONENODE | ALLNODES | AUTO}]
```

Parameters

Table F-59 rhpcctl add useraction Command Parameters

Parameter	Description
-useraction <i>user_action_name</i>	Specify the name of the user action you want to add.
-actionscript <i>script_name</i>	Associate a specific action script to run with the user action.
-actionfile <i>file_name</i>	Optionally, you can specify an action file that is required by the user action.
-pre -post	Use the -pre parameter to run the user action before the add operation or the -post parameter to run the user action after.

Table F-59 (Cont.) rhpctl add useraction Command Parameters

Parameter	Description
<code>-optype <i>option</i></code>	Optionally, you can specify the operation for which the user action is configured. Options include: IMPORT_IMAGE ADD_WORKINGCOPY DELETE_WORKINGCOPY ADD_DATABASE DELETE_DATABASE MOVE_DATABASE MOVE_GIHOME UPGRADE_DATABASE UPGRADE_GIHOME ADDNODE_GIHOME DELETENODE_GIHOME ADDNODE_DATABASE DELETENODE_DATABASE ADDNODE_WORKINGCOPY ZDTUPGRADE_DATABASE ZDTUPGRADE_DATABASE_SNAPDB ZDTUPGRADE_DATABASE_DBUA ZDTUPGRADE_DATABASE_SWITCHBACK
<code>-onerror {ABORT CONTINUE}</code>	Optionally, you can choose whether to abort or continue the operation if the user action encounters an error while it is running.
<code>-runscope {ONENODE ALLNODES AUTO}</code>	Optionally, you can specify the nodes where the user action is run. Choose ONENODE to run the user action for each database on the node on which a patch was applied to the database. Choose ALLNODES to run the user action for each database on every cluster node. Choose AUTO for a run scope based on the other command options.

rhpctl delete useraction

Deletes an existing user action configuration.

Syntax

```
rhpctl delete useraction -useraction user_action_name
```

Usage Notes

Specify the name of a user action you want to delete.

rhpctl modify useraction

Modifies the configuration of the specified user action name.

Syntax

```

rhpctl modify useraction -useraction user_action_name [-actionscript
script_name]
  [-actionfile file_name] [-pre | -post] [-optype option]
  [-onerror {ABORT | CONTINUE}] [-runscope {ONENODE | ALLNODES | AUTO}]

```

Parameters

Table F-60 rhpctl modify useraction Command Parameters

Parameter	Description
-useraction <i>user_action_name</i>	Specify the name of the user action you want to modify.
-actionscript <i>script_name</i>	Optionally, you can specify an action script to run.
-pre -post	Use the -pre parameter to run the user action before the modify operation or the -post parameter to run the user action after.
-optype <i>option</i>	Optionally, you can specify the operation for which the user action is configured. Options include: <ul style="list-style-type: none"> IMPORT_IMAGE ADD_WORKINGCOPY DELETE_WORKINGCOPY ADD_DATABASE DELETE_DATABASE MOVE_DATABASE MOVE_GIHOME UPGRADE_DATABASE UPGRADE_GIHOME ADDNODE_GIHOME DELETENODE_GIHOME ADDNODE_DATABASE DELETENODE_DATABASE ADDNODE_WORKINGCOPY ZDTUPGRADE_DATABASE ZDTUPGRADE_DATABASE_SNAPDB ZDTUPGRADE_DATABASE_DBUA ZDTUPGRADE_DATABASE_SWITCHBACK
-onerror {ABORT CONTINUE}	Optionally, you can choose whether to abort or continue the operation if the user action encounters an error while it is running.
-runscope {ONENODE ALLNODES AUTO}	Optionally, you can specify the nodes where the user action is run. Choose ONENODE to run the user action for each database on the node on which a patch was applied to the database. Choose ALLNODES to run the user action for each database on every cluster node. Choose AUTO for a run scope based on the other command options.

rhpcctl query useraction

Displays the configuration of a user action.

Syntax

```
rhpcctl query useraction [-useraction user_action_name | -imagetype  
image_type]  
    [-optype option]
```

Parameters

Table F-61 rhpcctl query useraction Command Parameters

Parameter	Description
-useraction <i>user_action_name</i>	Specify the name of the user action you want to query.
-imagetype <i>image_type</i>	Alternatively, you can specify the software type. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGISoftware for Oracle Grid Infrastructure software, and SOFTWARE for all other software. For a custom image type, use the image type name.
-optype <i>option</i>	Optionally, you can specify the operation for which to run the query. Options include: IMPORT_IMAGE ADD_WORKINGCOPY DELETE_WORKINGCOPY ADD_DATABASE DELETE_DATABASE MOVE_DATABASE MOVE_GIHOME UPGRADE_DATABASE UPGRADE_GIHOME ADDNODE_GIHOME DELETENODE_GIHOME ADDNODE_DATABASE DELETENODE_DATABASE ADDNODE_WORKINGCOPY

rhpcctl add workingcopy

Creates a working copy on a client cluster.

Syntax

To add a working copy to a client cluster:

```
rhpcctl add workingcopy -workingcopy workingcopy_name  
    {-image image_name | -series series_name}  
    [-oraclebase oraclebase_path] [-path where_path]
```

```
[-storagetype {LOCAL | RHP_MANAGED}] [-user user_name]
[-client cluster_name] [-ignoreprereq | -fixup]
[-responsefile response_file_path] [-clusternodes node_list]
[-groups group_list] [-root | -cred cred_name | -sudouser sudo_username
  -sudopath path_to_sudo_binary | -auth plugin_name [-arg1 name1:value1
  [-arg2 name2:value2 ...]]]
[-notify [-cc users_list]] [-asmclientdata data_path]
[-gnsclientdata data_path] [-clustermanifest data_path] [-softwareonly]
[-local] [-inventory inventory_path] [-targetnode target_node_name]
[-agpath read_write_path -aupath gold_image_path] [-setupssh]
[-useractiondata user_action_data] [-eval] [-schedule timer_value]
[-checkwcpatches -sourcehome source_home_path] [-scan scan_name]
[-diskDiscoveryString disk_discovery_string] [-dbnodes db_node_list]
[-cells cell_list] [-ibswitches ibswitch_list] [-fromnode node_name]
[-unkey] [-smtpfrom "address"] [-smtpto "addresses"] [-precheckonly]
[-modifyatprereq] [-resetforce] [-force] [-help options]
```

Parameters

Table F-62 rhpctl add workingcopy Command Parameters

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify a name for the working copy that you want to create.
{-image <i>image_name</i> - series <i>series_name</i> }	Specify the name of a configured image from which to create a working copy or the name of an image series from which RHPCTL takes the latest image when adding a working copy.
-oraclebase <i>oracle_base_path</i>	Specify an ORACLE_BASE path for provisioning an Oracle Database or Oracle Grid Infrastructure home. You can specify either an existing directory or a new directory. Note: This parameter is required only for the ORACLEDBSOFTWARE and ORACLEGISoftware image types.
-inventory <i>inventory_path</i>	Specify the location of the Oracle Inventory directory.
-path <i>absolute_path</i>	Specify the absolute path for provisioning the software home on the client side (this location must be empty). For Oracle Database images, this becomes the ORACLE_HOME. Note: This parameter is required for LOCAL storage types, and is invalid for RHP_MANAGED.
-storagetype {LOCAL RHP_MANAGED}	Specify the type of storage for the software home.
-user <i>user_name</i>	Specify the name of the user that will own the working copy being provisioned. If you do not specify this parameter, then the working copy is owned by the user running the command. If you are provisioning to a remote cluster, then the user name must be a valid user on the remote cluster. The user ID need not be the same between the two clusters, but the user name must exist on both. Note: You cannot use -user simultaneously with the -softwareonly parameter.
-client <i>cluster_name</i>	Specify the name of the client cluster.

Table F-62 (Cont.) rhpctl add workingcopy Command Parameters

Parameter	Description
<code>-ignoreprereq</code> <code>-fixup</code>	You can choose to ignore the Clusterware Verification Utility (CVU) checks or you can choose to run the recommended fixup script. Note: These parameters are valid only when you are provisioning Oracle Grid Infrastructure.
<code>-responsefile</code> <code>response_file_path</code>	Specify a response file to use when you provision Oracle Grid Infrastructure.
<code>-clusternodes</code> <code>node_name:node_vip[:node_role]</code> <code>[,node_name:node_vip[:node_role]...]</code>	Specify a comma-delimited list of cluster node information on which to provision Oracle Clusterware.
<code>-groups "OSDBA OSOPER OSASM OSBACKUP OSDG OSKM OSRAC=group_name[,...]"</code>	Specify a comma-delimited list of Oracle groups, enclosed in double quotation marks (" "), that you want to configure in the working copy. For example: <code>-groups "OSDBA=dba,OSOPER=oper"</code>
	When you create a gold image from a source home or working copy, the gold image inherits the groups configured in the source. When you create a working copy from that gold image using <code>rhpctl add workingcopy</code> , by default, the new working copy inherits the same groups as the gold image. If you use the <code>-groups</code> parameter on the command line, then: <ul style="list-style-type: none"> Groups configured in the gold image that you do not specify on the command line are inherited by the working copy. Groups configured in the gold image that you also specify on the command line are set to the value that you specify on the command line (command line parameters override the gold image). Groups that you specify on the command line that are not in the gold image are added to the configured groups in the gold image (the command line adds new groups). Notes: <ul style="list-style-type: none"> When you move or upgrade a source home (unmanaged or working copy), the groups in the destination working copy must match those of the source home. You cannot use <code>-groups</code> simultaneously with the <code>-softwareonly</code> parameter.
<code>-root</code> <code>-cred cred_name</code> <code>-sudouser</code> <code>sudo_user_name</code> <code>-sudopath</code> <code>sudo_binary_location</code> <code>-auth plugin_name</code> <code>plugin_args</code>	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>root</code> , a credential name, <code>sudo</code> , or an authentication plugin to access the remote node. Choose <code>-root</code> to perform super user operations as <code>root</code> . Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a <code>sudo</code> user by specifying a <code>sudo</code> user name and the path to the <code>sudo</code> binary, or to use an authentication plugin to access the remote node.
<code>-notify [-cc user_list]</code>	Specify this parameter to have email notifications sent to the owner of the working copy. Optionally, you can include a list of additional users who will receive notifications.
<code>-asmclientdata data_path</code>	Specify the path to a file that contains Oracle ASM client data.

Table F-62 (Cont.) `rhpcctl add workingcopy` Command Parameters

Parameter	Description
<code>-gnsclientdata data_path</code>	Specify the path to a file that contains the Grid Naming Service (GNS) data.
<code>-clustermanifest data_path</code>	Optionally, you can specify the location of cluster manifest file. You can use this parameter when the Fleet Patching and Provisioning Server is on a domain services cluster and you are creating a member cluster.
<code>-local</code>	Use this parameter to provision only Oracle Grid Infrastructure software on the local node. Note: You can only use this parameter in conjunction with the <code>-softwareonly</code> parameter, and only when running the <code>rhpcctl add workingcopy</code> command on a Fleet Patching and Provisioning Server.
<code>-softwareonly</code>	Use this parameter to provision only Oracle Grid Infrastructure software.
<code>-targetnode target_node_name</code>	Specify the name of a node in a remote cluster with no Fleet Patching and Provisioning Client on which you want to provision a working copy.
<code>-agpath read_write_path</code> <code>-aupath gold_image_path</code>	Use <code>-agpath</code> to specify the path to the read-write, site-specific configuration changes to set the persistent home path, and use <code>-aupath</code> to specify the path for the read-only gold image to set the persistent home path.
<code>-setupssh</code>	Use this parameter to set up passwordless SSH user equivalence on the remote nodes for the provisioning user.
<code>-useractiondata user_action_data</code>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
<code>-eval</code>	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.
<code>-schedule timer_value</code>	Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example: 2018-07-25T19:13:17+05
<code>-checkwcpatches - sourcehome source_home_path</code>	Optionally, you can use the <code>-checkwcpatches</code> and <code>-sourcehome</code> parameters to compare patches in a specific source home path with the patches in the working copy you want to add.
<code>-scan scan_name</code>	Optionally, you can use this parameter to specify a SCAN name.
<code>-diskDiscoveryString disk_discovery_string</code>	Optionally, you can use this parameter to specify a disk discovery string.
<code>-dbnodes db_node_list</code>	Optionally, you can use this parameter to specify a file containing the list of database servers that you want to patch, one node name in each line.
<code>-cells cell_list</code>	Optionally, you can use this parameter to specify a file containing a list of storage servers that you want to patch, one node name in each line.
<code>-ibswitches ibswitch_list</code>	Optionally, you can use this parameter to specify a file containing a list of InfiniBand switches that you want to patch, one node name in each line.
<code>-fromnode node_name</code>	Optionally, you can specify a database server node name from where the patch manager is to be run.
<code>-unkey</code>	Optionally, you can use this parameter to remove passwordless SSH access to the Oracle Exadata patch nodes from the database server node from where the patch manager is run before exit.

Table F-62 (Cont.) rhpctl add workingcopy Command Parameters

Parameter	Description
-smtpfrom "address"	Optionally, you can specify an email address enclosed in double quotation marks (" ") from which Fleet Patching and Provisioning sends patch manager notifications.
-smtpo "addresses"	Optionally, you can specify several email address enclosed in double quotation marks (" ") to which Fleet Patching and Provisioning sends patch manager notifications.
-precheckonly	Optionally, you can use this parameter to run only the patching precheck operations.
-modifyatprereq	Optionally, you can use this parameter to run the patchmgr -precheck command with the -modify_at_prereq option for database nodes.
-resetforce	Optionally, you can use this parameter to run the patchmgr -reset_force command before cell patching.
-force	Optionally, you can use this parameter to run the patchmgr upgrade command with the -force option for ibswitches.

Usage Notes

**Note:**

Effective with Oracle Grid Infrastructure 20c, Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported. However, Domain Services Clusters continue to support Members Clusters in releases previous to Oracle Grid Infrastructure 20c.

Member Clusters from previous releases are converted to Standalone Clusters using remote services (on the Domain Services Cluster) when upgraded to 20c.

**Note:**

Starting with Oracle Grid Infrastructure 20c, Domain Services Cluster (DSC), which is part of the Oracle Cluster Domain architecture, are deprecated and can be desupported in a future release.

- You can obtain context sensitive help for specific use cases for the rhpctl add workingcopy command, as follows:

```
$ rhpctl add workingcopy -help [REMOTEPROVISIONING | STORAGETYPE |
ADMINDB
| GRIDHOMEPROV | SWONLYGRIDHOMEPROV | STANDALONEPROVISIONING |
GGHOMEPROVISIONING]
```

- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Examples

- To create a working copy on a client cluster for yourself or another user:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
  -series series_name} -oraclebase oracle_base_path -client cluster_name
  [-user user_name]
```

- To create a working copy on storage that you specify:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
  -series series_name} -oraclebase oracle_base_path -storagetype
  {LOCAL | RHP_MANAGED} [-path absolute_path]
```

- To create and configure a working copy of Oracle Grid Infrastructure:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
  -series series_name} {-root | -cred cred_name | -sudouser sudo_user_name
  -sudopath sudo_binary_path} -responsefile response_file_path
  [-clusternodes node_information] [-user user_name] [-oraclebase oracle_base_path]
  [-path absolute_path] [-asmclientdata data_path] [-gnsclientdata data_path]
  [-ignoreprereq | -fixup]
```

- To provision a software-only working copy of Oracle Grid Infrastructure:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
  -series series_name} -softwareonly -path Grid_home_path -oraclebase
  oracle_base_path [-local | -client cluster_name]
  [-groups "Oracle_group=user_group[,...]"] [-node client_node_name] |
  {-root | -cred cred_name | -sudouser sudo_user_name -sudopath sudo_binary_path}
  -targetnode node_name]
```

- To provision a working copy on a node or a cluster where Fleet Patching and Provisioning does not exist:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
  -series series_name} -oraclebase oracle_base_path -user user_name
  -node node_name [-path absolute_path]
  {-root | -cred cred_name | -sudouser sudo_user_name -sudopath sudo_binary_path}
```

 **Note:**

If you are provisioning Oracle database software to a Fleet Patching and Provisioning Client that has been configured with an Oracle ASM disk group, then do not specify the `-path` parameter, so as to enable the Fleet Patching and Provisioning Client to use storage provided by Fleet Patching and Provisioning.

If the Fleet Patching and Provisioning Client is not configured with an Oracle ASM disk group, then specify the `-storagetype` parameter with `LOCAL`, in addition to specifying the `-path` parameter.

rhpctl addnode workingcopy

Extends an Oracle RAC database to another node or nodes in a cluster.

Syntax

```
rhpctl addnode workingcopy -workingcopy workingcopy_name -node node_list
  [-targetnode node_name {-root | -sudouser sudo_username -sudopath
  sudo_binary_path
  | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]} -
  setupssh]
  [-ignoreprereq] [-useractiondata user_action_data] [-eval]
```

Parameters

Table F-63 rhpctl addnode workingcopy Command Parameters

Parameter	Description
<code>-workingcopy</code> <i>workingcopy_name</i>	Specify the name of a working copy that contains the Oracle database you want to extend.
<code>-node</code> <i>node_list</i>	Specify a node or a comma-delimited list of nodes to which you want to extend the database.
<code>-targetnode</code> <i>node_name</i>	Optionally, you can specify a node on which to run this command.
<code>-root</code> <code>-sudouser</code> <i>sudo_username</i> - <code>-sudopath</code> <i>sudo_binary_path</i> - <code>-cred</code> <i>cred_name</i> - <code>-auth</code> <i>plugin_name</i> [- <code>-arg1</code> <i>name1:value1...</i>]	If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>sudo</code> or <code>root</code> to access the remote nodes. If you choose <code>sudo</code> , then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary. Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. Alternative to <code>-sudouser</code> , <code>-root</code> , or <code>-cred</code> , you can use <code>-auth</code> to specify an authentication plugin to access a remote node.
<code>-ignoreprereq</code>	Use this parameter to ignore the CVU prerequisite checks.

Table F-63 (Cont.) rhpctl addnode workingcopy Command Parameters

Parameter	Description
-setupssh	Sets up passwordless SSH user equivalence on the remote nodes for the provisioning user.
-useractiondata <i>user_action_data</i>	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
-eval	Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command.

Usage Notes

- If you are extending a policy-managed database, then the database automatically starts on the new nodes.
- If you are extending an administrator-managed database, then you must also run the `rhpctl addnode database` command to start the instance.
- If the target cluster is an Oracle Clusterware 11g release 2 (11.2) or 12c release 1 (12.1) cluster, then you must provide either root credentials or provide a sudo user. You must also specify a target node that must be the node name of one of the cluster nodes.

rhpctl delete workingcopy

Deletes an existing working copy.

Syntax

```
rhpctl delete workingcopy -workingcopy workingcopy_name [-notify [-cc
user_list]] [-force]
  [[-targetnode node_name] {-root | -sudouser sudo_user_name -sudopath
  sudo_binary_path -cred cred_name | -auth plugin_name [-arg1
  name1:value1...]}
  [-useractiondata user_action_data] [-schedule timer_value]
```

Parameters**Table F-64 rhpctl delete workingcopy Command Parameters**

Parameter	Description
-workingcopy <i>workingcopy_name</i>	Specify the name of a working copy that you want to delete.
-notify [-cc <i>user_list</i>]	Name of a node in a remote cluster with no Fleet Patching and Provisioning Client.
-targetnode <i>node_name</i>	Optionally, you can specify a particular node from which you want to delete a working copy.
-force	Use this parameter to forcibly delete the database working copy.

Table F-64 (Cont.) rhpctl delete workingcopy Command Parameters

Parameter	Description
-root -sudouser sudo_username - sudopath sudo_binary_path - cred cred_name	<p>If you choose to use the <code>-targetnode</code> parameter, then you must choose either <code>sudo</code> or <code>root</code> to access the remote node.</p> <p>If you choose <code>sudo</code>, then you must specify a user name to run super-user operations, and a path to the location of the <code>sudo</code> binary.</p> <p>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.</p> <p>Alternative to <code>-sudouser</code>, <code>-root</code>, or <code>-cred</code>, you can use <code>-auth</code> to specify an authentication plugin to access a remote node.</p>
-useractiondata user_action_data	Optionally, you can pass a value to the <code>useractiondata</code> parameter of the user action script.
-schedule timer_value	<p>Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:</p> <p>2018-07-25T19:13:17+05</p>

Usage Notes

- This command *will not* delete the working copy if there are any databases configured on it. Use the `-force` option to override this.
- This command *will not* delete the working copy if there are any running databases on it. The `-force` option *will not* override this.
- This command *does not* delete the Oracle base that was created when you ran `rhpctl add workingcopy`.
- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Examples

To delete a working copy:

```
$ rhpctl delete workingcopy -workingcopy wcl
```

rhpctl query workingcopy

Displays the configuration information of an existing working copy.

Syntax

```
rhpctl query workingcopy [-workingcopy workingcopy_name [-metadataonly] |  
[-image image_name [-drift]] [-client cluster_name]]  
[-rhpserver rhps_regex]
```

Parameters

Table F-65 rhpctl query workingcopy Command Parameters

Parameter	Description
<code>-workingcopy</code> <code>workingcopy_name</code>	Specify the name of a working copy for which you want to display the configuration information.
<code>-metadataonly</code>	Use this parameter only when you use the <code>-workingcopy</code> parameter to query only the metadata of the working copy, which is located in the repository and not run OPatch or connect to the target to query for extra information.
<code>-image image_name [-drift]</code>	Use this parameter to specify the name of a configured image that you want to query. If you specify an image name, then RHPCTL lists all the working copies based on that image. The <code>-drift</code> option lists the bug fixes not included in the golden image.
<code>-client cluster_name</code>	Optionally, you can specify a client cluster on which to query working copies.
<code>-rhpserver rhps_regex</code>	Specify a regular expression to match the cluster name of the servers where the operation must be executed.

Usage Notes

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles  
role=user_name[+user_name...][,role=user_name[+user_name...]]...
```

For information about granting roles with RHPCTL, refer to [rhpctl grant role](#)

rhpctl update workingcopy

Updates a working copy on a client cluster.

 Note:

You can only use this command with Oracle Exadata.

Syntax

```
rhpctl update workingcopy -image image_name -root {[-dbnodes db_node_list]  
[-cells cell_list] [-ibswitches ibswitch_list]}
```

```
[-fromnode node_name] [-unkey] [-smtpfrom "address"] [-smtpto
"addresses"]
[-precheckonly] [-modifyatprereq] [-resetforce] [-force]
```

Parameters

Table F-66 rhpctl update workingcopy Command Parameters

Parameter	Description
-image <i>image_name</i>	Specify the name of a configured image from which to update a working copy
-dbnodes <i>db_node_list</i>	You can use this parameter to specify a file containing the list of database servers that you want to patch, one node name in each line.
-cells <i>cell_list</i>	Optionally, you can use this parameter to specify a file containing a list of storage cells that you want to patch, one cell name in each line.
-ibswitches <i>ibswitch_list</i>	Optionally, you can use this parameter to specify a file containing a list of InfiniBand switches that you want to patch, one switch name in each line.
-fromnode <i>node_name</i>	Optionally, you can specify a database server node name from where the patch manager is to be run.
-unkey	Optionally, you can use this parameter to remove passwordless SSH access to the Oracle Exadata patch nodes from the database server node from where the patch manager is run before exit.
-smtpfrom " <i>address</i> "	Optionally, you can specify an email address enclosed in double quotation marks (" ") from which Fleet Patching and Provisioning sends patch manager notifications.
-smtpto " <i>addresses</i> "	Optionally, you can specify several email address enclosed in double quotation marks (" ") to which Fleet Patching and Provisioning sends patch manager notifications.
-precheckonly	Optionally, you can use this parameter to run only the patching precheck operations.
-modifyatprereq	Optionally, you can use this parameter to run the patchmgr -precheck command with the -modify_at_prereq option for database nodes.
-resetforce	Optionally, you can use this parameter to run the patchmgr -reset_force command before cell patching.
-force	Optionally, you can use this parameter to run the patchmgr upgrade command with the -force option for ibswitches.

G

Server Control (SRVCTL) Command Reference

Use the Server Control (SRVCTL) utility to manage various components and applications in your cluster.

This appendix contains reference information for Server Control commands, including utility usage information and a comprehensive listing of the SRVCTL commands.

Note:

SRVCTL commands specific to Oracle RAC administration operations are documented in *Oracle Real Application Clusters Administration and Deployment Guide*

SRVCTL Usage Information

SRVCTL is installed on each node in a cluster by default. To use SRVCTL, log in to the operating system of a node and enter the SRVCTL command and its parameters in case-sensitive syntax.

- Only use the version of SRVCTL that is provided with Oracle Database 12c on Oracle RAC databases that are created for or upgraded to Oracle Database 12c. The version of SRVCTL must be the same as the version of the object (listeners, Oracle ASM instances, Oracle RAC databases and their instances, and services) being managed. To ensure the versions are the same release, run SRVCTL commands from the Oracle home of the database or object you are managing.
- SRVCTL does not support concurrent executions of commands on the same object. Therefore, run only one SRVCTL command at a time for each database, service, or other object.
- When specifying a comma-delimited list as part of a SRVCTL command, there should not be any spaces between the items in the list. For example:

```
srvctl add database -serverpool "serverpool1,serverpool3"
```

When you specify a comma-delimited list in a Windows environment, you must enclose the list in double quotation marks (" "). You can enclose a comma-delimited list in double quotation marks in a Linux or UNIX environment but they will be ignored.

- If you are entering a SRVCTL command, and you want to continue the input on a new line, then you can use the operating system continuation character. In Linux, this is the backslash (\) symbol.

- A SRVCTL command that produces no output is a successful command. Not all SRVCTL commands return a message when it completes, successfully. However, if a SRVCTL command fails, then it always returns an error message.
- You can use the `-eval` parameter with several SRVCTL commands. This parameter, when you use it, enables you to simulate running a command without making any changes to the system. SRVCTL returns output that informs you what will happen if you run a particular command. For example, to know what might happen if you relocate a server:

```
$ srvctl relocate server -servers "rac1" -eval -serverpool pool2

Database db1
  will stop on node rac1
  will start on node rac7
Service mySrv1
  will stop on node rac1, it will not run on any node
Service myServ2
  will stop on node rac1
  will start on node rac6
Server rac1
  will be moved from pool myPoolX to pool pool2
```

The `-eval` parameter is available with the following commands:

- `srvctl add database`
- `srvctl add service`
- `srvctl add srvpool`
- `srvctl modify database`
- `srvctl modify service`
- `srvctl modify srvpool`
- `srvctl relocate server`
- `srvctl relocate service`
- `srvctl remove srvpool`
- `srvctl start database`
- `srvctl start service`
- `srvctl stop database`
- `srvctl stop service`

Specifying Command Parameters as Keywords Instead of Single Letters

Before Oracle Database 12c, the SRVCTL command-line interface used single letter parameters. However, this imposes a limit on the number of unique parameters available for use with SRVCTL commands. SRVCTL command parameters introduced in Oracle Database 12c are full words instead of single letters, such as `-multicastport` and `-subdomain`.

To support backward compatibility, you can use a mix of single-letter parameters and new keyword parameters. New parameters introduced with keywords can be used with single letter parameters.

 **Note:**

Starting with Oracle Database 12c, the single letter parameters are deprecated in favor of the keyword parameters to avoid using the same letter to implement different functionality depending on the command.

You can obtain the single-letter equivalents, where applicable, by adding the `-compatible` parameter after the `-help` parameter.

Character Set and Case Sensitivity of SRVCTL Object Values

SRVCTL interacts with many different types of objects. The character set and name length limitations, and whether the object name is case sensitive, can vary between object types.

Table G-1 String Restrictions for SRVCTL Object Names

Object Type	Character Set Limitations	Case Sensitive ?	Maximum Length
db_domain	Alpha-numeric characters, underscore (<code>_</code>), and number sign (<code>#</code>)	No	128 characters
db_unique_name	Alpha-numeric characters, underscore (<code>_</code>), number sign (<code>#</code>), and dollar sign (<code>\$</code>); the first 8 characters must be unique because those characters are used to form instance names for policy-managed databases	No	30 characters but the first 8 characters must be unique relative to any other database in the same cluster
diskgroup_name	Naming disk groups have the same limitations as naming other database objects.	No (all names are converted to uppercase)	30 characters but the first 8 characters must be unique relative to any other database in the same cluster
instance_name	Alphanumeric characters	Depends on the platform	15 characters
listener_name	Alphanumeric characters	Depends on the platform	15 characters
node_name	Alphanumeric characters	No	15 characters

Table G-1 (Cont.) String Restrictions for SRVCTL Object Names

Object Type	Character Set Limitations	Case Sensitive ?	Maximum Length
scan_name	The first character must be an alphabetic character	No	15 characters
server_pool	Alphanumeric characters, underscore (_), number sign (#), period (.), and dollar sign (\$); the name cannot begin with a period, contain single quotation marks (' '), nor can the name be "Generic" or "Free" because those two names are reserved for the built-in server pools	No	250 characters
service_name	Alphanumeric characters, underscore (_), number sign (#), period (.), and dollar sign (\$); the name cannot begin with a period, contain single quotation marks (' '), nor can the name be "Generic" or "Free" because those two names are reserved for the built-in server pools	No	250 characters
volume_name	Alphanumeric characters; dashes (-) are not allowed and the first character must be an alphabetic character.	No	11 characters

Using SRVCTL Help

This section includes information about using context sensitive help with SRVCTL.

To see help for all SRVCTL commands, from the command line enter:

```
srvctl -help
```

To see the command syntax and a list of parameters for each SRVCTL command, from the command line enter:

```
srvctl command (or verb) object (or noun) -help
```

When you request online help for a command using `-help`, SRVCTL prints the full words for each parameter. You can obtain the single-letter equivalents, where applicable, by adding the `-compatible` parameter after the `-help` parameter. For example:

```
$ srvctl config database -help -compatible
```


The preceding command prints usage information for the `srvctl config database` command, listing all parameters as full words followed by their single-letter equivalents in parentheses, where applicable.

To see the SRVCTL version number enter:

```
$ srvctl -version
```

SRVCTL Privileges and Security

To use SRVCTL to change your Oracle RAC database configuration, log in to the operating system as the software owner of the home that you want to manage.

For example, if different users installed Oracle Database and the Oracle Grid Infrastructure, then log in as the database software owner (for example, `ora_db`) to manage databases and log in as the Oracle Grid Infrastructure software owner (for example, `ora_asm`) to manage the Oracle ASM instances.

Users who are members of the OSDBA operating system group can start and stop the database. To stop and start an Oracle ASM instance, you must be a member of the OSASM operating system group.

To create or register objects such as listeners, Oracle Notification Services, and services, you must be logged in to the operating system as the software owner of the Oracle home. The objects you create or register for that Oracle home will run under the user account of the owner of the Oracle home. Databases run as the database installation owner of the home from which they run.

To perform `srvctl add` operations on any object, you must be logged in as the Oracle account owner of the home on which the object runs.

For some SRVCTL commands, on Linux and UNIX systems, you must be logged in as `root`, and on Windows systems, you must be logged in as a user with Administrator privileges to run them. In this appendix, those commands are preceded by the root prompt (`#`) in the command examples.

Additional SRVCTL Topics

- Use SRVCTL to manage Oracle-supplied resources such as listener, instances, disk groups, and networks, and CRSCTL for managing Oracle Clusterware and its resources.

Note:

Oracle strongly discourages directly manipulating Oracle-supplied resources (resources whose names begin with `ora`) using CRSCTL. This could adversely impact the cluster configuration.

- Although you may be able to cancel running SRVCTL commands by pressing the Control-C keys, you may corrupt your configuration data by doing this.

You are strongly advised *not* to attempt to terminate SRVCTL in this manner.

Deprecated SRVCTL Subprograms or Commands

A number of SRVCTL commands and parameters have been deprecated in this release.

Single Character Parameters for all SRVCTL Commands

Single-character parameters have been deprecated in Oracle Database 12c.

Use the full keyword for each parameter instead. To support older tools and scripts that still use single-character parameters, the current version of SRVCTL supports both single-character parameters and full keyword parameters.

The command reference in this appendix shows the keywords for each SRVCTL command. [Table G-2](#) lists the deprecated single-character parameters.

Table G-2 Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
A	address	{VIP_name IP}/netmask/[if1[if2...]]	VIP address specification for node applications	Node applications, VIP, network, Listener, SCAN VIP, and SCAN listener commands
a	all		All resources of that kind	srvctl config database Common
a	diskgroup	diskgroup_list	Comma-delimited list of Oracle ASM disk groups	Database, instance, Oracle ASM, disk group, and file system commands
a	detail		Print detailed configuration information	Common
a	available	available_list	A comma-delimited list of available instances	Service and server pool commands
a	abort		Abort failed online relocation	Relocate database
a	viponly		Display VIP configuration	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
B	rlbgoal	{NONE SERVICE_TIME THROUGHPUT}	The runtime load balancing goal of a service	Service and server pool commands
c	currentnode	current_node	Node name from which to relocate the service	Service and server pool commands

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
c	cardinality	{UNIFORM SINGLETON}	Whether the service should run on every active server in the server pool (UNIFORM) or just one server (SINGLETON)	Service and server pool commands
c	dbtype	<i>type</i>	Type of database: Oracle RAC One Node, Oracle RAC, or single instance	Database, instance, Oracle ASM, disk group, and file system commands
d	db or database	<i>db_unique_name</i>	Database unique name	Common
d	device	<i>volume_device</i>	Volume device path	Database, instance, Oracle ASM, disk group, and file system commands
d	domain		Display subdomain served by GNS	OC4J, home, CVU, and GNS commands
e	emport	<i>em_port_number</i>	Local listen port for Oracle Enterprise Manager	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
e	failover type	{NONE SESSION BASIC TRANSACTION}	The failover type for a service	Service and server pool commands
e	server	<i>server_list</i>	Candidate server list for Oracle RAC One Node database	Database, instance, Oracle ASM, disk group, and file system commands
f	force		Force remove	Common
g	diskgroup	<i>diskgroup_name</i>	Disk group name	File system, Diskgroup commands
g	gsdonly		Display GSD configuration	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
g	serverpool	<i>server_pool_name</i> <i>server_pool_list</i>	A server pool name Comma-delimited list of database server pool names	Service and server pool commands Database, instance, Oracle ASM, disk group, and file system commands
h	help			Common

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
i	importance	<i>number</i>	A number that represents the importance of the server pool	Service and server pool commands
i	instance	<i>instance_name</i> <i>instance_list</i>	Instance name prefix for administrator-managed Oracle RAC One Node database A comma-delimited list of instance names	Database, instance, Oracle ASM, disk group, and file system commands
I	ip	<i>ip_address</i>	VIP address on which GNS is to listen	OC4J, home, CVU, and GNS commands
i	oldinst	<i>instance_name</i>	The old instance name	Service and server pool commands
i	scannumber	<i>scan_ordinal_number</i>	Ordinal number of the IP address for the SCAN	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
i	vip	<i>vip_name</i> or " <i>vip_name_list</i> "	VIP names	Node applications, GNS, VIP, network, listener, SCAN VIP, and SCAN listener commands
j	acfspath	<i>acfs_path_list</i>	Comma-delimited list of Oracle ACFS paths where the dependency on the database will be set	Database, instance, Oracle ASM, disk group, and file system commands
j	clbgoal	{SHORT LONG}	The connection load balancing goal for a service	Service and server pool commands
k	netnum	<i>network_number</i>	The network number	Service and server pool commands Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands OC4J, home, CVU, and GNS commands
l	list		List all records in GNS	OC4J, home, CVU, and GNS commands
l	listener	<i>listener_name</i>	The name of a listener	ASM commands
l	loglevel	<i>log_level</i>	Specify the level (0-6) of logging that GNS should run with	OC4J, home, CVU, and GNS commands
l	min	<i>number</i>	The minimum size of the server pool	Service and server pool commands

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
l	onslocalport	<i>port_number</i>	Oracle Notification Service listening port for local client connections	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
l	role	<i>service_role</i>	Comma-delimited list of server roles within double quotation marks (" "), where each role is one of PRIMARY, PHYSICAL_STANDBY, LOGICAL_STANDBY, or SNAPSHOT_STANDBY	Service and server pool commands
m	domain	<i>domain_name</i>	The domain for the database	Database, instance, Oracle ASM, disk group, and file system commands
m	failovermethod	{NONE BASIC}	The failover method of a service	Service and server pool commands
m	multicastport		The port on which the GNS daemon is listening for multicast requests	OC4J, home, CVU, and GNS commands
m	path	<i>mountpoint_path</i>	Mountpoint path	Database, instance, Oracle ASM, disk group, and file system commands
n	name		Advertise a name through GNS using the given address	OC4J, home, CVU, and GNS commands
n	node	<i>node_name</i>	The name of a specific node	Common
n	nodes	<i>node_list</i>	A comma-delimited list of node names	File system commands
n	dbname	<i>database_name</i>	The database name (DB_NAME), if different from the unique name specified by the -db parameter	Database, instance, Oracle ASM, disk group, and file system commands
n	scanname	<i>scan_name</i>	Fully-qualified SCAN name (includes the domain)	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
n	servers	<i>server_list</i>	A comma-delimited list of candidate server names	Service and server pool commands

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
n	targetnode	<i>node_name</i>	Node name to which to relocate the service	Service and server pool commands
o	oraclehome	<i>oracle_home</i>	<i>\$ORACLE_HOME</i> path	Database commands
p	endpoints	[TCP:] <i>port_number</i> [/IPC: <i>key</i>][/NMP: <i>pipe_name</i>][/TCPS: <i>s_port</i>][/SDP: <i>port</i>]	SCAN listener endpoints	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
p	port		The port which the GNS daemon uses to communicate with the DNS server	OC4J, home, CVU, and GNS commands
p	rmiport	<i>port_number</i>	OC4J RMI port number	OC4J, home, CVU, and GNS commands
P	tafpolicy	{NONE BASIC}	TAF policy specification	Service and server pool commands
p	spfile	<i>spfile_location</i>	Server parameter file path	Database, instance, Oracle ASM, disk group, and file system commands
q	notification	{TRUE FALSE}	Whether FAN is enabled for OCI connections	Service commands
q	query		Query GNS for the records belonging to a name	OC4J, home, CVU, and GNS commands
r	preferred	<i>preferred_list</i>	A comma-delimited list of preferred instances	Service and server pool commands
r	onsremotep ort	<i>port_number</i>	Oracle Notification Service listening port for connections from remote hosts	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
r	relocate		Relocate the VIP	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
r	revert		Remove target node of failed online relocation request from the candidate server list of administrator-managed Oracle RAC One Node database	Relocate database

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
r	role	<i>role_type</i>	Role of the standby database: PRIMARY, PHYSICAL_STANDBY, LOGICAL_STANDBY, or SNAPSHOT_STANDBY	Database, instance, Oracle ASM, disk group, and file system commands
s	ononly		Display Oracle Notification Service daemon configuration	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
s	skip		Skip checking the ports	Listener, SCAN, and SCAN listener.
s	statfile	<i>file_name</i>	The file path of the <i>state_file</i> created by a previously executed <code>srvctl stop home</code> command	OC4J, home, CVU, and GNS commands
s	status		Display the status of GNS	OC4J, home, CVU, and GNS commands
S	subnet	<i>subnet/net_mask/[if1[if2...]]</i>	Network address specification for a network	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
s	service	<i>service_name</i> <i>service_name_list</i>	The name of a service A comma-delimited list of service names	Service and server pool commands
s	startoption	<i>start_options</i>	Startup options for the database (mount, open, read only)	Database, instance, Oracle ASM, disk group, and file system commands
t	checkinterval	<i>time_interval</i>	Interval in minutes between checks	OC4J, home, CVU, and GNS commands
t	edition	<i>edition_name</i>	The initial session edition of a service	Service and server pool commands
t	envs	" <i>name_list</i> "	A list of environment variables	Common
t	namevals	" <i>name=value,...</i> "	Names and values of environment variables	Common
T	nameval	" <i>name=value</i> "	Name and value of a single environment variable	Common
t	update	<i>instance_name</i>	The new instance name	Service and server pool commands

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
t	remoteservers	<i>host_name[:port_number][,host_name[:port_number]...]</i>	List of remote host name and port number pairs for Oracle Notification Service daemons outside this cluster	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
t	stopoption	<i>stop_options</i>	Stop options for the database (NORMAL, TRANSACTIONAL, IMMEDIATE, or ABORT)	Database, instance, Oracle ASM, disk group, and file system commands
t	toversion	<i>target_version</i>	Version to which you are downgrading	Database, instance, Oracle ASM, disk group, and file system commands
u	max	<i>number</i>	Maximum size of the server pool	Service and server pool commands
u	nettype	<i>network_type</i>	The network server type, which can be STATIC, DHCP, or MIXED	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
u	newinst		Add a new instance to the service configuration	Service commands
u	update		Update SCAN listeners to match the number of SCAN VIPs	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
u	user	<i>oracle_user</i>	Oracle user or other authorized user to mount and unmount file systems	Database, instance, Oracle ASM, disk group, and file system commands
v	verbose		Verbose output	Common
v	volume	<i>volume_name</i>	Name of a volume	Database, instance, Oracle ASM, disk group, and file system commands
V	versions			Common
w	failoverdelay	<i>number</i>	Failover delay	Service and server pool commands
w	nettype	<i>network_type</i>	The network server type, which can be STATIC, DHCP, or MIXED	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands

Table G-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
w	timeout	<i>timeout</i>	Online relocation timeout in minutes	Database, instance, Oracle ASM, disk group, and file system commands
x	dtp	{TRUE FALSE}	Whether to enable distributed transaction processing	Service and server pool commands
x	node	<i>node_name</i>	Node name (use this parameter only with noncluster databases)	Common
y	noprompt		Suppress the confirmation prompt	Common
y	policy	{AUTOMATIC MANUAL}	Management policy for the resource	Database, instance, Oracle ASM, disk group, file system, service and server pool commands
z	failoverretry	<i>number</i>	Number of failover retries	Service and server pool commands
z	rmdepondisk		To remove a database's dependency upon disk groups	Database, instance, Oracle ASM, disk group, and file system commands

Miscellaneous SRVCTL Commands and Parameters

The following command parameters have been deprecated in this release:

Table G-3 Deprecated Commands and Parameters for SRVCTL

Command	Deprecated Parameters
srvctl modify asm	-node <i>node_name</i>
srvctl modify instance	-z Instead, use the -node option with the value set to ""
srvctl modify gns	[-ip <i>ip_address</i>] [-advertise <i>host_name</i> -address <i>address</i>] [-delete <i>host_name</i> -address <i>address</i>] [-createalias <i>name</i> -alias <i>alias</i>] [-deletealias <i>alias</i>] Use the srvctl update gns command instead.
srvctl * oc4j	The oc4j noun has been deprecated and replaced with qosmsserver. SRVCTL still accepts the oc4j noun until it is desupported.
srvctl add service	The PRECONNECT option with the -tafpolicy parameter is deprecated.

Table G-3 (Cont.) Deprecated Commands and Parameters for SRVCTL

Command	Deprecated Parameters
srvctl modify service	The <code>-failovermethod {NONE BASIC}</code> is deprecated. The <code>PRECONNECToption</code> with the <code>-tafpolicy</code> parameter is deprecated.

SRVCTL Command Reference

A comprehensive list of SRVCTL commands to use in Oracle Grid Infrastructure environments.

SRVCTL uses the following command syntax:

```
srvctl command object [parameters]
```

In SRVCTL syntax:

- *command* is a verb such as `start`, `stop`, or `remove`
- *object* (also known as a *noun*) is the target or object on which SRVCTL performs the command, such as Oracle ASM or a file system. You can also use object abbreviations.
- *parameters* extend the use of a preceding command combination to include additional parameters for the command. Some parameters are required while others are optional.

Note:

If specifying a comma-delimited list in a command parameter on Windows, then you must enclose the list within double quotation marks (" ").

Table G-4 lists the keywords that can be used for the *object* portion of SRVCTL commands. You can use either the full name or the abbreviation for each object keyword. The **Purpose** column describes the object and the actions that can be performed on that object.

Table G-4 Object Keywords and Abbreviations

Object	Keyword	Purpose
Oracle ASM	asm	To add, modify, manage environment variables for, list the configuration of, enable, disable, start, stop, and obtain the status of Oracle ASM
Oracle ASM Network	asmnetwork	To add, configure, modify and remove an Oracle ASM network configuration

Table G-4 (Cont.) Object Keywords and Abbreviations

Object	Keyword	Purpose
Cluster Verification Utility	cvu	To add, modify, enable, disable, remove, relocate, start, stop, and obtain the status of the Cluster Verification Utility
Cross-cluster dependency	cdp	To add, modify, enable, disable, remove, relocate, start, stop, and obtain the status of cross-cluster dependency services.
Cross-cluster dependency proxies	cdpproxy	To add, modify, enable, disable, remove, relocate, start, stop, and obtain the status of cross-cluster dependency proxy resources.
Export file system	exportfs	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, and remove export file systems
Oracle Automatic Storage Cluster File System	filesystem	To add, modify, list the configuration of, remove, start, stop, and obtain the status of Oracle ACFS
Grid Naming Service	gns	To add, modify, list the configuration of, import, export, enable, disable, start, stop, obtain the status of, relocate, remove, and update the Grid Naming Service
Highly available VIP	havip	To add, list the configuration of, enable, start, obtain the status of, stop, disable, relocate, and remove highly available VIPs
Oracle ASM IO Server	ioserver	To add, list the configuration of, modify, enable, disable, start, stop, relocate, obtain the status of, manage environment variables for, and remove Oracle ASM IO Server
Management database	mgmtldb	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, manage environment variables for, and remove management databases (Cluster Health Monitor repositories)
Management listener	mgmtlsnr	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, manage environment variables for, and remove management listener resources for Cluster Health Monitor
Network Attached Storage (NAS)	mountfs	To add, list the configuration of, enable, start, obtain the status of, stop, disable, relocate, and remove NAS
Network Storage Service Resource	netstorageservice	To add, list the configuration of, enable, start, obtain the status of, stop, disable, and remove NetStorageService Resource
Virtual machine manager	ovmm	To add, list the configuration of, enable, modify, and remove a virtual machine manager
Oracle Database Quality of Service Management Server	qosmsserver	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, and remove Oracle Database Quality of Service Management Server

Table G-4 (Cont.) Object Keywords and Abbreviations

Object	Keyword	Purpose
Fleet Patching and Provisioning Client	rhpcient	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, and remove Fleet Patching and Provisioning Clients
Fleet Patching and Provisioning Server	rhpserver	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, and remove Fleet Patching and Provisioning Servers
Virtual machine	vm	To add, modify, list the configuration of, enable, disable, start, stop, obtain the status of, relocate, and remove virtual machines

 **Note:**

SRVCTL commands specific to Oracle RAC administration operations are documented in *Oracle Real Application Clusters Administration and Deployment Guide*

srvctl add asm

Adds a record for an Oracle ASM instance to the entire cluster.

This command must be run only one time from the Oracle Grid Infrastructure home.

Syntax

```
srvctl add asm [-listener listener_name] [-pwfile password_file_path
               [-flex [-count {number_of_instances | ALL}] | -proxy [-spfile
server_parameter_file_path]]]
```

Parameters

Table G-5 srvctl add asm Command Parameters

Parameter	Description
-listener listener_name	Enter the name of a listener. If you do not specify this parameter, then the listener name defaults to LISTENER.
-pwfile password_file_path	Enter the full path to the location of the password file on Oracle ASM.

Table G-5 (Cont.) srvctl add asm Command Parameters

Parameter	Description
-flex [-count { <i>number_of_instances</i> ALL}] -proxy [-spfile <i>server_parameter_file_path</i>]	Specify that the Oracle ASM instances are configured as Oracle Flex ASM instances. You can optionally specify the number of Oracle ASM instances to create or that an Oracle ASM instance should be created on all nodes. Alternatively, you can specify the -proxy parameter to configure the Oracle ASM Dynamic Volume Manager proxy instance resource. Optionally, you can specify the path to the server parameter file.

Usage Notes

To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Example

To add a clusterware resource for Oracle ASM to every node in the cluster, use the following command:

```
$ srvctl add asm
```

srvctl config asm

Displays the configuration for all Oracle ASM instances.

Syntax

```
srvctl config asm [-proxy] [-detail]
```

Parameters**Table G-6 srvctl config asm Command Parameters**

Parameter	Description
-proxy	Optionally, you can use this parameter to specify whether the Oracle ASM configuration to display is a proxy instance.
-detail	Optionally, you can use this parameter to print detailed configuration information.

Usage Notes

- Use this command only with Oracle Clusterware.
- To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have

Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Example

This command returns output similar to the following:

```
$ srvctl config asm

ASM Home: /u01/app/12.1.0/grid
ASM Listener: LISTENER
ASM instance count: 3
Password file:
```

srvctl disable asm

Disables the Oracle ASM proxy resource. Oracle ASM will restart if it contains Oracle Clusterware data or if the node restarts and it was running before the node failed. The `srvctl disable asm` command also prevents the `srvctl start asm` command from starting Oracle ASM instances.

Disabling the Oracle ASM proxy resource prevents the databases and disk groups from starting because they depend on the Oracle ASM proxy resource.

Syntax

```
srvctl disable asm [-proxy] [-node node_name]
```

Parameters

Table G-7 `srvctl disable asm` Command Parameters

Parameter	Description
<code>-proxy</code>	Optionally, you can use this parameter to specify whether you are disabling a proxy Oracle ASM configuration.
<code>-node <i>node_name</i></code>	Optionally, you can specify a particular node on which you want to disable Oracle ASM.

Usage Notes

- To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.
- Oracle ASM is part of the Oracle Clusterware stack and when OCR and voting files are stored on Oracle ASM, then Oracle ASM starts when OHASD starts the Oracle Clusterware stack. The `srvctl disable asm` command does not prevent the Oracle ASM instance managed by OHASD from starting. Oracle ASM, therefore, starts as needed by the Oracle Clusterware stack.

The `srvctl disable asm` command prevents the Oracle ASM Oracle Clusterware proxy resource, and any resources that depend on it, from starting. So, the command prevents Oracle Clusterware-managed objects, such as databases, disk

groups, and file systems that depend on Oracle ASM, from starting. For example, the `srvctl start database | diskgroup | filesystem` command fails to start any of those objects on nodes where the Oracle ASM Oracle Clusterware proxy resource is disabled. The command also prevents the `srvctl start asm` command from starting Oracle ASM on remote nodes.

Example

The following example disables Oracle ASM on a node named `crmnode1`:

```
$ srvctl disable asm -node crmnode1
```

srvctl enable asm

Enables an Oracle ASM instance.

Syntax

```
srvctl enable asm [-proxy] [-node node_name]
```

Parameters

Table G-8 `srvctl enable asm` Command Parameters

Parameter	Description
<code>-proxy</code>	Optionally, you can use this parameter to specify whether you are enabling a proxy Oracle ASM configuration.
<code>-node <i>node_name</i></code>	Optionally, you can specify a particular node on which you want to enable Oracle ASM.

Usage Notes

To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Example

An example of this command is:

```
$ srvctl enable asm -node crmnode1
```

srvctl getenv asm

Displays the values for environment variables associated with Oracle ASM.

Syntax

```
srvctl getenv asm [-envs "name_list"]
```

Usage Notes

- If you choose not use the `-env` parameter, then the values of all environment variables associated with Oracle ASM are displayed.
- You can specify a comma-delimited list of the names of environment variables for which you want to retrieve the values.

Example

The following example displays the current values for all the environment variables used by Oracle ASM:

```
$ srvctl getenv asm
```

srvctl modify asm

Modifies the listener used by Oracle ASM, the disk group discovery string used by Oracle ASM, or the SPFILE used by Oracle ASM for a noncluster database or a cluster database.



Note:

To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Syntax

```
srvctl modify asm [-listener listener_name] [-pfile password_file_path]
  [-count {number_of_instances | ALL}] [-diskstring asm_diskstring]
  [-spfile spfile_path_name [-proxy] [-force]]
```

Parameters

Parameter	Description
<code>-listener</code> <i>listener_name</i>	Specify the name of the listener with which Oracle ASM registers.
<code>-pfile</code> <i>password_file_path</i>	Specify the full path to the location of the password file for Oracle ASM.
<code>-count</code> { <i>number_of_instances</i> ALL}	Specify the number of Oracle ASM instances to create, or that an Oracle ASM instance should be created on all nodes.
<code>-diskstring</code> <i>asm_diskstring</i>	Specify a new Oracle ASM disk group discovery string.
<code>-spfile</code> <i>spfile_path_name</i>	Specify the path to the new spfile to be used by Oracle ASM.

Parameter	Description
-proxy	Use this parameter to specify an Oracle ASM proxy instance.
-force	Use this parameter to force modification of Oracle ASM.

Example

An example of this command to modify the configuration of Oracle ASM is:

```
$ srvctl modify asm -listener lsnr1
```

srvctl predict asm

Predicts the consequences of Oracle ASM failure.

Syntax

```
srvctl predict asm [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify a particular node on which you want to predict the consequences of an Oracle ASM failure.
- You can choose verbose output.

Example

An example of this command to predict the consequences of an Oracle ASM failure on a particular node:

```
$ srvctl predict asm -node crmnode2
```

srvctl relocate asm

Relocates an Oracle Flex ASM instance from one Hub Node of an Oracle Flex Cluster to another.

Syntax

```
srvctl relocate asm -currentnode current_node [-targetnode target_node] [-force]
```

Parameters**Table G-9** srvctl relocate asm Parameters

Parameter	Description
-currentnode <i>current_node</i>	Name of the node on which the Oracle Flex ASM instance currently resides.

Table G-9 (Cont.) srvctl relocate asm Parameters

Parameter	Description
-targetnode <i>target_node</i>	Name of the node to which you want to relocate the Oracle Flex ASM instance.
-force	Force the relocation of the Oracle Flex ASM instance.

Usage Notes

- You cannot use this command if there is only one Oracle Flex ASM instance running.
- Because of bootstrap issues between Oracle ASM and Oracle Clusterware, on the node where Oracle Clusterware uses Oracle ASM run either `crsctl stop crs` or `crsctl stop cluster` to stop Cluster Ready Services on the node before you relocate the Oracle ASM instance.

Example

An example of this command to relocate an Oracle Flex ASM instance from the current node to a node named `hubnode2` is:

```
$ srvctl relocate asm -currentnode hubnode1 -targetnode hubnode2
```

srvctl remove asm

Removes the Oracle ASM resource from Oracle Clusterware management.

Note:

To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Syntax

```
srvctl remove asm [-proxy] [-force]
```

Parameters**Table G-10 srvctl remove asm Command Parameters**

Parameter	Description
-proxy	Optionally, you can use this parameter to specify whether you are removing a proxy Oracle ASM configuration.

Table G-10 (Cont.) srvctl remove asm Command Parameters

Parameter	Description
-force	Forcefully removes an Oracle ASM resource (ignores any dependencies).

Example

An example of this command is:

```
$ srvctl remove asm -force
```

srvctl setenv asm

Sets Oracle ASM environment configurations.

Syntax

Use this command with one of the following syntaxes:

```
srvctl setenv asm -envs "name=value,[...]"
```

```
srvctl setenv asm -env "name=value"
```

Parameters**Table G-11 srvctl setenv asm Command Parameters**

Parameter	Description
-envs "name=value, [...]"	Use this parameter to specify either a single name-value pair or a comma-delimited list of multiple name-value pairs of environment variables enclosed in double quotation marks (" ").
-env "name=value"	Use this parameter to enable a single environment variable to be set to a value that contains commas or other special characters, enclosed in double quotation marks (" ").

Example

The following example sets the language environment configuration for Oracle ASM:

```
$ srvctl setenv asm -envs "LANG=en"
```

srvctl start asm

Starts an Oracle ASM instance.



Note:

To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Syntax

```
srvctl start asm [-proxy] [-node node_name [-startoption start_options]]
```

Parameters

Table G-12 srvctl start asm Command Parameters

Parameter	Description
-proxy	Optionally, you can use this parameter to specify whether you are starting a proxy Oracle ASM configuration.
-node <i>node_name</i>	Optionally, you can specify a particular node on which you want to start Oracle ASM. Note: You can use this parameter only with Oracle Clusterware.
-startoption <i>start_options</i>	Optionally, you can specify options to the startup command, such as OPEN, MOUNT, or NOMOUNT Note: For multi-word startup options, such as read only and read write, separate the words with a space and enclose in double quotation marks (" "). For example, "read only". See Also: <i>SQL*Plus User's Guide and Reference</i> for more information about startup options

Examples

An example of this command to start an Oracle ASM instance on a single node of a cluster is:

```
$ srvctl start asm -node crmnode1
```

An example to start an Oracle ASM instance on all nodes in the cluster, or for a noncluster database, is:

```
$ srvctl start asm
```

srvctl status asm

Displays the status of an Oracle ASM instance.

Note:

To manage Oracle ASM on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ASM.

Syntax

```
srvctl status asm [-proxy] [-node node_name] [-detail] [-verbose]
```

Parameters

Table G-13 srvctl status asm Command Parameters

Parameter	Description
-proxy	Optionally, you can use this parameter to display the status for an Oracle ASM proxy instance.
-node <i>node_name</i>	Optionally, you can specify a particular node. If you do not specify this parameter, then SRVCTL displays the status of all Oracle ASM instances.
-detail	Print detailed status information.
-verbose	Displays verbose output.

Example

This command returns output similar to the following:

```
$ srvctl status asm -verbose  
  
ASM is running on node2, node3, node4  
iDetailed state on node node2: Started  
Detailed state on node node3: Started  
Detailed state on node node4: Started
```

srvctl stop asm

Stops an Oracle ASM instance.

Syntax

```
srvctl stop asm [-proxy] [-node node_name] [-stopoption stop_options] [-force]
```

Parameters

Table G-14 srvctl stop asm Command Parameters

Parameter	Description
-proxy	Optionally, you can use this parameter to specify whether you are stopping a proxy Oracle ASM configuration.
-node <i>node_name</i>	Optionally, you can specify a particular node on which you want to stop Oracle ASM. If you do not specify this parameter, then SRVCTL stops the Oracle ASM instance on every active node in the cluster. Note: You can use this parameter only with Oracle Clusterware.
-stopoption <i>stop_options</i>	Optionally, you can specify options to the shutdown command, such as NORMAL, TRANSACTIONAL, IMMEDIATE, or ABORT. See Also: <i>SQL*Plus User's Guide and Reference</i> for more information about shutdown options
-force	Optionally, you can use this parameter to stop disk groups, file systems, and databases that depend on Oracle ASM.

Usage Notes

- To manage Oracle ASM on Oracle Database 12c installations, use SRVCTL in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use SRVCTL in the database home to manage Oracle ASM.
- You cannot use this command when OCR is stored in Oracle ASM because it will not stop Oracle ASM. To stop Oracle ASM you must shut down Oracle Clusterware.

Example

An example of this command is:

```
$ srvctl stop asm -node crmnode1 -stopoption IMMEDIATE
```

srvctl unsetenv asm

Unsets the Oracle ASM environment configurations.

Syntax

```
srvctl unsetenv asm -envs "name_list"
```

Usage Notes

Use the `-envs` parameter to specify a comma-delimited list of the names of environment variables enclosed in double quotation marks ("").

Example

The following example unsets the environment configuration for an Oracle ASM environment variable:

```
$ srvctl unsetenv asm -envs "CLASSPATH"
```

srvctl add asmnetwork

Adds an Oracle ASM network configuration to Oracle Clusterware.

Syntax

```
srvctl add asmnetwork [-netnum asm_network_number] [-deptype {optional | mandatory}] -subnet subnet[/netmask/iface1[|iface2...]]
```

Parameters

Table G-15 `srvctl add asmnetwork` Command Parameters

Parameter	Description
<code>-netnum</code> <i>asm_network_number</i>	Optionally, you can specify an Oracle ASM network number (default number is 1) that you want to add.
<code>-deptype</code> {optional mandatory}	Specify whether the Oracle ASM dependency type is optional or mandatory. The default is <code>optional</code> .
<code>-subnet</code> <i>subnet[/netmask/iface1[iface2...]]</i>	Specify a subnet for the Oracle ASM network you want to add. If you do not specify any interface names, then the network uses any interface on the given subnet. For IPv6, <i>netmask</i> is a prefix length, such as 64.

srvctl config asmnetwork

Displays the configuration information for Oracle ASM networks registered with Oracle Clusterware.

Syntax

```
srvctl config asmnetwork [-netnum asm_network_number]
```

Usage Notes

Optionally, you can specify an Oracle ASM network number (default number is 1) for which you want to view configuration information.

srvctl modify asmnetwork

Modifies an Oracle ASM network configuration in Oracle Clusterware.

Syntax

```
srvctl modify asmnetwork [-netnum asm_network_number] [-deptype {optional | mandatory}]
```

Parameters

Table G-16 srvctl modify asmnetwork Command Parameter

Parameter	Description
-netnum <i>asm_network_number</i>	Optionally, you can specify an Oracle ASM network number (default number is 1) that you want to modify.
-deptype {optional mandatory}	Specify whether the Oracle ASM dependency type is optional or mandatory.

srvctl remove asmnetwork

Removes an Oracle ASM network configuration from Oracle Clusterware.

Syntax

```
srvctl remove asmnetwork {-netnum asm_network_number | -all} [-force]
```

Parameters

Table G-17 srvctl remove asmnetwork Command Parameters

Parameter	Description
-netnum <i>asm_network_number</i> -all	Specify an Oracle ASM network number (default number is 1) that you want to remove or specify -all to remove all Oracle ASM network numbers configured on Oracle Clusterware.
-force	Use this parameter to forcibly remove an Oracle ASM network regardless of any dependencies.

srvctl add cdp

Adds a cross-cluster dependency proxy resource to a cluster (only on the first public network).

Syntax

```
srvctl add cdp [-port port_number] [-passfile_admin afile] [-passfile_readonly rfile]
```

Parameters

Table G-18 `srvctl add cdp` Command Parameters

Parameter	Description
<code>-port <i>port_number</i></code>	Optionally, you can specify a port on which all cross-cluster dependency proxy resources in the cluster listen. If you do not specify a port, then SRVCTL chooses a port that is available on all nodes of the Domain Services Cluster.
<code>-passfile_admin <i>afile</i></code>	If you have a password file for administrators, then you can specify it using this parameter. If no password file exists, then SRVCTL randomly generates a password.
<code>-passfile_readonly <i>rfile</i></code>	If you have a password file for read-only users, then you can specify it using this parameter. If no password file exists, then SRVCTL writes a randomly-generated password to a file.

srvctl config cdp

Displays the configuration of cross-cluster dependency proxy resources.

Syntax

```
srvctl config cdp [-cdpnumber cdp_ordinal_number]
```

Usage Notes

You can optionally specify the ordinal number of a specific cross-cluster dependency proxy resource for which you want to display configuration information. If you do not specify the ordinal number option, then SRVCTL displays the configuration information for all cross-cluster dependency proxy resources in a cluster.

srvctl disable cdp

Disables a cross-cluster dependency proxy resource.

Syntax

```
srvctl disable cdp [-cdpnumber cdp_ordinal_number]
```

Usage Notes

You can optionally specify the ordinal number of a specific cross-cluster dependency proxy resource that you want to disable. If you choose not to do so, then SRVCTL disables all cross-cluster dependency proxy resources in a cluster.

srvctl enable cdp

Enables a cross-cluster dependency proxy resource.

Syntax

```
srvctl enable cdp [-cdpnumber cdp_ordinal_number]
```

Usage Notes

You can optionally specify the ordinal number of a specific cross-cluster dependency proxy resource that you want to enable. If you choose not to do so, then SRVCTL enables all cross-cluster dependency proxy resources in cluster.

srvctl modify cdp

Modifies ports and passwords for a cross-cluster dependency proxy resource.

Syntax

```
srvctl modify cdp [-port port_number] [-passfile_admin afile] [-passfile_readonly rfile]
```

Parameters

Table G-19 srvctl modify cdp Command Parameters

Parameter	Description
<code>-port <i>port_number</i></code>	Optionally, you can modify the port on which all cross-cluster dependency proxy resources in the cluster listen.
<code>-passfile_admin <i>afile</i></code>	Optionally, you can modify the password information for administrators.
<code>-passfile_readonly <i>rfile</i></code>	Optionally, you can modify the password information for read-only users.

Usage Notes

You must stop the cross-cluster dependency proxy resource before you make any changes.

srvctl relocate cdp

Relocates a cross-cluster dependency proxy resource in a cluster.

Syntax

```
srvctl relocate cdp -cdpnumber cdp_ordinal_number [-node node_name] [-force]
```

Parameters

Table G-20 `srvctl relocate cdp` Command Parameters

Parameter	Description
<code>-cdpnumber</code> <i>cdp_ordinal_number</i>	Specify a cross-cluster dependency proxy resource that you want to relocate by specifying its identifying ordinal number.
<code>-node</code> <i>node_name</i>	Optionally, you can specify the name of a node on which a cross-cluster dependency proxy resource resides that you want to relocate.
<code>-force</code>	Optionally, you can use this parameter to relocate a cross-cluster dependency proxy resource with a running VIP. If you try to relocate a cross-cluster dependency proxy resource with a running VIP without using this parameter, then the command fails.

srvctl remove cdp

Removes a cross-cluster dependency proxy resource from a cluster without removing cross-cluster dependency proxy resources.

Syntax

```
srvctl remove cdp [-force]
```

Usage Notes

- Use the `-force` parameter to remove a running cross-cluster dependency proxy resource.
- This command completely removes all cross-cluster dependency proxy configuration. If you use `-force` while proxies exist, then the proxies will no longer function until they are re-created.

srvctl start cdp

Starts a cross-cluster dependency proxy resource on a cluster.

Syntax

```
srvctl start cdp [-cdpnumber cdp_ordinal_number] [-node node_name]
```

Parameters

Table G-21 `srvctl start cdp` Command Parameters

Parameter	Description
<code>-cdpnumber</code> <i>cdp_ordinal_number</i>	Optionally, you can use this parameter to start a specific cross-cluster dependency proxy resource by specifying its identifying ordinal number.

Table G-21 (Cont.) srvctl start cdp Command Parameters

Parameter	Description
<code>-node <i>node_name</i></code>	Optionally, you can specify the name of a node on which a cross-cluster dependency proxy resource resides that you want to start.

srvctl status cdp

Obtains the status of cross-cluster dependency proxy resources.

Syntax

```
srvctl status cdp [-cdpnumber cdp_ordinal_number]
```

Usage Notes

You can optionally specify the ordinal number of a specific cross-cluster dependency proxy resource. If you choose not to do so, then SRVCTL returns the status of all cross-cluster dependency proxy resources in the cluster.

srvctl stop cdp

Stops a cross-cluster dependency proxy resource on a cluster.

Syntax

```
srvctl stop cdp [-cdpnumber cdp_ordinal_number]
```

Usage Notes

Optionally, you can stop a specific cross-cluster dependency proxy resource by specifying its identifying ordinal number.

srvctl add cdproxy

Adds a cross-cluster dependency proxy resource to a cluster (credentials must exist or be imported, first).

Syntax

```
srvctl add cdproxy -client_type client_type -client_name client_name  
[-remote_start {YES|NO}]
```

Parameters

Table G-22 `srvctl add cdproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	Specify the cluster resource type that you want to proxy.
<code>-client_name</code> <i>client_name</i>	Specify a name for the cluster resource that you want to proxy.
<code>-remote_start</code> {YES NO}	Specify whether to start the proxy resource when the remote resource starts.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

`srvctl config cdproxy`

Displays configuration information for a cross-cluster dependency proxy resource.

Syntax

```
srvctl config cdproxy -client_type client_type [-client_name client_name]
```

Parameters

Table G-23 `srvctl config cdproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	The cluster resource type of the proxy for which you want to display the configuration information.
<code>-client_name</code> <i>client_name</i>	Specify a name for the cluster resource proxy that you want to display the configuration information.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

`srvctl disable cdproxy`

Disables a cross-cluster dependency proxy resource.

Syntax

```
srvctl disable cdproxy -client_type client_type -client_name client_name
```

Parameters

Table G-24 `srvctl disable cdpproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	The cluster resource type of the proxy you want to disable.
<code>-client_name</code> <i>client_name</i>	Specify a name for the cluster resource proxy that you want to disable.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

`srvctl enable cdpproxy`

Enables a cross-cluster dependency proxy resource.

Syntax

```
srvctl enable cdpproxy -client_type client_type -client_name client_name
```

Parameters

Table G-25 `srvctl enable cdpproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	The cluster resource type of the proxy you want to enable.
<code>-client_name</code> <i>client_name</i>	Specify a name for the cluster resource proxy that you want to enable.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

`srvctl modify cdpproxy`

Modifies the configuration of a cross-cluster dependency proxy resource.

Syntax

```
srvctl modify cdpproxy -client_type client_type -client_name client_name
[-remote_start {YES|NO}]
```

Parameters

Table G-26 `srvctl modify cdproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	The cluster resource type that you want to proxy.
<code>-client_name</code> <i>client_name</i>	Specify a name for the cluster resource that you want to proxy.
<code>-remote_start</code> {YES NO}	Specify whether to start the proxy resource when the remote resource starts.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

`srvctl relocate cdproxy`

Relocates a cross-cluster dependency proxy from one node of a cluster to another.

Syntax

```
srvctl relocate cdproxy -client_type client_type -client_name client_name
[-node node_name] [-force]
```

Parameters

Table G-27 `srvctl relocate cdproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	Specify the cluster resource type of the proxy that you want to relocate.
<code>-client_name</code> <i>client_name</i>	Specify the name of the cluster resource proxy that you want to relocate.
<code>-node</code> <i>node_name</i>	Optionally, you can specify the name of a node to which you want to relocate the cross-cluster dependency proxy.
<code>-force</code>	You must use this parameter if resources dependent on the cross-cluster dependency proxy are running.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

`srvctl remove cdproxy`

Removes a cross-cluster dependency proxy resource.

Syntax

```
srvctl remove cdpproxy -client_type client_type [-client_name client_name]
[-force]
```

Parameters

Table G-28 `srvctl remove cdpproxy` Command Parameters

Parameter	Description
-client_type <i>client_type</i>	Specify the cluster resource type of the proxy that you want to remove.
-client_name <i>client_name</i>	Optionally, you can specify the name of a cluster resource proxy that you want to remove.
-force	You must use this parameter if the cross-cluster dependency proxy resource you want to remove is running.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

srvctl start cdpproxy

Starts a cross-cluster dependency proxy resource.

Syntax

```
srvctl start cdpproxy -client_type client_type -client_name client_name
[-node node_name]
```

Parameters

Table G-29 `srvctl start cdpproxy` Command Parameters

Parameter	Description
-client_type <i>client_type</i>	Specify the cluster resource type of the proxy that you want to start.
-client_name <i>client_name</i>	Specify a name for the cluster resource proxy that you want to start.
-node <i>node_name</i>	Optionally, you can specify the name of a node on which you want to start the cross-cluster dependency proxy.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

srvctl status cdproxy

Retrieves the status of a cross-cluster dependency proxy resource.

Syntax

```
srvctl status cdproxy -client_type client_type -client_name client_name]
```

Parameters

Table G-30 `srvctl status cdproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	Specify the cluster resource type of the proxy of which you want to check the status.
<code>-client_name</code> <i>client_name</i>	Specify the name of the cluster resource proxy of which you want to check the status.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

srvctl stop cdproxy

Stops a cross-cluster dependency proxy resource.

Syntax

```
srvctl stop cdproxy -client_type client_type -client_name client_name  
[-node node_name]
```

Parameters

Table G-31 `srvctl stop cdproxy` Command Parameters

Parameter	Description
<code>-client_type</code> <i>client_type</i>	Specify the cluster resource type of the proxy that you want to stop.
<code>-client_name</code> <i>client_name</i>	Specify a name for the cluster resource proxy that you want to stop.
<code>-node</code> <i>node_name</i>	Optionally, you can specify the name of a node on which you want to stop the cross-cluster dependency proxy resource.

Usage Notes

- For Oracle Clusterware 18c, the client type is always `diskgroup`.
- For Oracle Clusterware 18c, the client name is the name of a remote disk group.

srvctl add cvu

Adds the Cluster Verification Utility (CVU) to an Oracle Clusterware configuration.

**Note:**

This command is only available with Oracle Clusterware.

Syntax and Parameters

Use the `srvctl add cvu` command with the following syntax:

```
srvctl add cvu [-checkinterval time_in_minutes]
```

Usage Notes

- Optionally, you can change the interval between checks. By default, CVU runs once every 6 hours and verifies the health of the cluster. The shortest interval you can specify for CVU check is 10 minutes.
- Run this command while logged in as the Grid home user.

Example

To add CVU with a check interval of 6 hours, use the following command:

```
$ srvctl add cvu -checkinterval 360
```

srvctl config cvu

Displays the time interval, in minutes, between CVU checks.

Syntax and Parameters

Use the `srvctl config cvu` command with the following syntax:

```
srvctl config cvu
```

This command has no parameters.

Example

This command returns output similar to the following:

```
CVU is configured to run once every 360 minutes
```

srvctl disable cvu

Disables the Cluster Verification Utility (CVU) for Oracle Clusterware management (if enabled).

Syntax

```
srvctl disable cvu [-node node_name]
```

Parameters

Optionally, you can specify the name of a particular node on which you want to disable CVU. Otherwise, SRVCTL disables CVU on all nodes.

Example

An example of this command to disable CVU on a node named `crmnode1` is:

```
$ srvctl disable cvu -node crmnode1
```

srvctl enable cvu

Enables the Cluster Verification Utility (CVU) for Oracle Clusterware management (if disabled).

Syntax

```
srvctl enable cvu [-node node_name]
```

Parameters

Optionally, you can specify the name of a particular node on which you want to enable CVU. Otherwise, SRVCTL enables CVU on all nodes.

Example

An example of this command to enable CVU on a node named `crmnode1` is:

```
$ srvctl enable cvu -node crmnode1
```

srvctl modify cvu

Modifies the check interval for CVU.

Syntax

```
srvctl modify cvu [-checkinterval time_in_minutes]
```

Parameters

By default, CVU runs once every 360 minutes (six hours) and verifies the health of cluster. If you change the interval, then the shortest interval you can specify for CVU check is 10 minutes.

Example

An example of this command to modify CVU to have a check interval of 4 hours is:

```
$ srvctl modify cvu -checkinterval 240
```

srvctl relocate cvu

Temporarily relocates the CVU to another node in a cluster.

Syntax

```
srvctl relocate cvu [-node node_name]
```

Usage Notes

Optionally, you can specify a particular node to which you want to relocate CVU.

Example

An example of this command to temporarily relocate CVU from the local node to the cluster node named `crmnode2` is:

```
$ srvctl relocate cvu -node crmnode2
```

srvctl remove cvu

Removes CVU from an Oracle Clusterware configuration.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove cvu [-force]
```

Usage Notes

Use the `-force` parameter to remove a CVU resource that is running or starting.

srvctl start cvu

Starts the CVU resource on one node in a cluster. If you specify a node name, then CVU starts on that node.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start cvu [-node node_name]
```

Usage Notes

Optionally, you can specify a particular node to which you want to start CVU. Otherwise, SRVCTL starts CVU on all nodes.

Example

An example of this command to start CVU on a single node of a cluster is:

```
$ srvctl start cvu -node crmnode1
```

srvctl status cvu

Displays the current state of the CVU resource on one node in a cluster. If you specify a node name, then the command checks CVU status on that node.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status cvu [-node node_name]
```

Usage Notes

Optionally, you can specify a particular node where the CVU resource is running, if it is not the local node.

Example

An example of this command to check the status of CVU on the node in the cluster named `crmmode1` is:

```
$ srvctl status cvu -node crmmode1
```

srvctl stop cvu

Stops the Cluster Verification Utility (CVU) that is either in the running or starting state.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop cvu [-force]
```

Usage Notes

Optionally, you can specify the `-force` parameter to stop CVU if it is a dependant resource.

Example

An example of this command to stop CVU:

```
$ srvctl stop cvu
```

srvctl add exportfs

Creates an export file system configuration in Oracle Clusterware.

Syntax

```
srvctl add exportfs -name unique_name -id havip_name -path path_to_export  
[-clients export_clients] [-options export_options] [-type NFS | SMB]
```

Parameters

Table G-32 `srvctl add exportfs` Command Parameters

Parameter	Description
<code>-name <i>unique_name</i></code>	Enter a unique name for the export file system that you are creating. This parameter is required.

Table G-32 (Cont.) srvctl add exportfs Command Parameters

Parameter	Description
<code>-id <i>havip_name</i></code>	Enter the number or name that identifies an existing HAVIP, that you specified in the <code>srvctl add havip</code> command. This parameter is required.
<code>-path <i>path_to_export</i></code>	Enter the path to the Oracle Automatic Storage Management Cluster File System (Oracle ACFS) that you want to export. This parameter is required.
<code>-clients</code> <code><i>export_clients</i> -</code> <code><i>options</i></code> <code><i>export_options</i></code>	Optionally, enter a comma-delimited list of clients to which you want the file system exported. You can enter net groups and wildcards. For example: <code>-clients @netgroup1,</code> <code>*.us.domain.com,</code> <code>16.140.0.0, hostname1</code> Note: You can use the <code>-clients</code> parameter only for Linux systems. Use the <code>-options</code> parameter for Solaris and AIX systems.
<code>-type NFS SMB</code>	Specify the export type (NFS for a network file system or SMB for Oracle ACFS server message block). The default is NFS for Linux and UNIX platforms, and SMB for Windows.

Usage Notes

- You cannot use the `-clients` parameter with `-type SMB`.
- You must run this command as `root` on Linux and UNIX platforms.

Example

To export with operating system defaults:

```
# srvctl add exportfs -name cifsExport -path "/acfs1" -options "browsable
= yes,read only = no,comment = acfs1"
```

To export read-only to certain clients:

```
# srvctl add exportfs -name export1 -id havip1 -path /u01/db1 -clients
node1 -options ro
```

srvctl config exportfs

Displays the configuration information for the export file system in Oracle Clusterware.

Syntax

```
srvctl config exportfs [-name unique_name | -id havip_name]
```

Parameters

Table G-33 `srvctl config exportfs` Command Parameters

Parameter	Description
<code>-name <i>unique_name</i></code>	Enter the unique name of the NFS export for which you want to display the configuration.
<code>-id <i>havip_name</i></code>	Enter the name of a HAVIP to display the configuration of all the NFS exports attached to the specific HAVIP.

Example

If you specify the `-name` parameter, then this command returns output similar to the following, whether you specify `-name` or `-id`:

```
$ srvctl config exportfs -id havip1

export file system kep1 is configured
Exported path: /scratch/ghnew/base/mymount3
Export options:
Export Type:
Exported clients:
export file system kexp2 is configured
Exported path: /scratch/ghnew/base/mymount3
Export options:
Export Type:
Exported clients:
```

srvctl disable exportfs

Disables an export file system managed by Oracle Clusterware.

Syntax

```
srvctl disable exportfs -name filesystem_name
```

Usage Notes

- This command takes only the unique name of the file system you want to disable.
- You must run this command as `root` user on Linux and UNIX platforms.

Example

The following example disables an export file system called `export1`:

```
# srvctl disable exportfs -name export1
```

srvctl enable exportfs

Enables an export file system configuration in Oracle Clusterware.

Syntax

```
srvctl enable exportfs -name filesystem_name
```

This command takes only the name of the file system you want to enable, which is a required parameter.

Usage Notes

You must run this command as the `root` user on Linux and UNIX platforms.

Example

The following command example enables a file system named `export1`:

```
# srvctl enable exportfs -name export1
```

srvctl modify exportfs

Modifies an export file system configuration in Oracle Clusterware.

Syntax

```
srvctl modify exportfs -name unique_name [-path path_to_export]  
[-clients node_list] [-options nfs_options_string]
```

Parameters

Table G-34 `srvctl modify exportfs` Command Parameters

Parameter	Description
<code>-name <i>unique_name</i></code>	Enter the unique name of the NFS export file system to modify. This parameter is required.
<code>-path <i>path_to_export</i></code>	You can modify the path to the Oracle Automatic Storage Management Cluster File System (Oracle ACFS) where the export is located.
<code>-clients <i>node_list</i></code>	You can modify the comma-delimited list of clients (hosts) to where the file system is exported. You can enter net groups and wildcards. For example: <pre>-clients @netgroup1,*.us.domain.com,16.140.0.0,hostname1</pre> <p>Note: You can use the <code>-clients</code> parameter only for Linux systems. Use the <code>-options</code> parameter for Solaris and AIX systems.</p>
<code>-options <i>nfs_client_string</i></code>	Use this parameter to modify the export options for the export file system.

Usage Notes

- You must run this command as root user on Linux and UNIX platforms.
- You must start and stop the NFS export before any changes you make take effect.

Example

The following command example modifies the path of an export file system named `exportfs1`:

```
# srvctl modify exportfs -name exportfs1 -path /mnt/racdbl
```

srvctl remove exportfs

Removes the specified export file system configuration.

Syntax

```
srvctl remove exportfs -name exportfs_name [-force]
```

Parameters

Table G-35 `srvctl remove exportfs` Command Parameters

Parameter	Description
<code>-name <i>exportfs_name</i></code>	Enter the unique name of the export file system you want to remove.
<code>-force</code>	Use this parameter to ignore any dependencies and forcibly remove the export file system.

Usage Notes

- You must run this command as `root` user on Linux and UNIX platforms.
- If you do not use the `-force` parameter, then you must first stop the export file system you want to remove before you remove it.
- If you use the `-force` parameter, then the resulting configuration can cause errors when attempting to shut down the Oracle Clusterware stack.

Examples

To remove the stopped export file system named `export1`:

```
# srvctl remove exportfs -name export1
```

To remove a running export file system, leaving it exported:

```
# srvctl remove exportfs -name export1 -force
```

srvctl start exportfs

Starts an export file system configuration in Oracle Clusterware.

Syntax

```
srvctl start exportfs {-name unique_name | -id havip_name}
```

Parameters

Table G-36 *srvctl start exportfs* Command Parameters

Parameter	Description
-name <i>unique_name</i>	Enter the unique name of the NFS export you want to start.
-id <i>havip_name</i>	The unique ID associated with the HAVIP resource.

Example

To start an export file system:

```
$ srvctl start exportfs -name export1
```

srvctl status exportfs

Displays the status of an export file system configuration.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status exportfs [-name unique_name | -id havip_name]
```

Parameters

Table G-37 *srvctl status exportfs* Command Parameters

Parameter	Description
-name <i>unique_name</i>	Optionally, you can specify the unique name of the NFS export for which you want to display the status.
-id <i>havip_name</i>	Alternatively, you can specify the name of an HAVIP resource and display the status of all of the NFS exports associated with the specified HAVIP resource.

Usage Notes

If you do not specify any parameters, then SRVCTL displays the status for all NFS exports in the cluster.

Example

This command returns output similar to the following:

```
$ srvctl status exportfs

export file system export1 is enabled
export file system export1 is not exported
export file system export2 is enabled
export file system export2 is exported on node node1
```

srvctl stop exportfs

Stops an export file system configuration managed by Oracle Clusterware.

Syntax

```
srvctl stop exportfs {-name unique_name | -id havip_name} [-force]
```

Parameters

Table G-38 *srvctl stop exportfs* Command Parameters

Parameter	Description
-name <i>unique_name</i>	Enter the unique name of the NFS export you want to stop.
-id <i>havip_name</i>	Alternatively, you can specify the name of an HAVIP. If you specify this parameter, then SRVCTL stops all of the NFS exports associated with the HAVIP resource you specify.
-force	Stop the NFS export, ignoring errors.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms.

Example

An example of this command is:

```
# srvctl stop exportfs -name export1
```

srvctl add filesystem

Adds a device containing a file system (Oracle Automatic Storage Management Cluster File System (Oracle ACFS) or other) to the Oracle Clusterware stack for automount and high availability.

An Oracle ACFS file system resource is typically created for use with application resource dependency lists. For example, if an Oracle ACFS file system is configured for use as an Oracle Database home, then a resource created for the file system can be included in the resource dependency list of the Oracle Database application. This will cause the file system and stack to be automatically mounted because of the start action of the database application.

Syntax

```
srvctl add filesystem -device volume_device -path mountpoint_path
  [-volume volume_name] [-diskgroup disk_group_name]
  [-node node_list | -serverpool server_pool_list] [-user user_list]
  [-fstype {ACFS | EXT3 | EXT4}] [-fsoptions options] [-description
description]
  [-appid application_id] [-autostart {ALWAYS | NEVER | RESTORE}]
```

 **Note:**

Oracle supports the EXT3 and EXT4 values for the `-fstype` parameter only for Linux.

Parameters

Table G-39 srvctl add filesystem Command Parameters

Parameter	Description
<code>-device</code> <i>volume_device</i>	Specify the path to the file system volume device you want to add.
<code>-path</code> <i>mountpoint_path</i>	Specify the mount point path that is used to set up dependencies on other resources for nested mounts. This must be an absolute path.
<code>-volume</code> <i>volume_name</i>	Optionally, you can specify the name of the volume.
<code>-diskgroup</code> <i>disk_group_name</i>	Optionally, you can specify the name of the Oracle ACFS disk group to which you want to add the device.

Table G-39 (Cont.) srvctl add filesystem Command Parameters

Parameter	Description
-node <i>node_list</i> - serverpool <i>server_pool_list</i>	<p>Optionally, you can specify a comma-delimited list of nodes on which to mount the file system device. If you specify a list of nodes or a server pool, then SRVCTL creates a single-node file system resource. Without any nodes or server pools, SRVCTL creates a file system resource that runs on every cluster node.</p> <p>Alternatively, you can specify a comma-delimited list of server pools in which a file system will run. If you specify a list of server pools, then you dynamically limit the number or type of nodes on which a file system will run. This <i>does not</i> count as a cluster-wide file system for later usage that may require a cluster-wide Oracle ACFS resource, such as HANFS.</p> <p>Note: Nodes and server pools are mutually exclusive.</p>
-user <i>user_list</i>	<p>Optionally, you can specify a comma-delimited list of users authorized to mount and unmount the file system.</p> <p>If you specify no users, then only <code>root</code> can mount and unmount the file system. If the file system is for Oracle Database software, then specify the name of the Oracle home owner.</p>
-fstype {ACFS EXT3 EXT4}	<p>Optionally, you can specify the type of file system to be mounted. The default value is ACFS.</p> <p>If the type of file system to be mounted is not ACFS, then it must be a single-node file system, and you must specify <code>-node</code> or <code>-serverpool</code>.</p>
-fsoptions <i>options</i>	<p>Optionally, you can specify options used to mount the file system. The options are file system and operating system dependent.</p>
-description <i>description</i>	<p>Optionally, you can specify a description of the file system. This is stored as an attribute for later viewing by an administrator.</p>
-appid <i>application_id</i>	<p>Optionally, you can specify an application ID, which is a unique identifier used to group file systems into logical groupings. You can use this method to group file systems that may be different on different nodes but are used by an application that must have a dependency on the resource type, which is <code>ora.id.fs.type</code>.</p>
-autostart {ALWAYS NEVER RESTORE}	<p>Optionally, you can specify the file system resource autostart policy.</p> <p>ALWAYS: The file system resource always automatically starts. NEVER: The file system does not automatically start. RESTORE: The file system is restored to its last state. This is the default option.</p>

Usage Notes

- Use this command only with Oracle Clusterware.
- You must run this command as `root` user on Linux and UNIX platforms, or as an Administrator user on Windows platforms.
- This command must be run only one time from the Oracle Grid Infrastructure home.

- To manage Oracle ACFS on Oracle Database 12c installations, use the SRVCTL binary in the Oracle Grid Infrastructure home for a cluster (Grid home). If you have Oracle RAC or Oracle Database installed, then you cannot use the SRVCTL binary in the database home to manage Oracle ACFS.

Examples

To add the `dlvolume1-295` disk device in the disk group `RAC_DATA` as the Oracle ACFS volume `VOLUME1` with a mount point of `/oracle/cluster1/acfs`:

```
# srvctl add filesystem -device /dev/asm/dlvolume1-295  
-path /oracle/cluster1/acfs1
```

To add an Oracle ACFS file system on the dynamic volume device `asm-test-55`, with this file system mounted on one of the specified nodes at a time:

```
# srvctl add filesystem -fstype ACFS -device asm-test-55 -path myacfs  
-node node1,node2,node3
```

srvctl config filesystem

Displays the configuration for a specific file system resource.

Note:

Use this command only with Oracle Clusterware.

Syntax

```
srvctl config filesystem -device volume_device
```

Usage Notes

Use the `-device` parameter to specify the path to the file system resource for which you want to display the configuration.

Examples

The following example lists the configuration of all file systems:

```
$ srvctl config filesystem
```

The following example displays the configuration for a specific device:

```
$ srvctl config filesystem -device /dev/asm/dlvolume1-295
```

srvctl disable filesystem

Disables a specific Oracle Clusterware-managed file system volume.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable filesystem -device volume_device
```

Usage Notes

Specify the path to the file system volume you want to disable.

Example

An example of this command is:

```
# srvctl disable filesystem -device /dev/asm/racv011
```

srvctl enable filesystem

Enables an Oracle ACFS volume or generic file system in Oracle Clusterware.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl enable filesystem -device volume_device
```

Usage Notes

Specify the path to the volume device you want to enable.

Example

An example of this command is:

```
# srvctl enable filesystem -device /dev/asm/racv011
```


srvctl modify filesystem

Modifies the configuration of the file system resource.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```

srvctl modify filesystem
    {-device volume_device | -volume volume_name -diskgroup dg_name}
    [-user {/+|/-}user_name | "user_list"]
    [-mountowner user_name] [-mountgroup group_name] [-mountperm
octal_permission]
    [-path mountpoint_path] [-node node_list | -serverpool
server_pool_list]
    [-fsoptions options] [-description description]
    [-autostart {ALWAYS | NEVER | RESTORE}] [-force]

```

Parameters

Table G-40 srvctl modify filesystem Command Parameters

Parameter	Description
-device <i>volume_device</i>	Specify the path to the file system volume device that you want to modify.
-volume <i>volume_name</i>	Specify the name of the volume that you want to modify.
-diskgroup <i>dg_name</i>	Specify the name of the disk group that you want to modify.
-user <i>user_name</i> <i>user_list</i>	Specify the user(s) who is authorized to mount and unmount the file system. You can add (/+) or remove (/-) single a user, or replace the entire set of users with a comma-delimited list. If you do not specify this parameter, then SRVCTL defaults to the user running the command.
-mountowner <i>user_name</i>	Modify the owner of the mount point with the specified value.
-mountgroup <i>group_name</i>	Modify the group of the mount point with the specified value.
-mountperm <i>octal_permission</i>	Modify the permissions of the user, group, and others with the specified value.
-path <i>mountpoint_path</i>	Modify the mount point path that is used to set up dependencies on other resources for nested mounts. This must be an absolute path.

Table G-40 (Cont.) srvctl modify filesystem Command Parameters

Parameter	Description
-node <i>node_list</i> - serverpool <i>server_pool_list</i>	<p>Modify a comma-delimited list of nodes on which to mount the file system device. If you specify a list of nodes or a server pool, then SRVCTL creates a single-node file system resource. Without any nodes or server pools, SRVCTL creates a file system resource that runs on every cluster node.</p> <p>Alternatively, you can modify a comma-delimited list of server pools in which a file system will run. If you specify a list of server pools, then you dynamically limit the number or type of nodes on which a file system will run. This <i>does not</i> count as a cluster-wide file system for later usage that may require a cluster-wide Oracle ACFS resource, such as HANFS.</p> <p>Note: Nodes and server pools are mutually exclusive.</p>
-fsoptions <i>options</i>	Modify options used to mount the file system. The options (comma-delimited list) are file system and operating system dependent.
-description <i>description</i>	Modify the description of the file system.
-autostart {ALWAYS NEVER RESTORE}	<p>Modify the file system resource autostart policy.</p> <p>ALWAYS: The file system resource always automatically starts</p> <p>NEVER: The file system does not automatically start</p> <p>RESTORE: The file system is restored to its last state. This is the default option.</p>
-force	Force modification. This option ignores dependencies.

Usage Notes

- You must run this command as `root` user on Linux and UNIX platforms.
- When performing online changes to the file system resource, the changes take effect only after the resource is next (re)started.

Example

The following example changes the authorized user to `sysad` for the `RACVOL1` volume:

```
# srvctl modify filesystem -device /dev/asm/racvol1 -user /+sysad
```

srvctl predict filesystem

Predicts the consequences of file system failure.

Syntax

```
srvctl predict filesystem -device volume_device_name [-verbose]
```

Usage Notes

- Specify the path name of the file system volume device.

- Optionally, you can choose verbose output.

Example

An example of this command is:

```
$ srvctl predict filesystem -device /dev/asm/volume1-123
```

srvctl remove filesystem

Removes a specific file system resource from the cluster.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove filesystem -device volume_device_name [-force]
```

Usage Notes

- Specify the path to the file system resource device you want to remove.
- Optionally, you can use the `-force` parameter to ignore any resource dependencies and forcibly remove the resource from the cluster.
- You must run this command as `root` user on Linux and UNIX platforms, or as an Administrator user on Windows platforms.

Examples

An example of this command is:

```
# srvctl remove filesystem -device /dev/asm/racv011
```

srvctl start filesystem

Starts (mounts) the file system resource.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start filesystem -device volume_device [-node node_name]
```

Parameters

Table G-41 `srvctl start filesystem` Command Parameters

Parameter	Description
<code>-device</code> <code>volume_device</code>	Specify the path of the file system resource device that you want to start.
<code>-node node_name</code>	Optionally, you can specify the name of the node on which the file system resource should be started. If you do not specify this parameter, then SRVCTL starts the file system resource on all the available nodes in the cluster.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms, or as an Administrator user on Windows platforms, or as a configured file system user.



See Also:

[srvctl add filesystem](#) for more information about configuring file system users

Examples

To start a file system on all configured nodes in the cluster:

```
$ srvctl start filesystem -device /dev/asm/data_db1-68
```

To start the file system on `node1`:

```
$ srvctl start filesystem -device /dev/asm/data_db1-68 -node node1
```

`srvctl status filesystem`

Displays the status of the file system resource.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status filesystem [-device volume_device] [-verbose]
```

Usage Notes

- Optionally, you can specify the path of the file system resource for which you want to obtain the status. If you do not specify this parameter, then SRVCTL displays the status of all file systems.
- Optionally, you can use the `-verbose` parameter to display detailed output.

Examples

This command displays output similar to the following, depending on whether you specify a device name.

If you specify a device name:

```
$ srvctl status filesystem -device /dev/asm/racvol_1
```

```
ACFS file system is not mounted on node1  
ACFS file system is not mounted on node2
```

If you do not specify a device name:

```
$ srvctl status filesystem
```

```
ACFS file system is not running  
ACFS file system is running on node1,node3
```

 **Note:**

In the preceding examples, the file system is Oracle ACFS. If you are using other file systems, then they will display as EXT3 or EXT4.

srvctl stop filesystem

Stops (unmounts) the Oracle ACFS file system or generic file system.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop filesystem -device volume_device [-node node_name] [-force]
```

Parameters

Table G-42 `srvctl stop filesystem` Command Parameters

Parameter	Description
<code>-device</code> <code>volume_device</code>	Specify the path to the file system volume device that you want to stop.
<code>-node</code> <i>node_name</i>	Optionally, you can specify the name of a node. If you do not specify this parameter, then SRVCTL stops the volume resource on all active nodes in the cluster.
<code>-force</code>	Optionally, you can use <code>-force</code> to stop the file system and also stop any databases or other resources that depend on this file system.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms, or as an Administrator user on Windows platforms, or as a configured file system user.

**See Also:**

[srvctl add filesystem](#) for more information about configuring file system users

Example

An example of this command is:

```
# srvctl stop filesystem -device /dev/asm/racvol_1 -force
```

srvctl add gns

Adds the Grid Naming Service (GNS) to a cluster when you are using a DHCP public network or creating a client cluster.

Syntax

Use the `srvctl add gns` command with one of the following syntax models:

To add GNS to a cluster that is not currently running GNS:

```
srvctl add gns [-domain domain_name] -vip {vip_name | ip_address} [-skip]
[-verbose]
```

To change a cluster that is not running GNS to be a client cluster of another cluster that is running GNS:

```
srvctl add gns -clientdata file_name
```

To add a secondary GNS to a cluster:

```
srvctl add gns -vip vip_name | ip_address -clientdata file_name
```

Parameters

Table G-43 srvctl add gns Command Parameters

Parameter	Description
-domain <i>domain_name</i>	The network subdomain that is used for Forward DNS Lookup of cluster IP addresses. You can only use this parameter if you specify <code>-vip</code> , and you must use this parameter if the cluster to which you are adding GNS has a GNS zone delegation and you are using DHCP. The cluster to which you add GNS using <code>-vip</code> with the <code>-domain</code> parameters becomes a server cluster. If you do not specify <code>-domain</code> , then SRVCTL adds GNS without a domain.
-vip { <i>vip_name</i> <i>ip_address</i> }	Specify either the virtual IP (VIP) name or IP address on which GNS listens for DNS requests. You <i>cannot</i> use <code>-vip</code> with <code>-clientdata</code> . Use the <code>-domain</code> parameter with <code>-vip</code> to make the cluster to which you are adding GNS a server cluster.
-skip	Specify this parameter to skip the reachability check of the VIP address.
-clientdata <i>path_to_file</i>	Specify the path to the file you created with the <code>srvctl export gns</code> command that contains the GNS credentials. You must copy this file to a node in the cluster you are adding before running the <code>srvctl add gns</code> command. The cluster to which you add GNS using <code>-clientdata</code> becomes either a client cluster or a secondary GNS. You can only use the <code>-clientdata</code> parameter with the <code>-vip</code> parameter when you are adding a secondary GNS.
-verbose	Verbose output.

Usage Notes

- You must run this command as `root` user on Linux and UNIX platforms, or as an Administrator user on Windows platforms.
- When you are adding GNS to a cluster for the first time, use the `-vip` parameter with the `-domain` parameter.
- When you are adding GNS to a cluster using the `-clientdata` parameter, you must first export the GNS data to a file and manually copy the file to a node in the cluster you are adding. Use the `-clientdata` parameter and run this command on any node in the cluster.
- You can only specify the `-vip` and `-clientdata` parameters at the same time when you are adding a secondary GNS.

Examples

To add GNS to a cluster, making it a server cluster:

```
# srvctl add gns -vip 192.168.16.17 -domain cluster.mycompany.com
```

To add GNS to a cluster, making it a client cluster:

```
# srvctl add gns -clientdata /tmp/gnsdata
```

srvctl config gns

Displays the configuration for the grid naming service (GNS).

Syntax

```
srvctl config gns [-detail] [-subdomain] [-multicastport] [-node
node_name] [-port] [-status]
[-version] [-query name] [-list] [-clusterguid] [-clustername] [-
clustertype] [-loglevel]
[-network] [-role] [-instances]
```

Parameters

Table G-44 srvctl config gns Command Parameters

Parameter	Description
-detail	Print detailed configuration information about the GNS.
-subdomain	Display the subdomain served by GNS.
-multicastport	Display the port on which the GNS daemon is listening for multicast requests.
-node <i>node_name</i>	Display the configuration information for GNS on the specified node.
-port	Display the port that the GNS daemon uses to communicate with the DNS server.
-status	Display the status of GNS.
-version	Display the version of GNS.
-query <i>name</i>	Query GNS for the records belonging to a specific name.
-list	List all records in GNS.
-clusterguid	Display the globally unique identifier of the cluster where GNS is running.
-clustername	Display the name of the cluster where GNS is running.
-clustertype	Display the type of configuration of GNS on this cluster.
-loglevel	Print the log level of the GNS.
-network	Display network on which GNS is listening.
-role	Display the role of the GNS instance.

Table G-44 (Cont.) srvctl config gns Command Parameters

Parameter	Description
-instances	Display the instance list.

srvctl disable gns

Disables GNS for a specific node, or all available nodes in the cluster.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable gns [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify the name of a node in the cluster on which you want to disable GNS. If you do not specify this parameter, then SRVCTL disables GNS for the entire cluster.
- Optionally, you can use the `-verbose` parameter to display detailed output.

Example

An example of this command to disable GNS on the node named `crm7` is:

```
$ srvctl disable gns -node crm7
```

srvctl enable gns

Enables GNS on all nodes or a specific node.

 **Note:**

This command can be used only with Oracle Clusterware.

Syntax

```
srvctl enable gns [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify the name of a node on which you want to enable GNS. If you do not specify this parameter, then SRVCTL enables GNS on all nodes in the cluster.
- Optionally, you can use the `-verbose` parameter to print detailed output.

Example

An example of this command is:

```
$ srvctl enable gns -node node1
```

srvctl export gns

Exports Grid Naming Service (GNS) instance data to a file that you can use when you are either moving GNS from one server cluster to another or when you are creating a client cluster.

SRVCTL extracts the data from OCR. Exported data includes:

- The credentials used to authorize users, which includes the VIP address on which the server listens
- The names and DNS records kept in GNS

Syntax

```
srvctl export gns {-instance path_to_file | {-clientdata path_to_file -  
role {CLIENT|SECONDARY} [-version]}}
```

Parameters

Table G-45 srvctl export gns Command Parameters

Parameter	Description
<code>-instance</code> <i>path_to_file</i>	Use this parameter to specify the file to which GNS instance data is written, when you are moving GNS from one server cluster to another.
<code>-clientdata</code> <i>path_to_file</i> <code>-role</code> {CLIENT SECONDARY}	Use this parameter to specify the file to which GNS credentials data is written, when you are adding a client cluster to create a file that contains credentials that the client cluster accesses to perform name management operations on the server cluster. You must also use the <code>-role</code> parameter to specify a role for which the client data will be exported.
<code>-version</code>	Optionally, you can use this parameter to specify the version for which the client data will be generated.

Example

The following command exports the GNS credentials data to a file named `gns.txt`:

```
$ srvctl export gns -clientdata /temp/gnsdata/gns.txt
```

srvctl import gns

The `srvctl import gns` command imports data from a file that you create when you run the `srvctl export gns` command. Use this command when you want to locate GNS to a different server cluster.

When you import GNS data, SRVCTL stores the credentials and places the record data into OCR. If another GNS instance is running or data for another instance is encountered during the import procedure, then an error occurs.

This command also makes the cluster in which you run it the server cluster.

 **Note:**

This command is only available with Oracle Clusterware.

srvctl modify gns

Modifies the IP address, domain, or other configuration parameters used by the grid naming service (GNS).

Syntax

Use the `srvctl modify gns` command with one of the following syntax models:

```
srvctl modify gns -loglevel log_level
```

or

```
srvctl modify gns [-resolve name] [-verify name]  
  [-parameter name:value [,name:value...]]  
  [-vip {vip_name | ip} [-skip]] [-clientdata file_name]  
  [-role {PRIMARY} [-force]] [-verbose]
```

Parameters

Table G-46 `srvctl modify gns` Command Parameters

Parameter	Description
<code>-loglevel <i>log_level</i></code>	Specify the GNS diagnostic logging level (acceptable values are 1-6).
<code>-resolve <i>name</i></code>	Resolve the name you specify through GNS.

Table G-46 (Cont.) srvctl modify gns Command Parameters

Parameter	Description
<code>-verify name</code>	Check to see if the specified name is advertised through GNS.
<code>-parameter name:value[,name:value...]</code>	Set the value of one or more configuration parameters.
<code>-vip {vip_name ip} [-skip]</code>	Specify a VIP name or IP address on which GNS is to listen. Note: You can modify this attribute using Online Resource Attribute Modification. Optionally, you can use the <code>-skip</code> parameter to skip the reachability check of the VIP address.
<code>-clientdata file_name</code>	Modifies the GNS client data with client data contained in the specified file.
<code>-role {PRIMARY} [-force]</code>	Convert a secondary GNS instance to primary. Use the <code>-force</code> parameter to forcibly convert the secondary GNS instance to primary.

Examples

An example of this command is:

```
$ srvctl modify gns -vip 192.0.2.15
```

srvctl relocate gns

Relocates GNS from its current hosting node to another node within the cluster.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl relocate gns [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify the name of a node to which you want to move GNS.
- Optionally, you can use the `-verbose` parameter to display verbose output.
- On Linux and UNIX systems, you must be logged in as `root` and on Windows, you must be logged in as a user with Administrator privileges to run this command.

Example

An example of this command is:

```
# srvctl relocate gns -node node1
```

srvctl remove gns

Removes GNS from the cluster.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove gns [-force] [-verbose]
```

Usage Notes

- Optionally, you can use the `-force` parameter to forcibly remove GNS from the cluster, regardless of any errors that might occur.
- Optionally, you can use the `-verbose` parameter to display verbose output.

Example

An example of this command is:

```
$ srvctl remove gns
```

srvctl start gns

Starts GNS on a specific node.

Syntax

```
srvctl start gns [-loglevel log_level] [-node node_name] [-verbose]
```

Parameters

Table G-47 srvctl start gns Command Parameters

Parameter	Description
<code>-loglevel <i>log_level</i></code>	Optionally, you can specify the level of logging with which GNS should run. Log levels vary between 1 (minimal tracing) and 6 (traces everything and is time consuming).

Table G-47 (Cont.) `srvctl start gns` Command Parameters

Parameter	Description
<code>-node node_name</code>	Optionally, you can specify the name of a node in the cluster where you want to start GNS.
<code>-verbose</code>	Optionally, you can use this parameter to display verbose output.

Usage Notes

You can only run this command in the [server cluster](#). If you attempt to run this command in a [client cluster](#), then an error occurs.

Example

An example of this command to start the GNS on the cluster node named `crmnodel` is:

```
$ srvctl start gns -node crmnodel
```

`srvctl status gns`

Displays the current state of GNS.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status gns [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify a node on which GNS is running for which you want to display the state.
- Optionally, you can use the `-verbose` parameter to display detailed output.

`srvctl stop gns`

Stops GNS in the cluster.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop gns [-node node_name] [-force] [-verbose]
```

Usage Notes

- Optionally, you can specify the name of a node on which GNS is running that you want to stop.
- Optionally, you can use `-force` and `-verbose` to forcibly stop GNS and print detailed output, respectively.

Example

An example of this command to stop GNS on the local node is:

```
$ srvctl stop gns
```

srvctl update gns

Modifies a Grid Naming Service (GNS) instance.

Note:

This command is only available with Oracle Clusterware.

Syntax

Use the `srvctl update gns` command with one of these syntax models:

```
srvctl update gns -advertise name -address ip_address [-timetolive  
time_to_live  
]  
[-verbose]
```

```
srvctl update gns -delete name [-address address] [-verbose]
```

```
srvctl update gns -alias alias -name name [-timetolive time_to_live] [-  
verbose]
```

```
srvctl update gns -deletealias alias [-verbose]
```

```
srvctl update gns -createsrv service -target target -protocol protocol  
[-weight weight] [-priority priority] [-port port_number]  
[-timetolive time_to_live] [-instance instance_name] [-verbose]
```

```
srvctl update gns -deletesrv service_name -target target -protocol protocol  
[-verbose]
```

```
srvctl update gns -createtxt name -target target [-timetolive time_to_live]  
[-namettl name_ttl] [-verbose]
```

```
srvctl update gns -deletetxt name -target target [-verbose]
```

```
srvctl update gns -createptr name -target target [-timetolive time_to_live]  
[-namettl name_ttl] [-verbose]
```

```
srvctl update gns -deleteptr name -target target [-verbose]
```

Parameters

Table G-48 srvctl update gns Command Parameters

Parameter	Description
-advertise <i>name</i>	Use this parameter to advertise a name through GNS.
-address <i>ip_address</i>	Specify the IP address for an advertised name.
-timetolive <i>time_to_live</i>	Optionally, you can specify, in seconds, an amount of time for the record to live.
-delete <i>name</i>	Use this parameter to remove the advertisement of a name from GNS.
-alias <i>alias</i>	Use this parameter to create an alias for an advertised name.
-name <i>name</i>	You must specify a name to associate with the alias.
-deletealias <i>alias</i>	Use this parameter to delete an alias.
-createsrv <i>service</i>	Use this parameter to create a service that is described by the record.
-target <i>target</i>	Use this parameter to associate a name with the service.
-protocol <i>protocol</i>	Use this parameter to specify which protocol is used to connect to the service.
-weight <i>weight</i>	Optionally, you can specify the weight of the record.
-priority <i>priority</i>	Optionally, you can specify a priority of the record, with a value ranging from 0 to 65535.
-port <i>port_number</i>	Optionally, you can specify a port number used to connect to the service, with a value ranging from 0 to 65535.
-instance <i>instance_name</i>	Optionally, you can specify an instance name for the service.
-deletesrv <i>service_name</i>	Use this parameter to specify the name of the service record you want to delete.
-createtxt <i>name</i>	Use this parameter to add a text (TXT) record using the specified name.
-namettl <i>name_ttl</i>	Optionally, you can specify the time, in seconds, to live for the name.
-deletetxt <i>name</i>	Use this parameter to delete a text (TXT) record for a specific name.
-createptr <i>name</i>	Use this parameter to add a pointer (PTR) record for a specific name.
-deleteptr <i>name</i>	Use this parameter to delete a pointer (PTR) record for a specific name.

Usage Notes

You must be logged in as the `root` user on Linux and UNIX platforms, or as an Administrator user on Windows platforms, to run this command.

Example

The following command advertises a name with GNS:

```
# srvctl update gns -advertise myname -address 192.168.1.45
```

srvctl add havip

Adds highly available VIPs (HAVIPs) (used for highly available NFS and SMB exports) to a cluster.

Syntax

```
srvctl add havip -id havip_name -address {host_name | ip_address}  
[-netnum network_number] [-description text] [-skip] [-homenode  
node_name]
```

Parameters

Table G-49 srvctl add havip Command Parameters

Parameter	Description
-id <i>havip_name</i>	Specify a unique ID for the HAVIP resource.
-address { <i>host_name</i> <i>ip_address</i> }	Specify either a host name or an IPv4 IP address for the HAVIP you are going to create. Note: You must configure the IPv4 IP address with a non-DHCP, non-round robin DNS address.
-netnum <i>network_number</i>	Optionally, you can specify a network resource upon which the HAVIP depends. The default value for this parameter is 1.
-description <i>text</i>	Optionally, you can specify a text description of the HAVIP.
-skip	Optionally, you can specify this parameter to skip the reachability check for the IP address.
-homenode <i>node_name</i>	Optionally, you can specify a preferred node or an empty string to clear the home node.

Example

An example of this command is:

```
# srvctl add havip -id myhavip -address 192.0.2.1 -netnum 2 -description  
"My HA VIP" -skip -homenode myNode2
```

srvctl config havip

Displays configuration information for a specific highly available VIP (HAVIP) (used for highly available NFS and SMB exports) or all HAVIP resources in a cluster.

Syntax

Use the `srvctl config havip` command with the following syntax:

```
srvctl config havip [-id havip_name]
```

Parameters

Table G-50 `srvctl config havip` Command Parameters

Parameter	Description
<code>-id <i>havip_name</i></code>	Specify the unique name of the HAVIP resource you want to display. If you do not specify this parameter, then SRVCTL displays the configuration information for all HAVIPs known to Oracle Clusterware.

Example

This command returns output similar to the following:

```
$ srvctl config havip -id hrexports
```

```
VIP: /mjk-vm3-vip/10.149.237.196/10.149.236.0/255.255.252.0/eth0
Description: HR Exports
```

srvctl disable havip

Prevents a specific highly-available VIP (HAVIP) (used for highly-available NFS and SMB exports) resource from running on a number of specified nodes.

Syntax

```
srvctl disable havip -id havip_name [-node node1,node2,...,noden]
```

Parameters

Table G-51 `srvctl disable havip` Command Parameters

Parameter	Description
<code>-id <i>havip_name</i></code>	Specify the unique name of the HAVIP resource you want to disable.
<code>-node <i>node1,node2,...,node n</i></code>	Specify the name of a node or a comma-delimited list of node names on which you do not want the HAVIP resource to run. If you specify all but one node in the cluster, then the HAVIP will not fail over.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms.

Example

An example of this command is:

```
# srvctl disable havip -id myhavip -node myNode1,myNode3
```

srvctl enable havip

Enables a specific highly-available VIP (HAVIP) (used for highly-available NFS and SMB exports) to run on a number of specified nodes.

Syntax

```
srvctl enable havip -id havip_name [-node node_name]
```

Parameters**Table G-52** `srvctl enable havip` Command Parameters

Parameter	Description
<code>-id <i>havip_name</i></code>	Specify the unique name of the HAVIP resource you want to enable.
<code>-node <i>node_name</i></code>	Specify the name of a node on which you want the HAVIP resource to be able to run.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms.

Examples

An example of this command is:

```
# srvctl enable havip -id myhavip -node myNode1
```

srvctl modify havip

Modifies a highly-available VIP (HAVIP) (used for highly-available NFS and SMB exports).

Syntax

```
srvctl modify havip -id havip_name [-address {host_name | ip_address}  
[-netnum network_number] [-skip]] [-description text] [-homenode  
node_name]
```

Parameters

Table G-53 `srvctl modify havip` Command Parameters

Parameter	Description
<code>-id havip_name</code>	Specify the unique name for the HAVIP resource you want to modify.
<code>-address {host_name ip_address}</code>	Specify either a host name or an IPv4 IP or IPv6 address for the HAVIP you want to modify. Note: You must configure the IPv4 IP address with a non-DHCP, non-round robin DNS address.
<code>-netnum network_number</code>	Optionally, you can change the network resource upon which the HAVIP depends. The default value for this optional parameter is 1.
<code>-skip</code>	Specify this parameter to skip checking the reachability of the IP address.
<code>-description text</code>	Specify a text description for the HAVIP.
<code>-homenode node_name</code>	Optionally, you can specify a preferred node or an empty string to clear the home node.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms.

Example

An example of this command is:

```
# srvctl modify havip -id myhavip -address 192.168.16.17 -netnum 2
```

srvctl relocate havip

Relocates a highly-available VIP (HAVIP) (used for highly-available NFS and SMB exports) to another node in a cluster.

Syntax

```
srvctl relocate havip -id havip_name [-node node_name] [-force]
```

Parameters

Table G-54 `srvctl relocate havip` Command Parameters

Parameter	Description
<code>-id havip_name</code>	Specify the unique name of the HAVIP resource you want to relocate.
<code>-node node_name</code>	Specify the name of the node to which you want to relocate the HAVIP resource.
<code>-force</code>	Optionally, you can force the relocation of the HAVIP resource.

Example

An example of this command is:

```
$ srvctl relocate havip -id myhavip -node node3
```

srvctl remove havip

Removes a highly-available VIP (HAVIP) (used for highly-available NFS and SMB exports).

Syntax

```
srvctl remove havip -id havip_name [-force]
```

Parameters

Table G-55 srvctl remove havip Command Parameters

Parameter	Description
-id <i>havip_name</i>	Specify the unique name of the HAVIP resource you want to remove.
-force	Use this parameter to forcibly remove the resource from the cluster and ignore any dependencies.

Usage Notes

- You must first stop the HAVIP resource before you attempt to remove it.
- You must run this command as `root` user on Linux and UNIX platforms.

Example

An example of this command is:

```
# srvctl remove havip -id myhavip -force
```

srvctl start havip

Starts a specific highly-available VIP (HAVIP) (used for highly-available NFS and SMB exports) on a specific node.

Syntax

```
srvctl start havip -id havip_name [-node node_name]
```

Parameters

Table G-56 `srvctl start havip` Command Parameters

Parameter	Description
<code>-id <i>havip_name</i></code>	Specify the unique name of the HAVIP resource you want to start.
<code>-node <i>node_name</i></code>	Optionally, you can use the <code>-node</code> parameter to specify the name of the node on which the HAVIP resource starts.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms.

Example

An example of this command is:

```
# srvctl start havip -id myhavip -node myNode1
```

`srvctl status havip`

Displays the status of all highly-available VIPs (HAVIPs) (used for highly-available NFS and SMB exports) in a cluster or one particular HAVIP.

Syntax

```
srvctl status havip [-id havip_name]
```

Usage Notes

Specify the unique name of the HAVIP resource you want to display. If you do not specify this parameter, then SRVCTL displays the status of all HAVIPs known to Oracle Clusterware.

Example

This command returns output similar to the following:

```
$ srvctl status havip

HAVIP ora.hal.havip is enabled
HAVIP ora.hal.havip is not running
```

`srvctl stop havip`

Stops the highly-available VIPs (HAVIPs) (used for highly-available NFS and SMB exports) on a specific node.

Syntax

```
srvctl stop havip -id havip_name [-node node_name] [-force]
```

Parameters

Table G-57 srvctl stop havip Command Parameters

Parameter	Description
-id <i>havip_name</i>	Specify the unique name of the HAVIP resource to stop.
-node <i>node_name</i>	Optionally, you can specify the name of the node on which the HAVIP resource to stop resides.
-force	Use this parameter to forcibly stop the HAVIP.

Usage Notes

You must run this command as `root` user on Linux and UNIX platforms.

Examples

An example of this command is:

```
# srvctl stop havip -id myhavip -node myNode1 -force
```

srvctl add ioserver

Adds an Oracle ASM IO Server configuration to the cluster.

When you configure, upgrade, or convert from a standard cluster to an Oracle Flex Cluster you can create Oracle ASM IO Servers on the Hub nodes in the cluster.

Syntax

```
srvctl add ioserver [-spfile spfile] [-count number_of_ioserver_instances | ALL] [-listener listener_name]
```

Parameters

Table G-58 srvctl add ioserver Command Parameters

Parameter	Description
-spfile <i>spfile</i>	Specify the path to the server parameter file.
-count <i>number_of_ioserver_instances</i> ALL	Specify the number of Oracle ASM IO Server instances you want to add or ALL to add instances to all nodes.

Table G-58 (Cont.) srvctl add ioserver Command Parameters

Parameter	Description
<code>-listener <i>listener_name</i></code>	Specify the name of the listener.

Example

An example of this command is:

```
# srvctl add ioserver -count 2
```

srvctl config ioserver

Displays the configuration details of Oracle ASM IO Servers.

Syntax

```
srvctl config ioserver
```

Usage Notes

This command has no parameters.

Example

This command displays output similar to the following:

```
$ srvctl config ioserver  
ASM I/O Server instance count: 3  
ASM I/O Server is enabled.  
ASM I/O server is enabled on nodes:  
ASM I/O server is disabled on nodes:
```

srvctl disable ioserver

Disables an Oracle ASM IO Server.

Syntax

```
srvctl disable ioserver -node node_name
```

Usage Notes

Use the `-node` parameter to specify the name of a Hub node on which you want to disable an Oracle ASM IO Server.

srvctl enable ioserver

Enable an Oracle ASM IO Server on a specific node.

Syntax

```
srvctl enable ioserver [-node node_name]
```

Usage Notes

Use the `-node` parameter to specify the name of a Hub node on which you want to enable an Oracle ASM IO Server.

srvctl getenv ioserver

Displays the values for environment variables associated with an Oracle ASM IO Server.

Syntax

```
srvctl getenv ioserver [-envs "name_list"]
```

Usage Notes

Specify a comma-delimited list of the names of environment variables. If you do not specify this parameter, then SRVCTL displays the values of all environment variables associated with the database.

srvctl modify ioserver

Modifies an Oracle ASM IO Server configuration on a cluster.

Syntax

```
srvctl modify ioserver [-spfile spfile] [-count  
number_of_ioserver_instances | ALL]  
[-listener listener_name] [-force]
```

Parameters

Table G-59 srvctl modify ioserver Command Parameters

Parameter	Description
<code>-spfile <i>spfile</i></code>	Optionally, specify the path to the server parameter file path.
<code>-count <i>number_of_ioserver_instances</i> ALL</code>	Optionally, specify a number of instances or ALL (to modify instances on all nodes) that you want to modify. If you modify the count, then, because this results in a change in cardinality of the cluster resource, the changes take effect immediately.
<code>-listener <i>listener_name</i></code>	Optionally, specify the name of the listener.

Table G-59 (Cont.) srvctl modify ioserver Command Parameters

Parameter	Description
-force	Optionally, you can ignore any dependencies and force the modification of the Oracle ASM IO Server.

srvctl relocate ioserver

Relocates an Oracle ASM IO Server instance from its current hosting node to another node within the cluster.

Syntax

```
srvctl relocate ioserver -currentnode node_name [-targetnode node_name] [-force]
```

Parameters

Table G-60 srvctl relocate ioserver Command Parameters

Parameter	Description
-currentnode <i>node_name</i>	The name of the node on which the Oracle ASM IO Server instance currently resides
-targetnode <i>node_name</i>	The name of the node to which to relocate the IO Server instance
-force	Optionally, you can force the relocation of the Oracle ASM IO Server instance

Example

The following command relocates the Oracle ASM IO Server from a node called `crsNode2` to a node called `crsNode5`:

```
$ srvctl relocate ioserver -currentnode crsNode2 -targetnode crsNode5
```

srvctl remove ioserver

Use this command to remove an Oracle ASM IO Server instance from the cluster.

Syntax

```
srvctl remove ioserver [-force]
```

Usage Notes

Use the `-force` parameter to forcibly remove the Oracle ASM IO Server instance, ignoring any dependencies.

srvctl setenv ioserver

Administers the environment configuration for an Oracle ASM IO Server.

Syntax

```
srvctl setenv ioserver {-envs "name=val[,name=val][...]" | -env "name=val"}
```

Parameters

Table G-61 srvctl setenv ioserver Command Parameters

Parameter	Description
-envs "name=val,..."	Comma-delimited list of name-value pairs of environment variables enclosed in double quotation marks (" ").
-env "name=val"	Alternatively, use this parameter to set a single environment variable to a value that contains commas or other special characters, enclosed in double quotation marks (" ").

srvctl start ioserver

Starts Oracle ASM IO Servers in the cluster.

You can also start an Oracle ASM IO Server on a particular Hub Node by specifying the node name.

Syntax

```
srvctl start ioserver [-node node_name]
```

Usage Notes

If you choose to start an Oracle ASM IO Server on a particular Hub Node, then use the `-node` parameter and specify the name of the node. If you do not specify a particular Hub Node, then SRVCTL starts all Oracle ASM IO Servers in the cluster.

Example

The following command starts an Oracle ASM IO Server on a Hub node named `mjkhub46`:

```
$ srvctl start ioserver -node mjkhub46
```

srvctl status ioserver

Displays the status of Oracle ASM IO Servers running in the cluster.

You can display the status of all Oracle ASM IO Servers running a cluster, the status of an Oracle ASM IO Server running on a particular Hub Node, or the information about clients from a specific database.

Syntax

```
srvctl status ioserver [-node node_name] [-detail]
```

Parameters

Table G-62 `srvctl status ioserver` Parameters

Parameter	Description
<code>-node <i>node_name</i></code>	Specify a particular Hub node to display the status of the Oracle ASM IO Server resource instance running there. If you do not specify a Hub node, then SRVCTL returns the status of all Oracle ASM IO Server resource instances running in the cluster.
<code>-detail</code>	Use this parameter to display more detailed information about the status of the Oracle ASM IO Server.

Example

If you run the command with the `-detail` parameter, then SRVCTL displays output similar to the following:

```
$ srvctl status ioserver -detail
ASM I/O Server is enabled
ASM I/O Server is running on nodes: mjk376, mjk377

ASM I/O Server instance +IOS1 running on node mjk376 is connected to ASM
instance +ASM2:
Number of clients connected: 2
Names of clients: later14, alus16, later12, alus9, db0010

ASM I/O Server instance +IOS2 running on node mjk377 is connected to ASM
instance +ASM3:
Number of clients connected: 3
Number of requests completed per sec: 110
Names of networks: eth1, eth2
Names of clients: later11, later3, later14, later15, later16, alus19,
alus17, alus13
```

srvctl stop ioserver

Stops an Oracle ASM IO Server in the cluster.

Syntax

```
srvctl stop ioserver [-node node_name] [-force]
```

Usage Notes

- Use the `-node` parameter to stop an Oracle ASM IO Server on a particular Hub Node. If you do not specify a particular Hub Node, then SRVCTL stops all of the Oracle ASM IO Servers in the cluster.

- Use the `-force` parameter to stop an Oracle ASM IOserver, regardless of any dependencies.

srvctl unsetenv ioserver

Unsets the Oracle ASM IOserver environment configurations.

Syntax

```
srvctl unsetenv ioserver -envs "name_list"
```

Usage Notes

Use the `-envs` parameter to specify a comma-delimited list of the names of environment variables enclosed in double quotation marks (" ").

srvctl add mgmtdb

Adds a management database (CHM repository) resource to the cluster.

Syntax

```
srvctl add mgmtdb [-domain domain]
```

Usage Notes

Optionally, you can specify the domain for the database. If you have the `DB_DOMAIN` database initialization parameter set, then you must provide a value for this parameter.

Example

To add a management database to the cluster:

```
$ srvctl add mgmtdb -domain example.com
```

srvctl config mgmtdb

Displays configuration information for the management database (CHM repository) resource.

Note:

Use this command only with Oracle Clusterware.

Syntax

```
srvctl config mgmtdb [-verbose] [-all]
```

Usage Notes

- Optionally, use the `-verbose` parameter to print detailed output.
- Optionally, use the `-all` parameter to print detailed configuration information.

srvctl disable mgmtdb

Disables the management database (CHM repository) resource on the specified node.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable mgmtdb [-node node_name]
```

Usage Notes

Optionally, you can specify the name of a cluster node on which the management database you want to disable is running.

Example

An example of this command is:

```
$ srvctl disable mgmtdb
```

srvctl enable mgmtdb

Enables a management database (CHM repository) resource.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl enable mgmtdb [-node node_name]
```

Usage Notes

Optionally, you can specify the name of a cluster node on which you want to enable a management database resource.

Example

To enable a management database on the node named `node5`, use the following command:

```
$ srvctl enable mgmtdb -node node5
```

srvctl getenv mgmtdb

Gets the environment variables for the management database (CHM repository).

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl getenv mgmtdb [-envs "name_list"]
```

Usage Notes

Specify a comma-delimited list of the names of environment variables enclosed in double quotation marks (" "). If you do not use this parameter, then SRVCTL displays the values of all environment variables associated with the listener.

Example

The following example lists all environment variables specified for the management database:

```
$ srvctl getenv mgmtdb
```

srvctl modify mgmtdb

Modifies the configuration for the management database (CHM repository).

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl modify mgmtdb [-pfile password_file_path] [-spfile spfile_path]  
  [-startoption start_option] [-stopoption stop_option] [-diskgroup  
  "diskgroup_list"]
```

Parameters

Table G-63 `srvctl modify mgmtdb` Command Parameters

Parameter	Description
<code>-pwfile</code> <code>password_file_path</code>	Modify the full path to the location of the password file for the management database.
<code>-spfile</code> <code>spfile_path</code>	Modify the path name of the new spfile to be used by the management database.
<code>-startoption</code> <code>start_option</code>	Modify the startup options for the management database, such as OPEN, MOUNT, or NOMOUNT. Note: For multi-word startup options, such as read only and read write, separate the words with a space and enclose in double quotation marks (" "). For example, "read only". See Also: <i>SQL*Plus User's Guide and Reference</i> for more information about startup options
<code>-stopoption</code> <code>stop_option</code>	Modify the stop options for the management database, such as NORMAL, TRANSACTIONAL, IMMEDIATE, or ABORT. See Also: <i>SQL*Plus User's Guide and Reference</i> for more information about shutdown options
<code>-diskgroup</code> <code>"diskgroup_list"</code>	Modify a comma-delimited list of Oracle ASM disk groups used by the management database, enclosed in double quotation marks (" ").

Example

The following example directs the management database to use the SYSFILES Oracle ASM disk group:

```
$ srvctl modify mgmtdb -diskgroup "SYSFILES"
```

srvctl relocate mgmtdb

Relocates the management database (CHM repository) resource from one node of the cluster to another.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl relocate mgmtdb [-node node_name]
```

Usage Notes

Specify a node to which you want to relocate the management database.

Example

An example of this command is:

```
$ srvctl relocate mgmtdb -node crsNode2
```

srvctl remove mgmtdb

Removes the management database (CHM repository) from Oracle Clusterware management.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove mgmtdb [-force] [-noprompt] [-verbose]
```

Parameters

Table G-64 srvctl remove mgmtdb Command Parameters

Parameter	Description
-force	Optionally, you can use this parameter to forcibly remove the management database from the cluster regardless of any errors that might occur.
-noprompt	Optionally, you can use this parameter to suppress prompts.
-verbose	Optionally, you can use this parameter to display verbose output.

Usage Notes

You must first shut down the management database before you attempt to remove it.

Example

An example of this command is:

```
$ srvctl remove mgmtdb -noprompt
```

srvctl setenv mgmtdb

Administers the environment configuration for the management database (CHM repository).

**Note:**

This command is only available with Oracle Clusterware.

Syntax

Use this command with one of these syntax models:

```
srvctl setenv mgmtdb -envs "name=val[,...]"
```

```
srvctl setenv mgmtdb -env "name=val"
```

Parameters

Table G-65 srvctl setenv mgmtdb Command Parameters

Parameter	Description
-envs "name=val[,...]"	Specify a comma-delimited list of name-value pairs of environment variables enclosed in double quotation marks (" ").
-env "name=val"	Use this parameter to enable a single environment variable to be set to a value that contains commas or other special characters enclosed in double quotation marks (" ").

Example

The following example sets the language environment configuration for the management database:

```
$ srvctl setenv mgmtdb -env LANG=en
```

srvctl start mgmtdb

Starts the management database (CHM repository) resource.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start mgmtdb [-startoption start_options] [-node node_name]
```

Parameters

Table G-66 `srvctl start mgmtdb` Command Parameters

Parameter	Description
<code>-startoption</code> <i>start_options</i>	<p>Optionally, you can specify options to the startup command, such as <code>OPEN</code>, <code>MOUNT</code>, or <code>NOMOUNT</code>.</p> <p>Note: For multi-word startup options, such as <code>read only</code> and <code>read write</code>, separate the words with a space and enclose in double quotation marks (" "). For example, "read only".</p> <p>See Also: <i>SQL*Plus User's Guide and Reference</i> for more information about startup options</p>
<code>-node</code> <i>node_name</i>	<p>Optionally, you can specify a particular node on which to start a management database resource.</p> <p>Note: This parameter can be used only with Oracle Clusterware.</p>

Example

An example of this command to start the management database on the `crmnode1` node of the cluster is:

```
$ srvctl start mgmtdb -node crmnode1
```

srvctl status mgmtdb

Displays the current state of the management database (CHM repository) resource.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status mgmtdb [-verbose]
```

Usage Notes

Optionally, you can use the `-verbose` parameter to display detailed output.

Example

An example of this command to check the status of the management database is:

```
$ srvctl status mgmtdb
```

srvctl stop mgmtdb

Stops the management database (CHM repository) resource.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop mgmtdb [-stopoption stop_options] [-force]
```

Parameters

Table G-67 srvctl stop mgmtdb Command Parameters

Parameter	Description
<code>-stopoption</code> <code><i>stop_options</i></code>	Optionally, you can use this parameter to specify shutdown command options, such as NORMAL, TRANSACTIONAL, IMMEDIATE, or ABORT. See Also: <i>SQL*Plus User's Guide and Reference</i> for more information about shutdown options
<code>-force</code>	Use this parameter to forcibly stop the database and any associated services and any dependent resources.

Example

An example of this command is:

```
$ srvctl stop mgmtdb -stopoption NORMAL
```

srvctl unsetenv mgmtdb

Unsets the management database (CHM repository) environment configurations.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl unsetenv mgmtldb -envs "name_list"
```

Usage Notes

Use the `-envs` parameter to specify a comma-delimited list of the names of environment variables enclosed in double quotation marks ("").

Example

The following example unsets the environment configuration for a management database environment variable:

```
$ srvctl unsetenv mgmtldb -envs "LANG"
```

srvctl add mgmtlsnr

Adds a management listener resource (for Cluster Health Monitor) to the cluster.

Syntax

```
srvctl add mgmtlsnr [-endpoints "[TCP:]port_list[/IPC:key][/NMP:pipe_name  
[/TCPS:s_port][/SDP:port][/EXADIRECT:port]" ] [-skip]]
```

Parameters

Table G-68 srvctl add mgmtlsnr Command Parameters

Parameter	Description
-endpoints "[TCP:]port_list [/IPC:key][/ NMP:pipe_name [/TCPS:s_port][/ SDP:port][/ EXADIRECT:port]"]	Protocol specifications for the listener. <i>port_list</i> is a comma-delimited list of TCP ports or listener endpoints
-skip	Indicates you want to skip the checking of ports.

Example

The following command adds a management listener that is listening on port 1341 to the cluster:

```
$ srvctl add mgmtlsnr -endpoints "TCP:1341"
```

srvctl config mgmtlsnr

Displays configuration information for the management listener resource (for CHM).

**Note:**

Use this command only with Oracle Clusterware.

Syntax

```
srvctl config mgmtlsnr [-all]
```

Usage Notes

Use the `-all` parameter to print detailed configuration information.

srvctl disable mgmtlsnr

Disables the management listener resource (for CHM) on the specified node.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable mgmtlsnr [-node node_name]
```

Usage Notes

Optionally, you can specify the name of a cluster node on which the management listener you want to disable is running.

Example

An example of this command is:

```
$ srvctl disable mgmtlsnr
```

srvctl enable mgmtlsnr

Enables a management database listener resource (for CHM).

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl enable mgmtlsnr [-node node_name]
```

Usage Notes

Optionally, you can specify the name of a cluster node on which you want to enable a management listener.

Example

To enable a management listener on a node named `node5`, use the following command:

```
$ srvctl enable mgmtlsnr -node node5
```

srvctl getenv mgmtlsnr

Gets the environment variables for the management listener resource (for CHM).

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl getenv mgmtlsnr [-envs "name_list"]
```

Usage Notes

Optionally, you can specify a comma-delimited list of environment variable names. If you do not use this parameter, then SRVCTL displays the values of all environment variables associated with the listener.

Example

The following example lists all environment variables specified for the management listener:

```
$ srvctl getenv mgmtlsnr
```

srvctl modify mgmtlsnr

Modifies the configuration for the management listener resource (for CHM).



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl modify mgmtlsnr [-endpoints "[TCP:]port_list[/IPC:key][/  
NMP:pipe_name]  
[/TCPS:s_port][/SDP:port"]]
```

Usage Notes

Optionally, you can specify protocol specifications for the management listener. *port_list* is comma-delimited list of port numbers. You can also modify this attribute using Online Resource Attribute Modification.

Example

The following example changes the TCP ports for the management listener:

```
$ srvctl modify mgmtlsnr -endpoints "TCP:2521,2522"
```

srvctl remove mgmtlsnr

Removes the management listener resource (for CHM) from Oracle Clusterware.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove mgmtlsnr [-force]
```


Usage Notes

Use the `-force` parameter to forcibly remove the management listener, ignoring any dependencies.

Example

An example of this command is:

```
$ srvctl remove mgmtlsnr
```

srvctl setenv mgmtlsnr

Administers the environment configuration for the management listener resource (for CHM).

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

Use this command with one of the following syntax models:

```
srvctl setenv mgmtlsnr -envs "name=val[,...]"
```

```
srvctl setenv mgmtlsnr -env "name=val"
```

Parameters

Table G-69 srvctl setenv mgmtlsnr Command Parameters

Parameter	Description
-envs "name=val[,...]"	Specify a comma-delimited list of name-value pairs of environment variables enclosed in double quotation marks (") that you want to set.
-env "name=val"	Use this parameter to enable a single environment variable to be set to a value that contains commas or other special characters enclosed in double quotation marks (").

Example

The following example sets the language environment configuration for the management listener:

```
$ srvctl setenv mgmtlsnr -env LANG=en
```

srvctl start mgmtlsnr

Starts the management listener resource (for CHM).

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start mgmtlsnr [-node node_name]
```

Usage Notes

Specify the node on which you want to start the management listener.

Example

To start the management listener on the `crmnode1` node:

```
$ srvctl start mgmtlsnr -node crmnode1
```

srvctl status mgmtlsnr

Displays the status of the management listener resource (for CHM).

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status mgmtlsnr [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify the name of a cluster node.
- Optionally, use the `-verbose` parameter to display detailed output.

Examples

To display the status of the management listener on the node `node2`, use the following command:

```
$ srvctl status mgmtlsnr -node node2
```

srvctl stop mgmtlsnr

Stops the management listener resource (for CHM) on all nodes or the specified node.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop mgmtlsnr [-node node_name] [-force]
```

Usage Notes

- Optionally, you can specify the name of a single node on which the management listener runs.
- Optionally, you can use the `-force` parameter to forcibly stop the management listener.

Example

The following command stops the management listener on the node `mynode1`:

```
$ srvctl stop mgmtlsnr -node mynode1
```

srvctl unsetenv mgmtlsnr

Unsets the management listener resource (for CHM) environment configurations.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl unsetenv mgmtlsnr -envs "name_list"
```

Usage Notes

Use the `-envs` parameter to specify a comma-delimited list of the names of environment variables enclosed in double quotation marks ("").

Example

The following example unsets the environment configuration for a management listener environment variable:

```
$ srvctl unsetenv mgmtlsnr -envs "LANG"
```

srvctl add mountfs

Adds Network Attached Storage (NAS) configuration to Oracle Clusterware.



Note:

Use this command only with Oracle Clusterware.

Syntax

```
srvctl add mountfs -name mountfs_name -path mount_path -exportserver  
server_name  
-exportpath path [-mountoptions mount_options] [-user user]
```

Parameters

Table G-70 srvctl add mountfs Command Parameters

Parameter	Description
-name <i>mountfs_name</i>	Specify a unique name for the NAS.
-path <i>mount_path</i>	Specify a mount path for the NAS.
-exportserver <i>server_name</i>	Specify a name for the export server.
-exportpath <i>path</i>	Specify an export file path.
-mountoptions <i>mount_options</i>	Optionally, you can specify mount options for the NAS.
-user <i>user</i>	Optionally, you can specify a user authorized to mount and dismount the NAS.

srvctl config mountfs

Displays the configuration information for a specific NAS.



Note:

Use this command only with Oracle Clusterware.

Syntax

```
srvctl config mountfs [-name mountfs_name]
```

Usage Notes

Specify the unique name of the NAS for which you want to display configuration information.

srvctl disable mountfs

Disables the specified NAS from Oracle Clusterware management.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable mountfs -name mountfs_name [-node node_list]
```

Parameters**Table G-71** *srvctl* disable mountfs Command Parameters

Parameter	Description
-name <i>mountfs_name</i>	Specify the unique name for the NAS you want to disable.
-node <i>node_list</i>	Optionally, you can specify a comma-delimited list of nodes on which to disable NAS.

srvctl enable mountfs

Enables the specified Network Attached Storage (NAS) for Oracle Clusterware management.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl enable mountfs -name mountfs_name [-node node_list]
```

Parameters

Table G-72 `srvctl enable mountfs` Command Parameters

Parameter	Description
<code>-name mountfs_name</code>	Specify the unique name for the NAS you want to enable.
<code>-node node_list</code>	Optionally, you can specify a comma-delimited list of nodes on which to enable NAS.

srvctl modify mountfs

Modifies Network Attached Storage (NAS) configuration in Oracle Clusterware.



Note:

This command is available only with Oracle Clusterware.

Syntax

```
srvctl modify mountfs -name mountfs_name [-path mount_path] [-exportserver
server_name]
[-exportpath path] [-mountoptions mount_options] [-user user_name]
```

Parameters

Table G-73 `srvctl modify mountfs` Command Parameters

Parameter	Description
<code>-name mountfs_name</code>	Specify a unique name for the NAS you want to modify.
<code>-path mount_path</code>	Modify the mount path for the NAS.
<code>-exportserver server_name</code>	Modify the name of the export server.
<code>-exportpath path</code>	Modify the export file path.
<code>-mountoptions mount_options</code>	Modify mount options for the NAS.
<code>-user user_name</code>	Modify the user authorized to mount and dismount the NAS.

srvctl remove mountfs

Removes a specific NAS configuration.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove mountfs -name mountfs_name [-force]
```

Usage Notes

- Specify the unique name for the NAS to mount.
- Optionally, you can use the `-force` parameter to ignore resource dependencies while removing the NAS.

srvctl start mountfs

Mounts a specific NAS.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start mountfs -name mountfs_name [-node node_list]
```

Parameters

Table G-74 srvctl start mountfs Command Parameters

Parameter	Description
<code>-name <i>mountfs_name</i></code>	Specify the unique name for the NAS to mount.
<code>-node <i>node_list</i></code>	Optionally, you can specify a comma-delimited list of nodes on which to mount NAS.

srvctl status mountfs

Displays the current state of a specific NAS.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status mountfs -name mountfs_name
```

Usage Notes

Specify the unique name of the NAS for which you want to display the status.

srvctl stop mountfs

Unmounts a specific NAS.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop mountfs -name mountfs_name [-node node_list] [-force]
```

Parameters

Table G-75 srvctl stop mountfs Command Parameters

Parameter	Description
-name <i>mountfs_name</i>	Specify the unique name for the NAS to unmount.
-node <i>node_list</i>	Optionally, you can specify a comma-delimited list of nodes from which to unmount NAS.
-force	Optionally, you can use the -force parameter to ignore resource dependencies while stopping NAS.

srvctl add netstorageservice

Adds the netstorageservice resource.

Syntax

```
srvctl add netstorageservice -device volume_device
```

Parameters

Table G-76 `srvctl add netstorageservice` Command Parameters

Parameter	Description
<code>-device <i>volume_device</i></code>	Specify the device and volume name to which you want to add the <code>netstorageservice</code> resource.

Examples

To add the `netstorageservice` resource:

```
$ srvctl add netstorageservice -device /dev/asm/data_db1-68
```

srvctl config netstorageservice

Displays the configuration details for the `netstorageservice` resource.

Syntax

```
srvctl config netstorageservice
```

Examples

To list the `netstorageservice` resource configuration details:

```
$ srvctl config netstorageservice
```

srvctl disable netstorageservice

Disables the `netstorageservice` resource.

Syntax

```
srvctl disable netstorageservice [-node node_name]
```

Parameters

Table G-77 `srvctl disable netstorageservice` Command Parameters

Parameter	Description
<code>-node <i>node_name</i></code>	Optionally, you can specify the name of the node on which the <code>netstorageservice</code> resource should be disabled. If you do not specify this parameter, then SRVCTL disables the <code>netstorageservice</code> resource on all the available nodes in the cluster.

Examples

To disable the `netstorageservice` resource on a specified node:

```
$ srvctl disable netstorageservice -node nodel
```

`srvctl enable netstorageservice`

Enables the `netstorageservice` resource.

Syntax

```
srvctl enable netstorageservice [-node node_name]
```

Parameters

Table G-78 `srvctl enable netstorageservice` Command Parameters

Parameter	Description
<code>-node <i>node_name</i></code>	Optionally, you can specify the name of the node on which the <code>netstorageservice</code> resource should be enabled. If you do not specify this parameter, then SRVCTL enables the <code>netstorageservice</code> resource on all the available nodes in the cluster.

Examples

To enable the `netstorageservice` resource on a specified node:

```
$ srvctl enable netstorageservice -node nodel
```

`srvctl remove netstorageservice`

Removes the `netstorageservice` resource.

Syntax

```
srvctl remove netstorageservice [-force]
```

Parameters**Table G-79** `srvctl remove netstorageservice` Command Parameters

Parameter	Description
-force	Force a stop.

Examples

To remove the `netstorageservice` resource:

```
$ srvctl remove netstorageservice
```

srvctl start netstorageservice

Starts the `netstorageservice` resource.

Syntax

```
srvctl start netstorageservice [-node node_name]
```

Parameters**Table G-80** `srvctl start netstorageservice` Command Parameters

Parameter	Description
-node <i>node_name</i>	Optionally, you can specify the name of the node on which the <code>netstorageservice</code> resource should be started. If you do not specify this parameter, then SRVCTL starts the <code>netstorageservice</code> resource on all the available nodes in the cluster.

Examples

To start the `netstorageservice` resource on a specified node:

```
$ srvctl start netstorageservice -node node1
```

srvctl status netstorageservice

Syntax

```
srvctl status netstorageservice
```

Examples

To display the status of the `netstorageservice` resource:

```
$ srvctl status netstorageservice
```

srvctl stop netstorageservice

Stops the `netstorageservice` resource.

Syntax

```
srvctl stop netstorageservice [-node node_name] [-force]
```

Parameters

Table G-81 `srvctl stop netstorageservice` Command Parameters

Parameter	Description
<code>-node <i>node_name</i></code>	Optionally, you can specify the name of the node on which the <code>netstorageservice</code> resource should be stopped. If you do not specify this parameter, then SRVCTL stops the <code>netstorageservice</code> resource on all the available nodes in the cluster.
<code>-force</code>	Force a stop.

Examples

To stop the `netstorageservice` resource on a specified node:

```
$ srvctl stop netstorageservice -node node1
```

srvctl add ovmm

Adds manager credentials to the credentials store for managing a virtual machine.

Syntax

```
srvctl add ovmm -wallet certificate_wallet_file -username user_name -  
ovmmhost host_name | ip_address  
-ovmmport port_number [-autologin]
```

Parameters

Table G-82 `srvctl add ovmm` Command Parameters

Parameter	Description
<code>-wallet</code> <code>certificate_wallet_file</code>	Specify the path to the certification wallet that contains the CA certificate used by the virtual machine manager.
<code>-autologin</code>	Optionally, you can include this parameter to indicate automatic login for the wallet credentials. If you choose not to specify this parameter, then the user will always be prompted to provide the password of the wallet.
<code>-username</code> <i>user_name</i>	Specify a user name for whom you want to add manager credentials. You must provide a manager password when prompted.
<code>-ovmmhost</code> <i>host_name</i> <i>ip_address</i>	Specify the host name or the IP address of the host designated as the virtual machine manager.
<code>-ovmmport</code> <i>port_number</i>	Specify the port number used by the virtual machine manager.

Usage Notes

You must run this command as `root` or equivalent user.

Example

An example of this command is:

```
# srvctl add ovmm -username scott -wallet /tmp/ovmmcred/cwallet.sso -
ovmmhost node1 -ovmmport 80
```

`srvctl config ovmm`

Lists the current configuration of the virtual machine manager.

Syntax

```
srvctl config ovmm
```

Usage Notes

This command takes no parameters.

Example

This command returns output similar to the following:

```
$ srvctl config ovmm

OVMM Host: host1
OVMM Port: 80
```

srvctl modify ovmm

Modifies the virtual machine manager configuration.

Syntax

```
srvctl modify ovmm [-wallet certificate_wallet_file [-autologin]] [-  
username user_name]  
    [-ovmmhost host_name | ip_address [-ovmmport port_number]]
```

Parameters

Table G-83 srvctl modify ovmm Command Parameters

Parameter	Description
-wallet <i>certificate_wallet_file</i>	Specify the path to the certification wallet that contains the CA certificate used by the virtual machine manager.
-autologin	Optionally, you can include this parameter to indicate automatic login for the wallet credentials. If you choose not to specify this parameter, then the user will always be prompted to provide the password of the wallet.
-username <i>user_name</i>	Specify a user name for whom you want to add manager credentials. You must provide a manager password when prompted.
-ovmmhost <i>host_ip</i>	Specify the host name or the IP address of the host designated as the virtual machine manager.
-ovmmport <i>port_number</i>	Specify the port number used by the virtual machine manager.

Usage Notes

- You must run this command as `root` or equivalent user.
- The `srvctl modify ovmm` command deletes all credentials before writing new ones.

Example

An example of this command is:

```
$ srvctl modify ovmm -username scott -ovmmport 81
```

srvctl remove ovmm

Removes the current configuration of the virtual machine manager.

Syntax

```
srvctl remove ovmm
```

Usage Notes

- You must run this command as `root` or equivalent user.
- This command takes no parameters.

Example

An example of this command is:

```
# srvctl remove ovmm
```

srvctl add qosmserver

Adds an Oracle Database QoS Management Server instance to all the nodes in the cluster.

Syntax

```
srvctl add qosmserver [-verbose]
```

Usage Notes

Optionally, you can use the `-verbose` parameter to display verbose output.

Example

The following command adds an Oracle Database QoS Management Server instance:

```
$ srvctl add qosmserver
```

srvctl config qosmserver

Displays configuration information for the `qosmserver` resource.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl config qosmserver
```

Parameters

This command has no parameters.

Example

The following command displays the configuration information for the `qosmsserver` resource.

```
$ srvctl config qosmsserver
```

srvctl disable qosmsserver

Disables the Oracle Database QoS Management Server instance on all nodes or on a specific node.

Syntax

```
srvctl disable qosmsserver [-node node_name] [-verbose]
```

Usage Notes

With this command you can optionally specify a particular node on which to disable an Oracle Database QoS Management Server instance, and you can also choose verbose output.

Example

The following example disables the Oracle Database QoS Management Server on a node called `crsNode3`:

```
$ srvctl disable qosmsserver -node crsNode3
```

srvctl enable qosmsserver

Enables an Oracle Database QoS Management Server instance on all nodes or on a specific node.

Syntax

```
srvctl enable qosmsserver [-node node_name] [-verbose]
```

Usage Notes

With this command you can optionally specify a particular node on which to enable an Oracle Database QoS Management Server instance, and you can also choose verbose output.

Example

The following command example enables an Oracle Database QoS Management Server instance on a node named `crsNode3`:

```
$ srvctl enable qosmsserver -node crsNode3
```


srvctl modify qosmserver

Modifies the RMI port for the Oracle Database QoS Management Server instances.

Syntax

```
srvctl modify qosmserver -rmiport port_number [-verbose] [-force]
```

Table G-84 *srvctl modify qosmserver* Command Parameters

Parameter	Description
<code>-rmiport <i>port_number</i></code>	Specify the RMI port number that the Oracle Database QoS Management Server instance uses. Note: You can modify this attribute using Online Resource Attribute Modification.
<code>-verbose</code>	Verbose output
<code>-force</code>	Stops and restarts the resource to effect a change

Example

The following command example sets the RMI port number to 5385:

```
$ srvctl modify qosmserver -rmiport 5385
```

srvctl predict qosmserver

The `srvctl predict qosmserver` command predicts the consequences of Oracle Database QoS Management Server failure.

Syntax

```
srvctl predict qosmserver [-verbose]
```

This command has only one optional parameter to display verbose output.

Example

This command returns output similar to the following:

```
$ srvctl predict qosmserver
```

```
Resource ora.qosmserver will be started on node crsNode1
```

srvctl relocate qosmserver

Relocates an Oracle Database QoS Management Server instance from its current hosting node to another node within the cluster.

Syntax

```
srvctl relocate qosmsserver [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify a particular node to which you want to relocate an Oracle Database QoS Management Server instance.
- Optionally, you can use the `-verbose` parameter to display verbose output.

Example

The following command example relocates the Oracle Database QoS Management Server instance to node `crsNode1` and displays verbose output:

```
$ srvctl relocate qosmsserver -node crsNode1 -verbose
```

srvctl remove qosmsserver

Removes the Oracle Database QoS Management Server instance from the Oracle Clusterware configuration.

Syntax

```
srvctl remove qosmsserver [-force] [-verbose]
```

Usage Notes

Optionally, you can use the `-force` and `-verbose` parameters that force the removal of the Oracle Database QoS Management Server instance and display verbose output, respectively.

Example

The following command forcibly removes the Oracle Database QoS Management Server instance:

```
$ srvctl remove qosmsserver -force
```

srvctl start qosmsserver

Starts an Oracle Database QoS Management Server instance.

Syntax

```
srvctl start qosmsserver [-node node_name] [-verbose]
```

Usage Notes

- Optionally, you can specify a particular node on which to start an Oracle Database QoS Management Server instance.

- Optionally, you can use the `-verbose` parameter to display verbose output.

Example

The following command example starts an Oracle Database QoS Management Server instance on a node called `crsNode1`:

```
$ srvctl start qosmsserver -node crsNode1
```

srvctl status qosmsserver

Determines which node is running the Oracle Database QoS Management Server instance.

Syntax

```
srvctl status qosmsserver [-node node_name] [-verbose]
```

You can optionally specify a particular about which you can obtain the status of the Oracle Database QoS Management Server instance, and you can also use the `-verbose` parameter to display verbose output.

Example

This command returns output similar to the following:

```
$ srvctl status qosmsserver

QoS Management Server is enabled.
QoS Management Server is running on node crsNode1.
```

srvctl stop qosmsserver

Stops an Oracle Database QoS Management Server instance.

Syntax

```
srvctl stop qosmsserver [-force] [-verbose]
```

Usage Notes

- Optionally, you can specify a particular node on which to stop an Oracle Database QoS Management Server instance.
- Optionally, you can use the `-verbose` parameter to display verbose output.

Example

The following command example forcibly stops an Oracle Database QoS Management Server instance:

```
$ srvctl stop qosmsserver -force
```

srvctl add rhpclient

Adds a Fleet Patching and Provisioning Client configuration file to the Oracle Clusterware configuration.

Syntax

```
srvctl add rhpclient -clientdata path_to_file [-diskgroup disk_group_name -
storage base_path]
  [-email email_address -mailserver mail_server_address -mailserverport
mail_server_port] [-verbose]
```

Parameters

Table G-85 srvctl add rhpclient Command Parameters

Parameter	Description
-clientdata <i>path_to_file</i>	Specify the path to the file that contains the Fleet Patching and Provisioning Client data.
-diskgroup <i>disk_group_name</i>	Specify the Oracle ASM disk group from which to create the Oracle ACFS file system for image storage.
-storage <i>base_path</i>	Specify a location that is available on all cluster nodes and that is used for mounting an Oracle ACFS that you create for image storage on the Fleet Patching and Provisioning Client.
-email <i>email_address</i>	Specify an email address.
-mailserver <i>mail_server_address</i>	Specify the mail server address.
-mailserverport <i>mail_server_port</i>	Specify the mail server port number.
-verbose	Displays verbose output.

Usage Notes

- You cannot run this command on a Fleet Patching and Provisioning Server.
- You must run this command as `root` user on Linux and UNIX platforms.
- This command does not create an Oracle Clusterware resource.
- To store working copies in `RHP_MANAGED_STORAGE` on the client, you must specify the `-diskgroup` and `-storage` parameters with this command (or with [page G-114](#)). An additional benefit of using these options applies to multi-node client clusters. When the Fleet Patching and Provisioning Server provisions a working copy to the client cluster, the image is transferred to one node, and the client distributes it, internally, to remaining nodes.

Example

The following command creates a Fleet Patching and Provisioning Client:

```
# srvctl add rhpclient -clientdata /tmp/gnsdata
```

srvctl config rhpclient

Displays configuration information for a Fleet Patching and Provisioning Client instance.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl config rhpclient
```

Example

This command returns output similar to the following:

```
$ srvctl config rhpclient
```

```
Fleet Patching and Provisioning Client is configured  
Cluster name: mjk9394  
Storage base path: /scratch/aime/storage  
Disk group: MJKRHPCDG  
Fleet Patching and Provisioning Server (RHPS): mjk8990clust  
Fleet Patching and Provisioning Server discovery string: 198.51.100.191  
Port number: 8896  
Fleet Patching and Provisioning Client is enabled  
Fleet Patching and Provisioning Client is individually enabled on nodes:  
Fleet Patching and Provisioning Client is individually disabled on nodes:
```

srvctl disable rhpclient

Disables a Fleet Patching and Provisioning Client instance on a specific node from Oracle Clusterware management.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable rhpclient [-node node_name]
```

Usage Notes

Optionally, you can specify a particular node on which to disable a Fleet Patching and Provisioning Client.

Example

An example of this command is:

```
$ srvctl disable rhpclient -node clusterNode01
```

srvctl enable rhpclient

Enables a Fleet Patching and Provisioning Client instance on a specific node.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl enable rhpclient [-node node_name]
```

Usage Notes

Optionally, you can specify the node on which you want to enable the Fleet Patching and Provisioning Client instance. If you do not specify a node, then SRVCTL enables all of the Fleet Patching and Provisioning Clients in the Oracle Clusterware configuration.

Example

An example of this command is:

```
$ srvctl enable rhpclient -node clusterNode03
```

srvctl modify rhpclient

Changes the locations of the client data and image storage for the Fleet Patching and Provisioning Client.

Syntax

```
srvctl modify rhpclient [-clientdata path_to_client_data] [-port rmi_port]  
[-diskgroup dg_name -storage base_path] [-email email_address -mailserver  
mail_server_address  
-mailserverport mail_server_port]
```

Parameters

Table G-86 `srvctl modify rhpclient` Command Parameters

Parameter	Description
<code>-clientdata</code> <i>path_to_client_data</i>	Modify the path to the Fleet Patching and Provisioning Client data.
<code>-port</code> <i>rmi_port</i>	Modify the RMI port number used by the Fleet Patching and Provisioning Client.
<code>-diskgroup</code> <i>dg_name</i>	Modify the Oracle ASM disk group from which to create the Oracle ACFS file systems for image storage.
<code>-storage</code> <i>base_path</i>	Modify the location which is available on every cluster node but is not necessarily shared. It is not required that this location exist when you add a Fleet Patching and Provisioning Client. All images are mounted on <i>base_path/images</i> for all local Oracle ACFS storage.
<code>-email</code> <i>email_address</i>	Modify an email address.
<code>-mailserver</code> <i>mail_server_address</i>	Modify the mail server address.
<code>-mailserverport</code> <i>mail_server_port</i>	Modify the mail server port number.

Usage Notes

- You must run this command as the `root` user on Linux and UNIX platforms.
- You run this command only on Fleet Patching and Provisioning Clients.
- After running this command, you must run `srvctl stop rhpclient` and `srvctl start rhpclient` on the Fleet Patching and Provisioning Client, so that it communicates the changes to the Fleet Patching and Provisioning Server.

`srvctl relocate rhpclient`

Relocates a Fleet Patching and Provisioning Client instance to a different node.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl relocate rhpclient -node node_name
```

Usage Notes

Specify a node to which you want to relocate the Fleet Patching and Provisioning Client instance.

Example

An example of this command is:

```
$ srvctl relocate rhpclient -node crsNode03
```

srvctl remove rhpclient

Removes a Fleet Patching and Provisioning Client instance.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl remove rhpclient [-force]
```

Usage Notes

Optionally, you can use the `-force` parameter to forcibly remove the Fleet Patching and Provisioning Client instance, ignoring any dependencies.

srvctl start rhpclient

Start a Fleet Patching and Provisioning Client instance on one or more nodes.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start rhpclient [-node node_name]
```

Usage Notes

Specify the node on which you want to start the Fleet Patching and Provisioning Client instance. If you do not specify a node, then Oracle Clusterware determines the node where the Fleet Patching and Provisioning Client is to be started. The Fleet Patching and Provisioning Client resource has cardinality of 1 and Oracle Clusterware decides the placement when the Fleet Patching and Provisioning Client resource is started.

Example

An example of this command to start the Fleet Patching and Provisioning Server on the node named `crs3`:

```
$ srvctl start rhpclient -node crs3
```

srvctl status rhpclient

Displays the current state of a Fleet Patching and Provisioning Client instance.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status rhpclient
```

Usage Notes

This command does not accept any parameters except for `-help`.

Example

This command displays output similar to the following:

```
$ srvctl status rhpclient
```

```
Fleet Patching and Provisioning Client is enabled  
Fleet Patching and Provisioning Client is running on node mjk1270093
```

srvctl stop rhpclient

Stops a Fleet Patching and Provisioning Client instance that is in a running or starting state.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop rhpclient
```

Usage Notes

This command does not accept any parameter, except for `-help`.

srvctl add rhpserver

Adds a Fleet Patching and Provisioning Server to the Oracle Clusterware configuration.

Syntax

```
srvctl add rhpserver -storage base_path -diskgroup disk_group_name
  [-email email_address -mailserver mail_server_address -mailserverport
mail_server_port]
  [-pl_port RHP_progress_listener_port] [-clport RHP_copy_listener_port]
  [-enableTLS {YES|NO}] [-enableHTTPS '{YES|NO}'] [-port_range low_val-
high_val]
```

Parameters

Table G-87 srvctl add rhpserver Command Parameters

Parameter	Description
<code>-storage <i>base_path</i></code>	Specify a location that is available on all cluster nodes. It is not necessary that this location be shared. The catalog for the server is kept in this location and all the image file systems are mounted on <i>base_path/images</i> .
<code>-diskgroup <i>disk_group_name</i></code>	Specify a particular Oracle ASM disk group from which to create the Oracle ACFS file system for storing images.
<code>-email <i>email_address</i></code>	Specify an email address.
<code>-mailserver <i>mail_server_address</i></code>	Specify the mail server address.
<code>-mailserverport <i>mail_server_port</i></code>	Specify the mail server port number.
<code>-pl_port <i>RHP_progress_listener_port</i></code>	Specify the Fleet Patching and Provisioning progress listener port number.
<code>-clport <i>RHP_copy_listener_port</i></code>	Specify the Fleet Patching and Provisioning copy listener port number.
<code>-enableTLS {YES NO}</code>	Enable Fleet Patching and Provisioning transport level security
<code>-enableHTTPS '{YES NO}'</code>	Enable Fleet Patching and Provisioning transport layer security for HTTP.
<code>-port_range <i>low_val-high_val</i></code>	Specify the lower and upper port values for the ports to use for file transfer.

Usage Notes

- You must be `root` user on Linux and UNIX platforms to run this command.

- The default user is the user that installed Oracle Clusterware, and this user gets added to the user list for the Fleet Patching and Provisioning Server resource.

srvctl config rhpserver

Displays configuration information for a Fleet Patching and Provisioning Server.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl config rhpserver
```

Example

This command returns output similar to the following:

```
$ srvctl config rhpserver
```

```
Storage base path: /scratch/aime/storagedg  
Disk Groups: MJKRHPSDG  
Port number: 8896  
Fleet Patching and Provisioning Server is enabled  
Fleet Patching and Provisioning Server is individually enabled on nodes:  
Fleet Patching and Provisioning Server is individually disabled on nodes:
```

srvctl disable rhpserver

Disables a Fleet Patching and Provisioning Server on a specific node from Oracle Clusterware management.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl disable rhpserver [-node node_name]
```

Usage Notes

Optionally, you can specify a node on which you want to disable the Fleet Patching and Provisioning Server.

Example

An example of this command is:

```
$ srvctl disable rhpsrvr -node clusterNode01
```

srvctl enable rhpsrvr

Use this command to enable a Fleet Patching and Provisioning Server on a specific node.



Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl enable rhpsrvr [-node node_name]
```

Usage Notes

Optionally, you can specify a node on which you want to enable the Fleet Patching and Provisioning Server. If you do not specify a node, then SRVCTL enables all of the Fleet Patching and Provisioning Servers in the Oracle Clusterware configuration.

Example

An example of this command is:

```
$ srvctl enable rhpsrvr -node clusterNode03
```

srvctl modify rhpsrvr

Modifies the storage for a Fleet Patching and Provisioning Server.

Syntax

```
srvctl modify rhpsrvr [-port rmi_port [-force]] [-email email_address  
-mailserver mail_server_address -mailserverport mail_server_port]  
[-pl_port RHP_progress_listener_port] [-clport RHP_copy_listener_port]  
[-enableTLS {YES|NO}] [-enableHTTPS '{YES|NO}'] [-port_range low_val-  
high_val]
```

Parameters

Table G-88 `srvctl modify rhpserver` Command Parameters

Parameter	Description
<code>-port rmi_port</code>	Modify the RMI port number used by the Fleet Patching and Provisioning Server.
<code>-force</code>	Specify this parameter to stop and restart the resource to effect a change.
<code>-email email_address</code>	Modify an email address.
<code>-mailserver mail_server_address</code>	Modify the mail server address.
<code>-mailserverport mail_server_port</code>	Modify the mail server port number.
<code>-pl_port RHP_progress_listener_port</code>	Modify the Fleet Patching and Provisioning progress listener port number.
<code>-clport RHP_copy_listener_port</code>	Modify the Fleet Patching and Provisioning copy listener port number.
<code>-enableTLS {YES NO}</code>	Enable Fleet Patching and Provisioning transport level security
<code>-enableHTTPS '{YES NO}'</code>	Enable Fleet Patching and Provisioning transport layer security for HTTP.
<code>-port_range low_val-high_val</code>	Modify the lower and upper port values for the ports to use for file transfer.

srvctl relocate rhpserver

Relocates a Fleet Patching and Provisioning Server to a different node.

 **Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl relocate rhpserver [-node node_name]
```

Usage Notes

Optionally, you can specify a node to which you want to relocate the Fleet Patching and Provisioning Server.

Example

An example of this command is:

```
$ srvctl relocate rhpserver -node crsNode03
```

srvctl remove rhpserver

Removes a Fleet Patching and Provisioning Server instance configured for the cluster.

Syntax

```
srvctl remove rhpserver [-resource] [-force] [-verbose]
```

Parameters**Table G-89** srvctl remove rhpserver Command Parameters

Parameter	Description
-resource	Optionally, you can remove the Fleet Patching and Provisioning Server resource only, leaving its repository unaffected.
-force	Forcibly remove the Fleet Patching and Provisioning Server, ignoring any dependencies.
-verbose	Print verbose output.

srvctl start rhpserver

Starts a Fleet Patching and Provisioning Server on a specific node.

**Note:**

This command is only available with Oracle Clusterware.

Syntax

```
srvctl start rhpserver [-node node_name]
```

Usage Notes

Optionally, you can specify a node on which you want to start the Fleet Patching and Provisioning Server.

Examples

The following example starts the Fleet Patching and Provisioning Server on a node named `clusterNode03`:

```
$ srvctl start rhpserver -node clusterNode03
```

srvctl status rhpserver

Displays the current state of a Fleet Patching and Provisioning Server.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl status rhpserver
```

Usage Notes

This command does not accept any parameters except for `-help`.

Example

This command returns output similar to the following:

```
$ srvctl status rhpserver
```

```
Fleet Patching and Provisioning Server is enabled  
Fleet Patching and Provisioning Server is running on node mjk1270089
```

srvctl stop rhpserver

Stops a Fleet Patching and Provisioning Server that is in a running or starting state.

Note:

This command is only available with Oracle Clusterware.

Syntax

```
srvctl stop rhpserver
```

Usage Notes

This command does not accept any parameter except for `-help`.

srvctl add vm

Adds a virtual machine resource to Oracle Clusterware management.

Syntax

```
srvctl add vm -name resource_name -vm "list_of_vms" [-serverpool
server_pool_name | -category server_category
| -node "node_list"] [-stoptimeout stop_timeout] [-checkinterval
check_interval]
```

Parameters

Table G-90 srvctl add vm Command Parameters

Parameter	Description
<code>-name <i>resource_name</i></code>	Specify a unique name for the virtual machine resource. If the name is not unique, then the command results in an error.
<code>-vm "<i>list_of_vms</i>"</code>	Specify a comma-delimited list of virtual machine names or virtual machine IDs that you want to add to the virtual machine resource. A list must be enclosed in double quotation marks (" "). Currently, Oracle Clusterware gives preference to virtual machines at the beginning of this list, so you should list your most important virtual machines first.
<code>-serverpool <i>server_pool_name</i> - category <i>server_category</i> - node "<i>node_list</i>"</code>	Optionally, you can specify the name of a server pool, server category, or list of nodes to which to add the virtual machine resource. A list of nodes must be enclosed in double quotation marks (" "). If you choose to specify a list of nodes, then the number of nodes must be greater than or equal to the number of virtual machine resources you specify.
<code>-stoptimeout <i>stop_timeout</i></code>	Optionally, you can specify a stop timeout, in seconds. The default is 180 if you do not assign a value to the parameter.
<code>-checkinterval <i>check_interval</i></code>	Optionally, you can specify an interval between checks, in seconds. The default is 10 if you do not assign a value to the parameter.

Usage Notes

You must run `srvctl add ovmm` and add virtual machine manager credentials before you add a virtual machine resource.

Example

An example of this command is:

```
$ srvctl add vm -name vmres -vm vm1 -node node1
```


srvctl check vm

Checks the status of a specific virtual machine or a virtual machine instance running on a specific node.

Syntax

```
srvctl check vm -name resource_name [-vm vm_name | -node node_name]
```

Parameters

Table G-91 srvctl check vm Command Parameters

Parameter	Description
-name <i>resource_name</i>	Specify the name of the virtual machine resource for which you want to check the status.
-vm <i>vm_name</i>	Specify the name or ID of the virtual machine on which the named resource resides.
-node <i>node_name</i>	Alternatively, specify the name of the node on which the virtual machine resides.

srvctl config vm

Lists the current configuration of a particular virtual machine.

Syntax

```
srvctl config vm [-name unique_name]
```

Usage Notes

Specify the unique name of the virtual machine resource for which you want to display the configuration information. If you do not specify a particular virtual machine resource, then the command displays all of the virtual machine resources in the cluster.

Example

This command displays output similar to the following, depending on the command syntax used:

```
$ srvctl config vm -name vmres1
```

```
VM resource name: vmres1
VMs: vm1, y, vm3
VM Names: x, y, z
VM IDs: vm1, vm2, vm3
Server pool: vmpool
Server category:
Nodes:
Stop Timeout: 7
```

```

Check Interval: 5
VM resource "vmres1" is enabled.
VM resource is individually enabled on nodes:
VM resource is individually disabled on nodes:
VM resource is individually disabled for VMs: vm2,vm3
$

```

```
$ srvctl config vm
```

```

vmres1
vmres2
vmres3
$

```

srvctl disable vm

Disables a virtual machine resource.

Syntax

```
srvctl disable vm -name unique_name [-vm "vm_list" | -node node_name]
```

Parameters

Table G-92 srvctl disable vm Command Parameters

Parameter	Description
-name <i>unique_name</i>	Specify the unique name of the virtual machine resource you want to disable.
-vm " <i>vm_list</i> "	Specify the name or ID, or a list of names or IDs enclosed in double quotation marks (" ") of the virtual machines on which the named resource resides. If you specify this parameter, then SRVCTL enables the cardinality instance matching that name or ID.
-node <i>node_name</i>	Alternatively, specify the name of the node on which the virtual machine resides. If you specify this parameter, then SRVCTL does not allow any instances of the specified resource to run on that node.

Usage Notes

If you do not specify any of the optional parameters, then SRVCTL disables the virtual machine resource, and the enabled or disabled states of all the nodes are purged, but the state of the virtual machines within the resource remains unchanged.

Example

An example of this command is:

```
$ srvctl disable vm -name vmres -vm "vm1,vm2,vm3"
```

srvctl enable vm

Enables a virtual machine resource.

Syntax

```
srvctl enable vm -name unique_name [-vm "vm_list" | -node node_name]
```

Parameters

Table G-93 srvctl enable vm Command Parameters

Parameter	Description
-name <i>unique_name</i>	Specify the name of the virtual machine resource you want to enable.
-vm <i>vm_list</i>	Specify the name or ID, or a list of names or IDs enclosed in double quotation marks (" ") of the virtual machines on which the named resource resides. If you specify this parameter, then SRVCTL disables the cardinality instance matching that name or ID.
-node <i>node_name</i>	Alternatively, specify the name of the node on which the virtual machine resides. If you specify this parameter, then SRVCTL does not allow any instances of the specified resource to run on that node.

Usage Notes

If you do not specify any of the optional parameters, then SRVCTL enables the virtual machine resource, and the enabled or disabled states of all the nodes are purged, but the state of the virtual machines within the resource remains unchanged.

Example

An example of this command is:

```
$ srvctl enable -name vmres -node node3
```

srvctl modify vm

Modifies the virtual machine resource configuration.

Syntax

```
srvctl modify vm -name unique_name [-addvm "list_of_vms" | -removevm "list_of_vms"  
[-serverpool server_pool_name | -category server_category | -node "node_list" ]  
[-stoptimeout stop_timeout] [-checkinterval check_interval]
```

Parameters

Table G-94 `srvctl modify vm` Command Parameters

Parameter	Description
<code>-name unique_name</code>	Specify the unique name of the virtual machine resource you want to modify.
<code>-addvm "list_of_vms"</code>	Specify a comma-delimited list of virtual machine names or virtual machine IDs, enclosed in double quotation marks (" "), that you want to add to the virtual machine resource. You can add virtual machines to a virtual machine resource at any time regardless of whether the virtual machine resource instance is running. Adding virtual machines creates new cardinality IDs.
<code>-removevms "list_of_vms"</code>	Specify a comma-delimited list of virtual machine names or virtual machine IDs that you want to remove from the virtual machine resource. You cannot remove a virtual machine if there are existing instances of the virtual machine resource still running. An attempt to do so results in an error.
<code>-serverpool server_pool_name -category server_category -node "node_list"</code>	You can modify the server pool, server category, or list of nodes to which the virtual machine resource belongs. A list of nodes must be enclosed in double quotation marks (" "). If you choose to specify a list of nodes, then the number of nodes must be greater than or equal to the number of virtual machine resources you specify.
<code>-stoptimeout stop_timeout</code>	You can modify the stop timeout, in seconds. The default is 180 if you do not assign a value to the parameter.
<code>-checkinterval check_interval</code>	You can modify the interval between checks, in seconds. The default is 10 if you do not assign a value to the parameter.

Example

An example of this command is:

```
$ srvctl modify vm -name vmres -addvm vm2 -removevm vm1 -stoptimeout 93
```

srvctl relocate vm

Relocates a running virtual machine resource to another node, or relocates the virtual machine resource running on the specified node to another node.

Syntax

```
srvctl relocate vm -name unique_name {-vm vm_name | -srcnode source_node_name}
    -node destination_node_name]
```

Parameters

Table G-95 `srvctl relocate vm` Command Parameters

Parameter	Description
<code>-name unique_name</code>	Specify the unique name of the virtual machine resource you want to relocate.
<code>-vm vm_name</code>	Specify the name or ID of the virtual machine on which the named resource resides to which you want to relocate the virtual machine resource, together with the name of the node to which you want to relocate the virtual machine resource.
<code>-srcnode source_node_name</code>	Alternatively, you can specify a node from which you want to relocate a virtual machine resource.
<code>-node destination_node_name</code>	Optionally, you can specify the name of a destination node to which you want to relocate the virtual machine resource.

Example

An example of this command is:

```
$ srvctl relocate -name vmres -vm vm1 -node node3
```

srvctl remove vm

Removes a specific virtual machine resource.

Syntax

```
srvctl remove vm -name unique_name [-force]
```

Usage Notes

Specify the unique name of the virtual machine resource you want to remove and, optionally, you can forcibly remove a running virtual machine resource.

srvctl start vm

Starts a specific virtual machine resource on a node.

Syntax

```
srvctl start vm -name unique_name [-vm vm_name -node node_name]
```

Parameters

Table G-96 `srvctl start vm` Command Parameters

Parameter	Description
<code>-name resource_name</code>	Specify the unique name of the virtual machine resource you want to start.
<code>-vm vm_name</code>	Optionally, you can specify the virtual machine on which the virtual machine resource resides that you want to start.
<code>-node node_name</code>	Optionally, you can specify the name of a node on which a virtual machine resides that contains the virtual machine resource you want to start.

Usage Notes

You can use the `-vm` and `-node` parameters, together, to start a particular virtual machine resource on a particular node. You can also use either the `-vm` parameter or the `-node` parameter to start a particular virtual machine resource or the virtual machine resource that resides on a particular node, respectively.

Example

An example of this command is:

```
$ srvctl start vm -name vmres -vm vm3 -node node3
```

srvctl status vm

Displays the currently known state of the virtual machine resource.

Syntax

```
srvctl status vm -name unique_name [-vm vm_name | -node node_name]
```

Parameters

Table G-97 `srvctl status vm` Command Parameters

Parameter	Description
<code>-name unique_name</code>	Specify the unique name of the virtual machine resource for which you want to display the status.
<code>-vm vm_name</code>	Optionally, you can specify the virtual machine on which the virtual machine resource resides that you want to display the status.
<code>-node node_name</code>	Optionally, you can specify the name of a node on which a virtual machine resides that contains the virtual machine resource you want to display the status.

Examples

This command returns output similar to following, depending on which parameters you specify:

```
$ srvctl status vm -name vmres1 -vm vm1
```

```
Virtual machine "vm1" of VM resource "vmres1" is running on node aime1.  
$
```

```
$ srvctl status vm -name vmres1
```

```
Virtual machine "vm1" of VM resource "vmres1" is running on node aime1.  
Virtual machine "vm2" of VM resource "vmres1" is not running.  
Virtual machine "vm3" of VM resource "vmres1" is running on node aime3  
$
```

```
$ srvctl status vm -name vmres1 -vm vm1 -S 1
```

```
result: vm_resource={vmres1} inst_name={vm1} node_name={aime1} up={true}  
$
```

srvctl stop vm

Stops a specific virtual machine resource, or one on a specific virtual machine or node.

Syntax

```
srvctl stop vm -name unique_name [-vm vm_name | -node node_name]
```

Parameters

Table G-98 srvctl stop vm Command Parameters

Parameter	Description
-name <i>unique_name</i>	Specify the unique name of the virtual machine resource you want to stop.
-vm <i>vm_name</i>	Optionally, you can specify the virtual machine on which the virtual machine resource resides that you want to stop.
-node <i>node_name</i>	Optionally, you can specify the name of a node on which a virtual machine resides that contains the virtual machine resource you want to stop.

Example

An example of this command is:

```
$ srvctl stop vm -name vmres1 -node node1
```

H

Oracle Clusterware Agent Framework C Application Program Interfaces

Oracle Clusterware manages applications registered as resources with Oracle Clusterware and must have access to application-specific primitives that have the ability to start, stop, and monitor a specific resource. Oracle Clusterware runs all resource-specific commands through an entity called an *agent*.

An agent is a program that contains the agent framework and user code to manage resources. The agent framework is a C library that enables you to plug in your application-specific code to manage customized applications. You program all of the actual application management functions in C or C++, such as starting, stopping, and checking the health of an application, into the agent. These functions are referred to as *entry points*. The agent framework is responsible for invoking these entry point functions on behalf of Oracle Clusterware. Agent developers can use these entry points to plug in the required functionality for a specific resource regarding how to start, stop, and monitor a resource. Agents are capable of managing multiple resources.

This appendix includes the following topics:

- [Agent Framework Data Types](#)
- [Agent Framework Context Initialization and Persistence](#)
- [Prototype for C and C++ Entry Point Functions](#)
- [C and C++ Entry Point Types and Codes](#)
- [C and C++ Entry Point Function Return Values](#)
- [Multithreading Considerations](#)
- [Deprecated APIs](#)
- [API Reference](#)
- [Agent Example](#)

Agent Framework Data Types

The Oracle Clusterware agent framework uses Oracle-defined portable data types to represent strings and numbers. By using these data types, you can build and execute agent programs on multiple platforms. [Table H-1](#) lists and describes the Oracle data types used in the agent framework.

Table H-1 Oracle Data Types Used in the Agent Framework

Oracle Data Type	Description
ub4	Unsigned 32-bit integer
sb4	Signed 32-bit integer

Table H-1 (Cont.) Oracle Data Types Used in the Agent Framework

Oracle Data Type	Description
oratext*	C style null-terminated strings, explicit cast to char * required

The agent framework APIs return a status value of type `clsagfwret`, as follows:

`CLSAGFW_SUCCESS`: Function executed successfully

`CLSAGFW_FAIL`: There was error in the parameters or in execution of the function

String values returned by an API are only accessible in the scope of the current command entry point. Agents can copy the returned string value to its own buffers to retain the value.

Agent Framework Context Initialization and Persistence

The agent framework is initialized by the `clsagfw_init()` function call. There is no context that you need to maintain. Action entry point functions are passed a pointer to the agent framework context (type `clsagfw_aectx`). This context is specific to the entry point being called and cannot be used beyond the scope of the entry point. Other agent framework APIs take this context pointer as a parameter. This is an opaque data structure that the agent framework maintains but does not explicitly initialize or destroy.

Prototype for C and C++ Entry Point Functions

The agent implements the entry point functions for the resource management commands and registers these functions with the agent framework. The agent framework calls an entry point function to execute a command or action on a resource. The entry point functions need to conform to the following prototype and convention:

```
typedef ub4 (*clsagfw_action_entry) (const clsagfw_aectx *aectx)
```

The entry point function is called with a pointer to a context area that the agent framework maintains. This function should return a value indicating the status of execution of the entry point. For the CHECK entry point, the function returns the current state of the resource.

If you develop the agent using C++, then you declare the entry point function prototypes using the `extern 'C'` keyword.

C and C++ Entry Point Types and Codes

The agent framework currently defines seven entry point types corresponding to resource management commands and operations performed on a resource. Agents must register entry points for START, STOP, CHECK, and CLEAN commands. Other command entry points are optional. The agent registers entry points by calling the `clsagfw_set_entrypoint()` API for each resource type and command. [Table H-2](#) lists and describes entry point types and their equivalent type codes.

Table H-2 Entry Point Types

Name	Description	Type Code
START	Start a resource and bring it online	CLSAGFW_ACTION_START
STOP	Stop a resource and take it offline	CLSAGFW_ACTION_STOP
CLEAN	Clean up after a failed command on the resource	CLSAGFW_ACTION_CLEAN
CHECK	Check the state of the resource	CLSAGFW_ACTION_CHECK
ABORT	End an executing command that has timed out	CLSAGFW_ACTION_ABORT
DELETE	Resource is about to be deleted	CLSAGFW_ACTION_RES_DELETED
MODIFY	Resource is modified	CLSAGFW_RESATTR_MODIFIED
ACTION	Invoke custom action on the resource	CLSAGFW_RES_ACTION

**See Also:**

"[Oracle Clusterware Resources and Agents](#)" for more information about entry points

C and C++ Entry Point Function Return Values

All entry point functions, except for the CHECK entry point, return one of the following values to indicate the status of execution of the entry point:

- **CLSAGFW_AE_SUCCESS:** Entry point execution completed successfully
- **CLSAGFW_AE_FAIL:** Entry point execution failed

The agent framework calls the CHECK entry point function to monitor the state of the resource. This function returns one of the following values that represents the current state of the resource:

```
CLSAGFW_ONLINE
CLSAGFW_UNPLANNED_OFFLINE
CLSAGFW_PLANNED_OFFLINE
CLSAGFW_UNKNOWN
CLSAGFW_PARTIAL
CLSAGFW_FAILED
```

**See Also:**

"[Agents in Oracle Clusterware](#)" for more information about evaluating and returning the state of a resource

Multithreading Considerations

The Oracle Clusterware agent framework is multithreaded by design. The agent code can also start new threads to execute application-specific functionality, such as monitoring external event sources. The agent framework executes only one entry point of a resource at a time, with the only exception being the ABORT entry point which is invoked to end a currently active action on a resource. C and C++ developers must ensure that all agent code is multithread safe. Access to global variables and other resources must be serialized using locking mechanisms, such as mutexes. Other external C or C++ libraries included by the agent should also be multithread safe.

Deprecated APIs

Table H-3 lists the deprecated agent framework APIs and the corresponding replacement APIs for Oracle Clusterware.

Table H-3 Deprecated Agent Framework APIs

Deprecated API	Replacement
<code>clsagfw_exit()</code>	<code>clsagfw_exit2()</code>
<code>clsagfw_send_status()</code>	<code>clsagfw_send_status2()</code>

API Reference

This section lists and describes the following functions:

- `clsagfw_add_type()`
- `clsagfw_check_resource()`
- `clsagfw_create_attr_iterator()`
- `clsagfw_delete_cookie()`
- `clsagfw_exit2()`
- `clsagfw_get_attr_from_iterator()`
- `clsagfw_get_attrvalue()`
- `clsagfw_get_check_type()`
- `clsagfw_get_cmdid()`
- `clsagfw_get_cookie()`
- `clsagfw_get_request_action_name()`
- `clsagfw_get_resource_id()`
- `clsagfw_get_resource_name()`
- `clsagfw_get_retry_count()`
- `clsagfw_get_type_name()`

- [clsagfw_init\(\)](#)
- [clsagfw_is_cmd_timedout\(\)](#)
- [clsagfw_log\(\)](#)
- [clsagfw_modify_attribute\(\)](#)
- [clsagfw_reset_attr_iterator\(\)](#)
- [clsagfw_send_status2\(\)](#)
- [clsagfw_set_cookie\(\)](#)
- [clsagfw_set_entrypoint\(\)](#)
- [clsagfw_set_exitcb\(\)](#)
- [clsagfw_set_resource_state_label\(\)](#)
- [clsagfw_startup\(\)](#)

clsagfw_add_type()

Registers a resource type that the agent manages.

Syntax

```
clsagfwret clsagfw_add_type(const oratext *type_name);
```

Input

`type_name`: Resource type name that the agent is to manage

Usage Notes

On startup, an agent must register all the resource types that it manages. A single agent can manage multiple resource types and multiple resources.

clsagfw_check_resource()

Initiates a check action on a specific resource.

Syntax

```
void clsagfw_check_resource(const oratext *pResId)
```

Input

`pResId`: Identification of the resource

Usage Notes

Agents can use this function in entry points or in other background threads to initiate a check action on a specific resource. The agent framework schedules and executes the check action in another thread. This function has no effect if invoked from a CHECK entry point.

clsagfw_create_attr_iterator()

Creates an iterator to access all the attributes of the current resource.

Syntax

```
clsagfwret clsagfw_create_attr_iterator(const clsagfw_aectx *pAeCtx,  
                                       clsagfw_ai_flags  flags);
```

Input

- pAeCtx: Agent framework context pointer that is passed to entry points
- flags:
 - CLSAGFW_ALL_ATTRS: All attributes of the resource
 - CLSAGFW_MODIFIED_ATTRS: Only modified attributes

Usage Notes

Agents call this function to create an iterator to navigate through all the attributes of the resource. This is useful for agents managing multiple resources of different types. The iterator is an internal iterator maintained in the agent framework context. Each attribute is retrieved by calling the `clsagfw_get_attr_from_iterator()` API. Modified attributes can be accessed only in the MODIFY entry point. At any given time, an entry point can create only one iterator. All subsequent attempts to create an iterator gets the pointer to the iterator that is already created.

clsagfw_delete_cookie()

Deletes a cookie stored in the agent framework.

Syntax

```
clsagfwret clsagfw_delete_cookie(const oratext *key);
```

Input

key: Key of the cookie, null terminated string

Usage Notes

This function deletes the cookie and frees all the memory used by the cookie and its value.

clsagfw_exit2()

Terminates the agent process with resource retry feedback.

Syntax

```
void clsagfw_exit2(const oratext *rid, sb4 exitCode, bool retry);
```

Input

- `rid`: Resource ID of the resource causing the irrecoverable error
- `exitCode`: Process exit code
- `retry`: option to retry resource

Usage Notes

An agent can encounter a irrecoverable error when executing an entry point for a resource and then must terminate the current process. An agent can use this API to end the agent process and provide feedback to the agent framework on the resource that is the cause for the irrecoverable error. If the `retry` parameter is set to `true`, then the current command on the resource is retried when the agent process restarts. If the parameter is set to `false`, then the current resource command is not retried and monitoring of the resource is disabled. This option can be used when a particular resource is the cause of repeated failures of an agent. The Oracle Clusterware server keeps track of the resources causing agent exits and automatically disables a resource if the failure rate is high. To re-enable a resource disabled due to high failure rate, must issue an explicit request to start the resource (for example, using `CRSCTL`).

The Oracle Clusterware server restarts the agent to continue monitoring resources managed by the agent.

`clsagfw_get_attr_from_iterator()`

Returns the next attribute name and value from the created iterator.

Syntax

```
clsagfwret clsagfw_get_attr_from_iterator(const clsagfw_aectx *pAeCtx,  
                                         const oratext **argName,  
                                         const oratext **argValue);
```

Input

`pAeCtx`: Agent framework context pointer that is passed to entry points

Returns

- `argname`: Attribute name
- `argValue`: Attribute value

Usage Notes

Use this function to retrieve attributes after the iterator has been created in the agent framework context. The iterator is advanced to the next attribute when this function returns. If the API returns `CLSAGFW_FAIL`, then there are no more attributes to retrieve for the resource.

`clsagfw_get_attrvalue()`

Obtains the value of an attribute of the current resource.

Syntax

```
void clsagfw_get_attrvalue(const clsagfw_aectx *pAeCtx,  
                          const oratext *pAttrName,  
                          const oratext **pAttrValue);
```

Input

- `pAeCtx`: Agent framework context pointer that is passed to entry points
- `pAttrName`: Name of attribute to retrieve

Returns

`pAttrValue`: Value of the specified attribute

Usage Notes

Agents use this API to retrieve the value of a single attribute of the current command resource.

clsagfw_get_check_type()

Returns the type of the check command currently being executed.

Syntax

```
clsagfw_checktype  
clsagfw_get_check_type(const clsagfw_aectx *pAeCtx);
```

Input

`pAeCtx`: Agent framework context pointer that is passed to entry points

Usage Notes

The agent framework invokes the CHECK entry point of a resource for multiple conditions. By using this function, the agent can identify the exact condition for the current CHECK entry point and perform appropriate actions.

Possible conditions include:

- **CLSAGFW_INITIAL_CHECK**: Check entry point invoked as part of the initial probe stage when the Oracle Clusterware server is started or restarted.
- **CLSAGFW_CHECK_AFTER_CMD**: Check entry point invoked to follow up the prior action (such as start or stop) to determine the resultant resource state after the action execution.
- **CLSAGFW_PERIODIC_CHECK**: Check entry point invoked by the agent framework at periodic intervals to monitor the state of a resource. The time interval is specified in the `CHECK_INTERVAL` attribute of the resource.
- **CLSAGFW_EXTERNAL_CHECK**: Check entry point invoked by agent framework when it receives an explicit check request from external entities, such as CRSCCTL.

- **CLSAGFW_CHECKFROM_EP**: Check action initiated by the agent itself by calling the `clsagfw_check_resource()` API.

clsagfw_get_cmdid()

Returns the type of the command for which the entry point is invoked.

Syntax

```
clsagfw_aecode clsagfw_get_cmdid(const clsagfw_aectx *pAeCtx);
```

Input

`pAeCtx`: Agent framework context pointer that is passed to entry points

Usage Notes

The agent can call this API in the ABORT entry point to determine the resource command that is being aborted.

clsagfw_get_cookie()

Retrieves the value of a cookie stored in the agent framework.

Syntax

```
clsagfwret clsagfwret clsagfw_set_cookie(const oratext *key,  
                                         const void **value);
```

Input

`key`: Key of the cookie, null terminated string

`Value`: Pointer to the value of the cookie

clsagfw_get_request_action_name()

Returns the name of the custom action to be executed on a resource.

Syntax

```
const oratext *  
clsagfw_get_request_action_name(const clsagfw_aectx *pAeCtx);
```

Input

`pAeCtx`: Agent framework context that is passed to user entry points

Usage Notes

This function is called in the ACTION entry point to retrieve the action name that you specified in a request action command. You configure the list of actions for a resource in the ACTION attribute of the resource.

clsagfw_get_resource_id()

Retrieves the ID of the resource for which the entry point is being executed.

Syntax

```
const oratext* clsagfw_get_resource_id(const clsagfw_aectx *pAeCtx);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

Usage Notes

This function returns the Oracle Clusterware-generated resource ID. To retrieve the public display name of the resource, use the [clsagfw_get_resource_name\(\)](#) API.

clsagfw_get_resource_name()

Retrieves the resource name of the current command.

Syntax

```
const oratext*  
  clsagfw_get_resource_name(const clsagfw_aectx *pAeCtx);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

Usage Notes

This function returns the name of the resource for which the current command and entry point are being executed.

clsagfw_get_retry_count()

Returns the retry attempt number for the current command.

Syntax

```
ub4 clsagfw_get_retry_count(const clsagfw_aectx *pAeCtx)
```

Input

pAeCtx: Agent framework context that is passed to user endpoints

Usage Notes

Agents can call this function in endpoints to check if the current command is being retried. An agent process or Oracle Clusterware server process can terminate in the middle of executing a command on a resource. Oracle Clusterware retries the failed command when the agent or server process restarts. The command may have

successfully or partially executed in the previous attempt. The agent can perform resume or clean up actions when Oracle Clusterware retries the command. The number of times a command can be retried is an internal parameter of Oracle Clusterware.

clsagfw_get_type_name()

Retrieves the type name of the resource for which the entry point is being executed.

Syntax

```
const oratext* clsagfw_get_type_name(const clsagfw_aectx *pAeCtx);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

Usage Notes

This function returns the Oracle Clusterware-generated resource ID. To retrieve the public display name of the resource, use the [clsagfw_get_resource_name\(\)](#) API.

clsagfw_init()

Initializes the agent framework.

Syntax

```
clsagfwret clsagfw_init(sb4 argc,  
                        oratext **argv,  
                        ub4 flags,  
                        const oratext *logName,  
                        void *reserved);
```

Input

argc: Number of command line arguments passed to the agent process

argv: Command line arguments passed to the agent process

flags:

logName: Name of the log file for this agent

reserved: Pass NULL

Usage Notes

Oracle Clusterware automatically starts the agent configured for a resource when actions must be performed on the resource. On startup, the agent process calls this function first to initialize the agent framework run time. The command line arguments passed to the agent process by the Oracle Clusterware server must be passed to `clsagfw_init()`. After initialization is complete, the agent can register resource types that are managed by the agent.

clsagfw_is_cmd_timedout()

Returns the cause of the ABORT entry point.

Syntax

```
bool clsagfw_is_cmd_timedout(const clsagfw_aectx *pAeCtx);
```

Input

`pAeCtx`: Agent framework context pointer that is passed to entry points

Usage Notes

Agents can use this function in the ABORT entry point to identify the cause of the ABORT entry point invocation. The ABORT entry point can either be invoked if a command execution time exceeds its time out value or if the Oracle Clusterware administrator aborts the command.

clsagfw_log()

Prints a diagnostic message to the agent log file.

Syntax

```
void clsagfw_log(const clsagfw_aectx *pAeCtx,  
                const ub4 log_level,  
                const oratext *fmt, ...);
```

Input

`pAeCtx`: Agent framework context that is passed to user entry points

`log_level`: Log level of the message in the range of 1-5

`fmt`: Message format string, according to Standard C library print specifications

`...`: Variable parameters corresponding to the format string

Usage Notes

Agents call this function to log diagnostic messages that help in debugging and understanding agent execution. You can use the `LOG_LEVEL` parameter to control the volume of log messages. You can configure the current logging level of a resource using `CRSCTL`, as follows:

```
$ crsctl set log level res "myResource1=3"
```

The misusages are logged to the agent log file in the `TODO` directory. By default, the name of the log file is the same as the agent executable name. You can change this by passing the `logName` parameter to the `clsagfw_init()` API.

clsagfw_modify_attribute()

Modifies an attribute of the current resource.

Syntax

```
void clsagfw_modify_attribute(const clsagfw_aectx *pAeCtx,  
                             const oratext *pAttrName,  
                             const oratext *pAttrValue);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

pAttrName: Name of attribute to be modified

pAttrValue: New value of attribute

Usage Notes

Agents can modify the value of an attribute of the current command resource by calling this API. A resource modification request is sent to the Oracle Clusterware server for validation and the updated attribute value is reflected in the agent only after confirmation from the Oracle Clusterware server. This function can be only called from the START, STOP, and CLEAN entry points.

clsagfw_reset_attr_iterator()

Resets the attribute iterator.

Syntax

```
clsagfwret clsagfw_reset_attr_iterator(const clsagfw_aectx *pAeCtx,  
                                       clsagfw_ai_flags flags);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

flags: Which attributes to be accessed:

CLSAGFW_ALL_ATTRS: All attributes of the resource

CLSAGFW_MODIFIED_ATTRS: Only modified attributes

Usage Notes

This function resets the iterator in the agent framework context so that agents can restart attribute navigation.

clsagfw_send_status2()

Reports progress, warning, and error messages to the Oracle Clusterware server and the user.

Syntax

```
void clsagfw_send_status2(const clsagfw_aectx* pAeCtx,  
                        const oratext*      pResId,  
                        clsagfw_status_type type,  
                        const oratext* fmtp,...);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

pResId: Resource for which the message is being reported

type: One of the following types of messages:

CLSAGFW_STATUS_PROGRESS: Progress message

CLSAGFW_STATUS_WARNING: Warning message

CLSAGFW_STATUS_ERROR: Error message

fmtp: Message format string

...: Variables corresponding to positions in the message format string

Usage Notes

During execution of a command, the agent can send informational messages to the Oracle Clusterware server and to the user performing operations on resources. These messages can inform you about execution progress and if there are any warnings or errors.

clsagfw_set_cookie()

Stores a cookie and its value in the agent framework.

Syntax

```
clsagfwret clsagfwret clsagfw_set_cookie(const oratext *key,  
                                         const void *value);
```

Input

key: Key of the cookie, null terminated string

value: Pointer to the value of the cookie to be stored

Usage Notes

The value of the saved cookie can be later retrieved by calling `clsagfw_get_cookie()`.

clsagfw_set_entrypoint()

Sets the C/C++ entry point function to be executed for a particular action on the resource.

Syntax

```
void clsagfw_set_entrypoint(const oratext *type_name,  
                           clsagfw_action_entry action_entry,  
                           clsagfw_aecode action_entrycode)
```

Input

key: Key of the cookie, null terminated string

Value: Pointer to the value of the cookie to be stored

Usage Notes

The value of the saved cookie can be later retrieved by calling `clsagfw_get_cookie()`.

clsagfw_set_exitcb()

Registers a function to be called when the agent process exits.

Syntax

```
typedef void (*clsagfw_exit_callback) (sb4 exitCode);  
void clsagfw_set_exitcb(clsagfw_exit_callback exit_func)
```

Input

exit_func: Function to be invoked when agent process exits

Usage Notes

Agents can register a callback function to be invoked when the agent framework terminates the agent process. The callback function can perform proper clean up of any internal resources instantiated by the agent. If the agent process is exiting to recover from an irrecoverable error, then the Oracle Clusterware server does not restart the agent until the old incarnation has exited.

clsagfw_set_resource_state_label()

Sets a custom label on the current resource.

Syntax

```
void clsagfw_set_resource_state_label(const clsagfw_aectx *pAeCtx,  
                                     const oratext *pLabel);
```

Input

pAeCtx: Agent framework context pointer that is passed to entry points

pLabel: Label to associate with the resource, null terminated string

Usage Notes

Agents can use this function to set a descriptive label associated with the state of the resource. This label is reported in the `STATE_DETAILS` attribute of the resource.

clsagfw_startup()

Starts the agent framework.

Syntax

```
clsagfwret clsagfw_startup(void)
```

Usage Notes

After the agent framework is initialized and the resource types and their entry points are registered, the agent calls this API to start the agent framework, which, in turn, starts the multithreaded infrastructure to monitor and perform actions on the resources.

On success, this function never returns to the calling thread. Entry points for the resource actions are invoked by the agent framework threads as and when required.

Agent Example

[Example H-1](#) details an agent implementation to manage a resource that represents a file.

Example H-1 demoagent1.cpp

```
/* Copyright (c) 2004, 2012, Oracle and/or its affiliates.
All rights reserved. */

/*
 * NAME
 * Demoagent1.cpp: Demonstrates agent to monitor a file
 *
 * DESCRIPTION
 * This is a sample program that demonstrates an agent to monitor
 * a file. The agent has the following tasks:
 * - On startup      : Create the file.
 * - On shutdown    : Delete the file.
 * - On check command : Detect whether the file is present or not.
 * - On clean command : Delete the file.
 * This program can be used as a basis to build up more complicated
 * agents that manage real-world resources.
 *

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <clsagfw.h>
```

Include the agent framework `clsagfw.h` header file.

```

#define TEST_TYPE1                (oratext *) "HOTFILE_TYPE"
#define TEST_PATHNAME_ATTR        (oratext *) "PATH_NAME"

/*
 * NAME
 *   test_agent_exit
 * DESCRIPTION
 *   Function to clean up, called by framework whenever the agent
 *   process is about to terminate
 * PARAMS
 *   exitCode, which is an sb4 describing the reason for exit.
 * RETURNS
 *   void
 */
void
test_agent_exit(sb4 exitCode)
{
    clsagfw_log(NULL, 1, (oratext *) "Demo Agent is exiting.\n");
}

```

Exit callback function registered with the agent framework.

```

/*
 * NAME
 *   type1_start
 * DESCRIPTION
 *   Action entry point for the 'start' command. Creates the file for
 *   monitoring.
 * PARAMS
 *   Pointer to the agent framework context
 * RETURNS
 *   ub4 which could be CLSAGFW_AE_FAIL or CLSAGFW_AE_SUCCESS
 *   depending on whether the action failed or succeeded.
 */
ub4
type1_start(const clsagfw_aectx *ectx)
{
    ub4 ret = CLSAGFW_AE_FAIL;
    const oratext *pResName = NULL;
    const oratext *pPathName = NULL;
    FILE *fp;

    clsagfw_log(ectx, 1, (oratext *) "Start action called..");

    /* Try to read the resource name */
    if (clsagfw_get_attrvalue(ectx, (oratext *) "NAME", &pResName) !=
        CLSAGFW_SUCCESS)
    {
        goto done;
    }

    /* Try to read the PATH_NAME attribute */
    if (clsagfw_get_attrvalue(ectx, TEST_PATHNAME_ATTR, &pPathName) !=
        CLSAGFW_SUCCESS)
    {
        goto done;
    }
}

```

Retrieve the resource name and the value of the PATH_NAME attribute using `clsagfw_get_attrvalue()`.


```

clsagfw_log(ectx, 1,
            (oratext *)"Start action arguments: resName = %s,
            pathName = %s", pResName, pPathName);

/* Try to create the file */
fp = fopen((char *)pPathName, "w");

if(!fp)
{
    /* Could not create the file */
    clsagfw_log(ectx, 1, (oratext *) "START action for resource %s:
    FAILED\n", pResName);
    ret = CLSAGFW_AE_FAIL;
}
else
{
    /* Created the file successfully */
    clsagfw_log(ectx, 1, (oratext *) "START action for resource %s:
    SUCCESS\n", pResName);
    ret = CLSAGFW_AE_SUCCESS;
    fclose(fp);
}
}

done:
return ret;
}

```

Diagnostic and error messages are logged to the agent log file by calling `clsagfw_log()`.

```

/*
 * NAME
 *   type1_stop
 * DESCRIPTION
 *   Action entry point for the 'stop' and 'clean' commands. Deletes
 *   the file being monitored.
 * PARAMS
 *   Pointer to agent framework context
 * RETURNS
 *   ub4 which could be CLSAGFW_AE_FAIL or CLSAGFW_AE_SUCCESS
 *   depending on whether the action failed or succeeded.
 */
ub4
type1_stop (const clsagfw_aectx *ectx)
{
    ub4          ret = CLSAGFW_AE_FAIL;
    const oratext *pResName = NULL;
    const oratext *pPathName = NULL;

    clsagfw_log(ectx, 1, (oratext *)"Stop action called..");

    /* Try to read the resource name */
    if (clsagfw_get_attrvalue(ectx, (oratext *)"NAME", &pResName) !=
        CLSAGFW_SUCCESS)
    {
        clsagfw_log(ectx, 1, (oratext *)"STOP action: %s: Could not
        read attribute\n", pResName);
        goto done;
    }

    /* Try to read the PATH_NAME attribute */

```

```
if (clsagfw_get_attrvalue(ectx, TEST_PATHNAME_ATTR, &pPathName) !=
    CLSAGFW_SUCCESS)
{
    pPathName = pResName;
}

clsagfw_log(ectx, 1,
            (oratext *)"Stop action arguments: resName = %s,
            pathName = %s", pResName, pPathName);

/* Try to delete the file */
if (remove((char *)pPathName))
{
    /* Could not delete the file */
    clsagfw_log(ectx, 1, (oratext *)"STOP action for resource %s:
        FAILED\n", pResName);
    ret = CLSAGFW_AE_FAIL;
}
else
{
    /* Deleted the file successfully */
    clsagfw_log(ectx, 1, (oratext *)"STOP action for resource %s:
        SUCCESS\n", pResName);
    ret = CLSAGFW_AE_SUCCESS;
}

done:
    return ret;
}

/*
 * NAME
 *   type1_check
 * DESCRIPTION
 *   Action entry point for the 'check' command. Determines if the
 *   file exists.
 * PARAMS
 *   Pointer to agent framework context.
 * RETURNS
 *   ub4 which gives the status of the resource. Check the
 *   agent framework reference for details on this function's
 *   return codes.
 */
ub4
type1_check(const clsagfw_aectx *ectx)
{
    ub4          ret = CLSAGFW_UNKNOWN;
    const oratext *pResName = NULL;
    const oratext *pPathName = NULL;
    FILE *fp;

    clsagfw_log(ectx, 1, (oratext *)"Check action called..");

    /* Try to read the resource name */
    if (clsagfw_get_attrvalue(ectx, (oratext *) "NAME", &pResName) !=
        CLSAGFW_SUCCESS)
    {
        goto done;
    }

    /* Try to read the PATH_NAME attribute */
```

```
if (clsagfw_get_attrvalue(ectx, TEST_PATHNAME_ATTR, &pPathName) !=
    CLSAGFW_SUCCESS)
{
    clsagfw_log(ectx, 1, (oratext *)"CHECK action: %s: Could not
        read attribute\n", pResName);
    goto done;
}

clsagfw_log(ectx, 1,
    (oratext *)"Check action arguments: resName = %s,
    pathName = %s", pResName, pPathName);

/* Check if the file is accessible */
fp = fopen((char *)pPathName, "r");
if (!fp)
{
    /* Could not open file */
    clsagfw_log(ectx, 1, (oratext *)"CHECK action: %s status -
        UNPLANNED_OFFLINE\n", pResName);
    ret = CLSAGFW_UNPLANNED_OFFLINE;
}
else
{
    /* Opened file successfully */
    clsagfw_log(ectx, 1, (oratext *)"CHECK action: %s status -
        ONLINE\n", pResName);
    fclose(fp);
    ret = CLSAGFW_ONLINE;
}

done:
    return ret;
}

/*
 * Initialization of the agent framework and registration of types is
 * done in main.
 */
int main(sb4 argc, oratext **argv)
{
    clsagfw_log(NULL, 1, (oratext *)" *** Agent Framework Demo Agent
        Started *** \n");

    /*
     * Initialize the agent framework
     */
    if (clsagfw_init(argc, argv, 0, NULL, 0)
        != CLSAGFW_SUCCESS)
    {
        clsagfw_log(NULL, 1, (oratext *)"Failed to initilize the agent
            framework\n");
        clsagfw_exit(-1);
    }

    /*
     * Set the exit callback function
     */
    clsagfw_set_exitchb(test_agent_exit);

    /*
     * Add the type definition to the framework

```

```
    */
    if (clsagfw_add_type(TEST_TYPE1) != CLSAGFW_SUCCESS)
    {
        clsagfw_log(NULL, 1, (oratext *)"Failed in adding type %s to the
            framework\n", TEST_TYPE1);
        clsagfw_exit(-1);
    }

    /*
    * Set all entry points for for HOTFILE_TYPE
    */
    clsagfw_set_entrypoint(TEST_TYPE1, type1_start,
        CLSAGFW_ACTION_START);
    clsagfw_set_entrypoint(TEST_TYPE1, type1_stop,
        CLSAGFW_ACTION_STOP);
    clsagfw_set_entrypoint(TEST_TYPE1, type1_check,
        CLSAGFW_ACTION_CHECK);
    clsagfw_set_entrypoint(TEST_TYPE1, type1_stop,
        CLSAGFW_ACTION_CLEAN);

    clsagfw_log(NULL, 1, (oratext *)"Added resource type [%s] to the
        agent framework\n", TEST_TYPE1);

    /*
    * All set to go, Start the framework. This function does not
    * return if the framework starts successfully.
    */
    clsagfw_startup();

    /*** NOT REACHED ***/

    return 0;
}
```

Oracle Clusterware C Application Program Interfaces

This appendix describes the Oracle Clusterware C application program interfaces (APIs). This appendix contains the following topics:

- [About the Programming Interface \(C API\) to Oracle Clusterware](#)
- [Interactive CLSCRS APIs](#)
- [Non-Interactive CLSCRS APIs](#)
- [Command Evaluation APIs](#)
- [Server Categorization APIs](#)
- [STAT3 API](#)
- [Miscellaneous APIs](#)

See Also:

[Making Applications Highly Available Using Oracle Clusterware](#) for detailed information about using Oracle Clusterware to make applications highly available

About the Programming Interface (C API) to Oracle Clusterware

This section contains information about using the programming interface (C API) to Oracle Clusterware (CLSCRS).

- [Overview](#)
- [Operational Notes](#)
- [Deprecated CLSCRS APIs](#)
- [Changes to Existing CLSCRS APIs](#)

Overview

CLSCRS is a set of C-based APIs for Oracle Clusterware. The CLSCRS APIs enable you to manage the operation of entities that are managed by Oracle Clusterware. These entities include resources, resource types, servers, and server pools. You use the APIs to register user applications with Oracle Clusterware so that the clusterware can manage them and maintain high availability. Once an application is registered, you can manage it and query the application's status. If you no longer need the application, then you can stop it and unregister it from Oracle Clusterware.

Oracle Clusterware services are provided by Cluster Ready Services that runs as part of Oracle Clusterware. The CLSCRS API uses a context that is explicitly named in all function calls. The API does not store anything at the process or thread level. You can use the callbacks for diagnostic logging.



Note:

You can install the Oracle Clusterware high availability API from the Oracle Database client installation media.

Operational Notes

This section includes the following topics:

- [Context Initialization and Persistence](#)
- [Threading Support](#)
- [CLSCRS API Data Structures](#)
- [Memory Management](#)
- [Error Handling and Tracing](#)
- [Callback Mechanism](#)
- [Filters](#)
- [Script Agent Usage](#)
- [Help Interface](#)

Context Initialization and Persistence

To use the CLSCRS APIs, you must first initialize the `clscrs` context. The calls to create and terminate this context are:

- `clscrs_init_crs`: Initializes the `clscrs` context
- `clscrs_term_crs`: Terminates the `clscrs` context

The caller is responsible for terminating the context when it is no longer needed.

Threading Support

If initialized with the `CLSCRS_FLAG_USETHREADS` flag, then the CLSCRS API may spawn threads internally. Every API function executes in the context of the calling thread. The API context object may not be used concurrently by multiple threads. However, no thread-affinity on the part of the client is required. A process may create multiple API contexts and use those on different threads, subject to the one-thread-per-one-context-at-a-time rule.

CLSCRS API Data Structures

The following entities are passed into the API calls and contain return values from the API call:

- `clscrs_sp`: A *stringpair* (`sp`) contains a name and a value string. The value can be NULL. It is created and destroyed, and its contents can be examined and the

value replaced. A stringpair can be a member of exactly one stringpair list (splist).

- `clscrs_splist`: A stringpair list (splist) is a list of zero or more stringpairs used in various contexts. An API can add stringpairs to or remove them from a stringpair list, or the API can iterate stringpairs.
- `clscrs_entity_type`: The enumeration type you can use to identify the type of Oracle Clusterware entity.

The enumeration types are defined, as follows:

```
/* enum to specify the entity type */
typedef enum
{
    clscrs_entity_res           = 1, /* resource */
    clscrs_entity_restype      = 2, /* resource type */
    clscrs_entity_serverpool   = 3, /* server pool */
    clscrs_entity_server       = 4, /* server */
    clscrs_entity_resinst      = 5, /* resource instances */
    clscrs_entity_config_policy = 6, /* configuration policy */
    clscrs_entity_config_policyset = 7, /* configuration policy
set */
    clscrs_entity_server_category = 8 /* server category */
} clscrs_entity_type;
```

- `clscrs_crsentity`: This data structure (crsentity) represents an Oracle Clusterware entity, which contains the name and additional data appropriate to the context in which the Oracle Clusterware entity is used. Sometimes an API contains Oracle Clusterware entity attribute data and other times it carries status and return messages about an operation. A single entity can be a member of exactly one `clscrs_crsentitylist`.
- `clscrs_crsentitylist`: An entity list (crsentitylist) is a data structure that contains zero or more instances of a `clscrs_crsentity`. An API can add entities to or remove them from an entity list, or the API can iterate entities.

Memory Management

The CLSCRS APIs work on elements and lists. The elements are added to lists. The memory for both elements and lists is allocated and released through explicit API calls. It is the caller's responsibility to release the memory that they allocate. However, when elements are added to lists, only the list must be destroyed: the destruction of the list destroys its elements implicitly. The elements must be destroyed when they are not added to any list. For recursive lists, destroying the parent list also destroys any lists contained within it. The `clscrs_sp` and `clscrs_crsentity` elements must be destroyed by the caller. If they are part of a `clscrs_splist` or `clscrs_crsentitylist`, destroying the list destroys the respective `clscrs_sp` and `clscrs_crsentity` entities.

For example, when a resource is created and added to a resource list, only the resource list must be destroyed, but not the individual resource. Destroying the resource list releases the memory for the individual resource, too.

Memory is allocated by the API through the following calls:

```
clscrs_sp_create()
clscrs_crsentity_create()
```

```

clscrs_serverpool_create()
clscrs_type_create()
clscrs_splist_create()
clscrs_crsentitylist_create()
clscrs_entity_id_create()

```

Each of the calls in the preceding list has a corresponding `clscrs*_destroy()` call.

Error Handling and Tracing

Interactive and non-interactive CLSCRS APIs each use a different error-handling mechanism.

For non-interactive CLSCRS APIs, the error code is returned as the return value of the function call. For example:

```

clscrsret clscrs_sp_get_value(clscrs_sp *sp, oratext **value);

```

The error code is returned as a `clscrsret` value.

For interactive CLSCRS APIs, the output result is represented, as follows:

1. The return value of the function call provides a high-level output of the request. Did the request reach the server? Was it completely successful, or completely or only partially unsuccessful? A successful return value means the request was received, processed, and the outcome was successful for all entities requested.
2. For each entity on which the request operated, there is a programmatic completion code stored in the `op_status` list. If the value is not `success`, it indicates the high-level type of the problem specific to processing the request for the particular object.
3. Optionally, the API might indicate that it wants to receive localized, human-readable error, warning, or status messages by using the callback mechanism. Each invocation of the callback provides the message, message type (severity), and the ID of the object to which the callback invocation pertains.

For example:

```

CLSCRS_STAT clscrs_register_resource2(clscrs_crsentitylist
*in_crsentitylist, uword flags,
                                     clscrs_msgf2 msgf, void *msgarg,
                                     clscrs_crsentitylist *op_status);

```

1. The function returns an error code of value `CLSCRS_STAT`.
2. The CRSD sends error messages, warning messages, and progress messages back to the client through the `clscrs_msgf2` callback. The client must implement the callback to process these messages returned by the CRSD.

3. In previous Oracle Clusterware releases, the API also contained results of each operation on the Oracle Clusterware entities as part of the `op_status` list. You can access that information using the following API:

```

clscrsret clscrs_entity_get_op_status(clscrs_entity *res, CLSCRS_STAT
*status,
                                     oratext **msg);

```

The `status` argument contains a status code about the CRSD operation on the Oracle Clusterware entity. Additionally, the `msg` argument contains a message from the CRSD about the result of the operation. Though the `op_status` list continues to contain the results of the CRSD operation for each Oracle Clusterware entity in the `msg` argument, usage of the `msg` argument to get the error codes and messages has now been deprecated and is not supported for any use of the API on a new entity. Only pre-existing use cases (for acting on resources, specifically) are supported. Use the callback function to process any messages returned by the CRSD.

Callback Mechanism

Interactive CLSCRS APIs provide a callback mechanism that the clients can use to process error messages, warning messages, and progress messages sent by the CRSD.

The signature of the callback mechanism is:

```

typedef void (*clscrs_msgf2)(void *usrp, const oratext *id, const oratext
*msg,
                             clscrs_msgtype msgtype);

```

In the preceding syntax:

- `usrp`: Is a user-supplied pointer that probably contains the context of the call
- `id`: Is the identifier of the entity to which the message corresponds
- `msg`: Is the output text
- `msgtype`: Is the type of the message; either error, warning, or progress

[Example I-1](#) describes an example of the callback mechanism.

Example I-1 Callback Mechanism

```

void myCallback(void *arg, const oratext *pId, const oratext *pMsg,
               clscrs_msgtype msgType)
{
    if (pMsg != NULL)
    {
        cout << pMsg << endl;
    }
}

```

[Example I-2](#) describes how to use the callback mechanism in an interactive API.

Example I-2 Using the Callback Mechanism In an Interactive API

```
clscrs_start_resource2(pResIdList, NULL,
                      env, myCallback, NULL,
                      0, pOpStatus);
```

You can also print debug trace messages for the API, itself by passing the `CLSCRS_FLAG_TRACE` flag when creating the context. The signature for context creation is:

```
CLSCRS_STAT clscrs_init_crs(clscrs_ctx **ctx, clscrs_msgf2 errf, void
*errCtx,
                          ub4 flags);
```

For the trace messages to work, you must specify both the `CLSCRS_FLAG_TRACE` flag and a `clscrs_msgf2` callback mechanism in the `clscrs_init_crs` API.

The `clscrs_msgf2` callback mechanism has the following signature:

```
typedef void (*clscrs_msgf)(void *usrp, const oratext *msg, sword msglen);
```

Filters

You can use filters to narrow down Oracle Clusterware entities upon which a CLSCRS API operates. Simple filters are attribute-value pairs with an operator. Operators must be surrounded by spaces, as shown in the examples. You can combine simple filters into expressions called expression filters using Boolean operators.

Supported filter operators are:

```
=
>
<
!=
co: Contains
st: Starts with
en: Ends with
```

Supported Boolean operators are AND and OR.

Examples of filters are:

- TYPE = type1
- ((TYPE = type1) AND (CHECK_INTERVAL > 50))
- (TYPE = type1) AND ((CHECK_INTERVAL > 30) OR (AUTO_START co never))
- NAME en network.res
- TYPE st ora.db

 **See Also:**

Use the `clscrs_comparator` enum and the `clscrs_operator` enum located in the `clscrsx.h` file (which you can download from the Oracle Clusterware web page at <http://www.oracle.com/goto/clusterware>) to get the correct type for the above comparators and operators, respectively, in the API calls

There are two types of filters and CLSCRS has a set of APIs to create these filters:

- **Comparison filter:** A simple filter that compares two values. For example:

```
TYPE = ora.db.type
```

Use the `clscrs_compfilter_create` API to create a comparison filter. For example, to create the `(TYPE = ora.db.type)` comparison filter:

```
clscrs_compfilter_create(ctx, clscrs_TYPE,
                        clscrs_comparator_eq, (const oratext
*)"ora.db.type",
                        &myCompFilter);
```

- **Expression filter:** A filter that is created from either a set of comparison filters or expression filters, or both. For example:

```
((TYPE = ora.db.type) AND (CHECK_INTERVAL > 50))
```

Use the `clscrs_expfilter_create` API to create a comparison filter. For example, to create an expression filter:

```
clscrs_exprfilter_create(myCompFilter1, clscrs_operator_or,
                        myCompFilter2, &myExprFilter);
```

 **See Also:**

The `clscrsx.h` file (which you can download from the Oracle Clusterware web page at <http://www.oracle.com/goto/clusterware>) for usage information for the `clscrs_compfilter_create` and `clscrs_expfilter_create` APIs

 **Note:**

Both the `clscrs_compfilter_create` and `clscrs_expfilter_create` APIs allocate memory that must be freed by calling `clscrs_filter_destroy()`.

You can use filters in the following interactive CLSCRS APIs in place of an entity list:

```

clscrs_start_resource2
clscrs_stat2
clscrs_stop_resource2
clscrs_check_resource2
clscrs_relocate_resource2

```

[Example I-3](#) describes using filters in an interactive CLSCRS API.

Example I-3 Filters In an Interactive CLSCRS API

```

clscrs_start_resource2(myCompFilter, NULL,
                      env, msgf2, NULL,
                      0, pOpStatus);

```

Script Agent Usage

When you use CLSCRS APIs inside script agent entry points, keep the following in mind:

1. Some actions, such as start, stop, and clean, are executed under a lock on the resource instance. Thus, issuing a request to the server to act on the resource directly or by extension of a relation results in a dead-lock.
2. Issuing read-only (`clscrs_stat2`) is generally safe unless it is an initial check, where the script agent must not call back on the server, because that results in a dead-lock, as well. Use the `clsagfw` APIs to query the check entry point.



See Also:

[Oracle Clusterware Resource Reference](#) for examples of script agents

Help Interface

You can find the entire list of CLSCRS APIs, including usage information for each, in the `clscrsx.h` file (which you can download from the Oracle Clusterware web page at <http://www.oracle.com/goto/clusterware>), along with a demo called `crsapp.c`.

Deprecated CLSCRS APIs

[Table I-1](#) lists the deprecated CLSCRS APIs and the corresponding replacement APIs for Oracle Clusterware.

Table I-1 Deprecated CLSCRS APIs

Deprecated API	Replacement
<code>clscrs_check_resource</code>	<code>clscrs_check_resource2</code>
<code>clscrs_entity</code>	<code>clscrs_entity_type</code>
<code>clscrs_fail_resource</code>	No replacement

Table I-1 (Cont.) Deprecated CLSCRS APIs

Deprecated API	Replacement
<code>clscrs_msgf</code>	<code>clscrs_msgf2</code>
<code>clscrs_register_resource</code>	<code>clscrs_register_resource2</code>
<code>clscrs_relocate_resource</code>	<code>clscrs_relocate_resource2</code>
<code>clscrs_res_attr_count</code>	<code>clscrs_crsentity_attr_count</code>
<code>clscrs_res_create</code>	<code>clscrs_crsentity_create</code>
<code>clscrs_res_create</code>	<code>clscrs_crsentitylist_create</code>
<code>clscrs_res_destroy</code>	<code>clscrs_crsentity_destroy</code>
<code>clscrs_res_get_attr</code>	<code>clscrs_crsentity_get_attr</code>
<code>clscrs_res_get_attr_list</code>	<code>clscrs_crsentity_get_attr_list</code>
<code>clscrs_res_get_name</code>	<code>clscrs_crsentity_get_name</code>
<code>clscrs_res_get_node_list</code>	<code>clscrs_crsentity_get_node_list</code>
<code>clscrs_res_get_op_status</code>	<code>clscrs_crsentity_get_op_status</code>
<code>clscrs_res_get_reslist</code>	<code>clscrs_crsentity_get_crsentitylist</code>
<code>clscrs_res_set_attr</code>	<code>clscrs_crsentity_set_attr</code>
<code>clscrs_res_set_attr_list</code>	<code>clscrs_crsentity_set_attr_list</code>
<code>clscrs_res_set_reslist</code>	<code>clscrs_crsentity_set_crsentitylist</code>
<code>clscrs_reslist_append</code>	<code>clscrs_crsentitylist_append</code>
<code>clscrs_reslist_count</code>	<code>clscrs_crsentitylist_count</code>
<code>clscrs_reslist_delete_res</code>	<code>clscrs_crsentitylist_delete_crsentity</code>
<code>clscrs_reslist_destroy</code>	<code>clscrs_crsentitylist_destroy</code>
<code>clscrs_reslist_find</code>	<code>clscrs_crsentitylist_find</code>

Table I-1 (Cont.) Deprecated CLSCRS APIs

Deprecated API	Replacement
<code>clscrs_reslist_first</code>	<code>clscrs_crsentitylist_first</code>
<code>clscrs_reslist_next</code>	<code>clscrs_crsentitylist_next</code>
<code>clscrs_start_resource</code>	<code>clscrs_start_resource2</code>
<code>clscrs_stat</code>	<code>clscrs_stat2</code>
<code>clscrs_stop_resource</code>	<code>clscrs_stop_resource2</code>
<code>clscrs_unregister_resource</code>	<code>clscrs_unregister_resource2</code>

Changes to Existing CLSCRS APIs

Oracle has added the following flags to the `clscrs_stat2` API:

- `CLSCRS_STATFLAG_SERVERBYCATEGORY`: Use this flag to query the servers that match a particular server category.
- `CLSCRS_STATFLAG_CATEGORYBYSERVER`: Use this flag to query the server categories that match a particular server.

Oracle has added the following flags to the `clscrs_start_resource2` and `clscrs_stop_resource2` APIs:

- `CLSCRS_FLAG_T_HA_PREPARE`: Use this flag with Transparent HA (`start2/stop2` APIs) to instruct the Oracle Clusterware daemon (CRSD) to prepare for an external management interface to act on a resource. When the call comes back, if successful, then the interface then is expected to start or stop the resource and call back with the other flag (`CLSCRS_FLAG_T_HA_FINISH`).
- `CLSCRS_FLAG_T_HA_FINISH`: Use this flag with Transparent HA with `CLSCRS_FLAG_T_HA_PREPARE`. You must use this flag in the second call to the CRSD (`start2/stop2` APIs) when the start or stop action has finished. Note that clients must indicate resource *instances* and never resources with this flag, because the CRSD must know to which instances the invocation of the API applies.
- `CLSCRS_NO_STATE_RESTORATION`: This flag is available for use to start and stop resource APIs and to instruct the CRSD to skip resource state restoration of the resources whose state was affected. That procedure is usually attempted unless you use this flag.

Interactive CLSCRS APIs

These APIs make calls to the Cluster Ready Services daemon (CRSD) to run commands.

The CRSD must be up and running for these APIs to function.

Table I-2 Summary of Interactive CLSCRS APIs for Oracle Clusterware

C API	Description
clscrs_check_resource2	Notifies Oracle Clusterware to run the check entry points for the identified resources.
clscrs_get_server_by_category	Obtains a list of servers that match a particular server category.
clscrs_is_crs_admin	Checks whether the user is an Oracle Clusterware administrator.
clscrs_register_resource2	Registers the resources in the input resource list.
clscrs_register_servercategory	Registers server categories in the input server category list.
clscrs_register_serverpool	Registers a server pool for the input list of servers.
clscrs_register_type	Registers the resource types in the input resource list.
clscrs_relocate_resource2	Relocates the list of resource identifiers.
clscrs_relocate_server	Relocates a list of servers.
clscrs_request_action	Notifies Oracle Clusterware to run a specific set of actions.
clscrs_restart_resource	Instructs Oracle Clusterware to restart a named set of resources.
clscrs_start_resource2	Notifies Oracle Clusterware to start a named set of resources.
clscrs_stat2	Obtains information about specific resources.
clscrs_stat3	Obtains information about specific entities.
clscrs_stop_resource2	Notifies Oracle Clusterware to stop a named set of resources.
clscrs_stop_resource_in_pools	Instructs Oracle Clusterware to stop specific resources in server pools.
clscrs_start_resource_in_pools	Instructs Oracle Clusterware to start specific resources in server pools.
clscrs_unregister_resource2	Unregisters the resources in the input list of resource names.
clscrs_unregister_servercategory	Unregisters server categories in the input server category list.
clscrs_unregister_serverpool	Unregisters the given server pool.
clscrs_unregister_type	Unregisters the resource types in the input list.
clscrs_whatif_add_server	Simulates what happens if you add a server.
clscrs_whatif_delete_server	Simulates what happens if you delete a server.
clscrs_whatif_fail_resource	Simulates what happens if a resource fails.
clscrs_whatif_register_resource	Simulates what happens if you register a resource.

Table I-2 (Cont.) Summary of Interactive CLSCRS APIs for Oracle Clusterware

C API	Description
<code>clscrs_whatif_register_serverpool</code>	Simulates what happens if you register a server pool.
<code>clscrs_whatif_relocate_resource</code>	Simulates what happens if you relocate a resource.
<code>clscrs_whatif_relocate_server</code>	Simulates what happens if you relocate a server.
<code>clscrs_whatif_set_activepolicy</code>	Simulates what happens if you activate a policy.
<code>clscrs_whatif_start_resource</code>	Simulates what happens if you start a resource.
<code>clscrs_whatif_stop_resource</code>	Simulates what happens if you stop a resource.
<code>clscrs_whatif_unregister_serverpool</code>	Simulates what happens if you unregister a server pool.

Non-Interactive CLSCRS APIs

You can use non-interactive CLSCRS APIs for functions such as context initialization, preparing request payloads for interactive APIs, and post-processing output of the interactive APIs. The non-interactive CLSCRS APIs do not call the CRSD.

A callback error reporting mechanism is not available for the non-interactive CLSCRS APIs. All interactive CLSCRS APIs, except `clscrs_stat2`, `clscrs_stat3`, and all `clscrs_whatif_*` APIs, use this callback mechanism. Clients of these APIs also use the callback mechanism to receive error, warning, and progress messages from the CRSD.

You can also use filters to reduce the list of Oracle Clusterware entities. You can also use filters in the interactive APIs to reduce the list of Oracle Clusterware entities.

Thread Safety

The routines provided to manage API data structures cannot be used with the same API context in multiple threads concurrently; however, no thread-affinity on the part of the client is required. If a separate API context is used in each instance, then a process may invoke these routines on multiple threads.

The following table lists describes the non-interactive CLSCRS APIs.

Table I-3 Non-Interactive CLSCRS APIs

C API	Description
<code>clscrs_action_getentity</code>	Returns the entity for the action.
<code>clscrs_action_getparams</code>	Returns the list of parameters for the action.
<code>clscrs_action_gettype</code>	Returns the type for the action.
<code>clscrs_actionlist_count</code>	Counts the number of actions in the action list.

Table I-3 (Cont.) Non-Interactive CLSCRS APIs

C API	Description
<code>clscrs_actionlist_create</code>	Creates an action list.
<code>clscrs_actionlist_destroy</code>	Destroys the action list.
<code>clscrs_actionlist_first</code>	Returns the first action in an action list.
<code>clscrs_actionlist_next</code>	Returns the next action in an action list.
<code>clscrs_actionlist_print</code>	Prints the action list.
<code>clscrs_actionlist_seqid</code>	Returns the sequence ID for the action lists
<code>clscrs_compfilter_create</code>	Constructs a simple filter that compares two values.
<code>clscrs_crsentity_attr_count</code>	Counts the number of attributes for an entity.
<code>clscrs_crsentity_create</code>	Creates a new entity (allocates memory).
<code>clscrs_crsentity_destroy</code>	Destroys an entity and frees up memory.
<code>clscrs_crsentity_get_attr</code>	Obtains the value of an entity, server pool, or server attribute.
<code>clscrs_crsentity_get_attr_list</code>	Obtains the attribute list for an entity, resource type, server pool, or server.
<code>clscrs_crsentity_get_crsentitylist</code>	Obtains the entity list for an entity.
<code>clscrs_crsentity_get_name</code>	Obtains the name of an entity.
<code>clscrs_crsentity_get_node_list</code>	Obtains a list of nodes currently hosting the entity.
<code>clscrs_crsentity_get_op_status</code>	Obtains the status of an operation for an entity.
<code>clscrs_crsentity_get_registered</code>	Obtains the registration status of an entity.
<code>clscrs_crsentity_set_attr</code>	Sets an attribute for an entity and a server pool.
<code>clscrs_crsentity_set_attr_list</code>	Sets the attribute list for an entity, resource type, server pool, or server.
<code>clscrs_crsentity_set_crsentitylist</code>	Sets the resource list for an entity.
<code>clscrs_crsentitylist_append</code>	Adds an entity to an entity list.
<code>clscrs_crsentitylist_count</code>	Counts the number of entities in an entity list.
<code>clscrs_crsentitylist_create</code>	Creates a list of entities.
<code>clscrs_crsentitylist_delete_crsentity</code>	Deletes an entity matching a given name from an entity list.
<code>clscrs_crsentitylist_destroy</code>	Destroys an entity list and frees up memory.
<code>clscrs_crsentitylist_find</code>	Finds an entity in an entity list matching a given name.
<code>clscrs_crsentitylist_first</code>	Obtains the first entity on an entity list.
<code>clscrs_crsentitylist_next</code>	Obtains the current next entity from an entity list.
<code>clscrs_entity_id_create</code>	Creates an entity identifier that identifies an Oracle Clusterware entity such as a resource, resource type, server group, and so on.

Table I-3 (Cont.) Non-Interactive CLSCRS APIs

C API	Description
<code>clscrs_entity_id_destroy</code>	Frees the memory associated with an entity identifier created from <code>clscrs_entity_id_create()</code> .
<code>clscrs_exprfilter_create</code>	Constructs an expression filter from comparison or expression filters, or both.
<code>clscrs_filter_destroy</code>	Frees the memory for a filter.
<code>clscrs_get_entity_type</code>	Obtains the entity type corresponding to the entity identifier provided.
<code>clscrs_get_fixed_attrlist</code>	Obtains the list of attributes that correspond to an attribute group identifier.
<code>clscrs_get_resource_instance_details</code>	Obtains the resource instance details, such as resource name, cardinality, and degree, from the resource instance identifier that is used.
<code>clscrs_getnodename</code>	Obtains the node name.
<code>clscrs_init_crs</code>	Initializes a context for communications with Oracle Clusterware.
<code>clscrs_sp_get</code>	Obtains the name and value components of a stringpair.
<code>clscrs_sp_get_value</code>	Obtains the value component of a stringpair.
<code>clscrs_sp_set</code>	Changes the value part of a stringpair.
<code>clscrs_splist_append</code>	Adds a new stringpair (<code>sp</code>) to a stringpair list (<code>splist</code>).
<code>clscrs_splist_count</code>	Counts the number of stringpairs (<code>sp</code>) in a stringpair list (<code>splist</code>).
<code>clscrs_splist_create</code>	Creates a new stringpair list.
<code>clscrs_splist_create_and_set</code>	Creates a new stringpair list (<code>splist</code>) and set the name and value for the first stringpair in the list.
<code>clscrs_splist_delete_sp</code>	Deletes a stringpair (<code>sp</code>) from a stringpair list (<code>splist</code>).
<code>clscrs_splist_destroy</code>	Frees the memory for a stringpair list (<code>splist</code>).
<code>clscrs_splist_find</code>	Finds the value for a stringpair (<code>sp</code>) in a stringpair list (<code>splist</code>).
<code>clscrs_splist_first</code>	Obtains the first stringpair (<code>sp</code>) from a stringpair list (<code>splist</code>).
<code>clscrs_splist_next</code>	Obtains the current next stringpair (<code>sp</code>) from a stringpair list (<code>splist</code>). Current next stringpair is effectively the next stringpair in the stringpair list. The list iterator is stored within the API and is not exposed.
<code>clscrs_splist_replace</code>	Replaces the value for a stringpair (<code>sp</code>) in a stringpair list (<code>splist</code>).

Table I-3 (Cont.) Non-Interactive CLSCRS APIs

C API	Description
<code>clscrs_term_crs</code>	Releases a context for communications with Oracle Clusterware.
<code>clscrs_type_create</code>	Creates a new resource type.
<code>clscrs_type_get_attr</code>	Obtains the value/properties of a resource type attribute.
<code>clscrs_type_set_attr</code>	Adds an attribute to a resource type.

Related Topics

- [Error Handling and Tracing](#)
- [Callback Mechanism](#)
- [Filters](#)

Command Evaluation APIs

You can use the command evaluation APIs to predict Oracle Clusterware's response to a hypothetical planned or unplanned event.

Oracle Clusterware can react to events in the system and produce a response action plan. This action plan consists of a series of resource state transitions or server pool reconfigurations, or both. The command evaluation APIs provide a mechanism to expose this action plan externally and to enable clients to predict the consequences of an event before it actually happens.

Additionally, by specifying a flag to the APIs listed in subsequent sections, you can obtain reasoned command evaluation information that explains why Oracle Clusterware made the decisions it did to form a response action plan to the event.

Command evaluation response plans are available for the following event categories:

- **Resources:** Start, stop, relocate, add, and modify
- **Server pools:** Add, remove, and modify
- **Servers:** Add, remove, and relocate
- **Policy:** Change active policy
- **Server category:** Modify

Reasoned command evaluation response plans are available for the following event categories:

- **Server pools:** Add, remove, and modify
- **Servers:** Add and remove
- **Policy:** Change active policy

Oracle Clusterware provides command evaluation output as a list of *actions*, where each action represents a specific step performed by Oracle Clusterware. Each action is encapsulated by a `clscrs_action` structure, and the `clscrs_actionlist` structure represents the entire sequence of actions. Oracle Clusterware also provides a set of

functions (`clscrs_action_*` for action structures and `clscrs_actionlist_*` for action lists) to create, examine, iterate over, and destroy these structures. Their usage is identical to that of the corresponding entity list and entity functions.

The command evaluation APIs also provide clients with the ability to make a query on the projected state of the system. The `clscrs_querylist` structure using the `stat3` format specifies the query, and the `clscrs_crsentitylist` structure provides the result. Refer to the `stat3` section for details on their usage.

Each command evaluation response that Oracle Clusterware provides includes a sequence ID, which indicates the current state of Oracle Clusterware. The sequence ID is incremented for every new event that Oracle Clusterware manages. Oracle Clusterware guarantees that, as long the sequence ID has not changed, the action plan provided will be executed, as is, for the event in question. For example, the action plan that Oracle Clusterware provides for a `whatif start resource FOO` request will be identical to the actions Oracle Clusterware takes take when an actual `start resource FOO` request is submitted, provided the sequence ID has not changed.

[Example I-4](#) describes how you can use command evaluation APIs.

Example I-4 Sample Usage of Command Evaluation API

```
boolean          tracectx = TRUE;
oratext          *resid;
clscrs_ctx       *ctx;
clscrs_env       env;
clscrs_splist   *resid_list;
clscrs_action    *cur_actn;
clscrs_actionlist *alist;
clscrs_splist    *params;

// Init crs
clscrs_init_crs(&ctx, (clscrs_msgf)clsuslztrace, &tracectx, (ub4)0);

// Init parameters to the call
clscrs_entity_id_create(ctx, "MYRES", clscrs_entity_res, &resid);
clscrs_splist_create(ctx, &resid_list);
clscrs_splist_append(resid_list, resid, NULL);
clscrs_actionlist_create(ctx, &alist);

// Make call into the what-if API
clscrs_whatif_start_resource(resid_list, nodename, flags, NULL, NULL,
alist);

// Process the resulting list of actions
for(clscrs_actionlist_first(alist,&cur_actn);cur_actn;clscrs_actionlist_next(alist,&cur_actn))
{
    params = clscrs_action_getparams(cur_actn);
    switch(clscrs_action_gettype(cur_actn))
    {
        case clscrs_actiontype_resstate:
            // Read params and do something
            break;
        case clscrs_actiontype_srvmove:
            // Read params and do something
    }
}
```

```

        break;
    case clscrs_actiontype_newgroup:
        // Read params and do something
        break;
    case clscrs_actiontype_errorcase:
        // Read params and do something
        break;
    }
}
clscrs_actionlist_destroy(alist);
clscrs_splist_destroy(resid_list);
clscrs_term_crs(&ctx);

```

Parameters for the APIs listed in this section are separated into those for which you provide input and those that display information when the function completes successfully.

clscrs_whatif_set_activepolicy

Determines the actions that Oracle Clusterware takes if you activate a specific policy.

Syntax

```

clscrs_whatif_set_activepolicy(const oratext *name, uword flags,
                               clscrs_querylist *qlist,
                               clscrs_crsentitylist *status,
                               clscrs_actionlist *alist);

```

Input

name: Name of the policy.

flags: CLSCRS_FLAG_NONE

CLSCRS_FLAG_WHATIF_VERBOSE

CLSCRS_FLAG_WHYIF: Displays reasoned command evaluation information

CLSCRS_FLAG_FORCE

qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.

alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.

CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_fail_resource

Determines the actions that Oracle Clusterware takes if specific resources fail.

Syntax

```
clscrs_whatif_fail_resource(clscrs_splist *name, const oratext *server,  
                           uword flags, clscrs_querylist *qlist,  
                           clscrs_crentitylist *status,  
                           clscrs_actionlist *alist);
```

Input

name: Resource or instance ID, or a filter.
server: Name of the server on which the resource failure occurs. NULL is allowed.
flags: CLSCRS_FLAG_NONE
CLSCRS_FLAG_WHATIF_VERBOSE
qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.
alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.
CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.
CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.
CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.
CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_register_resource

Determines the actions that Oracle Clusterware takes if you add or modify a specific resource.

Syntax

```
clscrs_whatif_register_resource(const oratext *name, clscrs_splist *attrs,  
                                uword flags, clscrs_querylist *qlist,  
                                clscrs_crentitylist *status,  
                                clscrs_actionlist *alist);
```

Input

name: Name of the resource.

`attrs`: The attributes of the specified resource.
`flags`: `CLSCRS_FLAG_NONE`
`CLSCRS_FLAG_WHATIF_VERBOSE`
`CLSCRS_FLAG_FORCE`
`CLSCRS_FLAG_REG_UPDATE` (to modify the resource)
`qlist`: Specifies the client query on Oracle Clusterware entity status. `NULL` indicates no query specified.

Output

`status`: List containing the returned entity information.
`alist`: Action list containing the actions that Oracle Clusterware will perform.

Returns

`CLSCRS_STAT_SUCCESS`: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.
`CLSCRS_STAT_AUTHORIZATION_FAILURE`: Displays when authorization fails.
`CLSCRS_STAT_CONNECTION`: Displays if there is a communication error.
`CLSCRS_STAT_WRONG_ACTIVE_VERSION`: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.
`CLSCRS_STAT_INTERNAL_ERROR`: Displays if an unexpected, non-user error occurs.
`CLSCRS_STAT_INVALID_ARGS`: Displays if you provide incorrect arguments.

`clscrs_whatif_relocate_resource`

Determines the actions that Oracle Clusterware takes if you relocate specific resources.

Syntax

```
clscrs_whatif_relocate_resource(clscrs_splist *name, const oratext
*destnode,
                                uword flags, clscrs_querylist *qlist,
                                clscrs_crsentitylist *status,
                                clscrs_actionlist *alist);
```

Input

`name`: Resource or instance ID, or a filter.
`flags`: `CLSCRS_FLAG_NONE`
`CLSCRS_FLAG_WHATIF_VERBOSE`
`CLSCRS_FLAG_FORCE`
`qlist`: Specifies the client query on Oracle Clusterware entity status. `NULL` indicates no query specified.

Output

`status`: List containing the returned entity information.
`alist`: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.

CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_start_resource

Determines the actions that Oracle Clusterware takes if you start specific resources.

Syntax

```
clscrs_whatif_start_resource(clscrs_splist *name, const oratext *node,  
                             uword flags, clscrs_querylist *qlist,  
                             clscrs_crsentitylist *status,  
                             clscrs_actionlist *alist);
```

Input

name: Resource or instance ID, or a filter.

node: Name of the node on which you want to start the resource. NULL is allowed.

flags: CLSCRS_FLAG_NONE
CLSCRS_FLAG_WHATIF_VERBOSE
CLSCRS_FLAG_FORCE

qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.

alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.

CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_stop_resource

Determines the actions that Oracle Clusterware takes if you stop specific resources.

Syntax

```
clscrs_whatif_stop_resource(clscrs_splist *name, uword flags,  
                           clscrs_querylist *qlist,  
                           clscrs_crentitylist *status,  
                           clscrs_actionlist *alist);
```

Input

name: Resource or instance ID, or a filter.

flags: CLSCRS_FLAG_NONE

CLSCRS_FLAG_WHATIF_VERBOSE

CLSCRS_FLAG_FORCE

qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.

alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.

CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_register_serverpool

Determines the actions that Oracle Clusterware takes if you register a specific server pool.

Syntax

```
clscrs_whatif_register_serverpool(const oratext *pname, clscrs_splist  
*attrs,  
                                  uword flags, clscrs_querylist *qlist,  
                                  clscrs_crentitylist *status,  
                                  clscrs_actionlist *alist);
```

Input

name: Name of the server pool.

attrs: The attributes of the specified server pool.

flags: CLSCRS_FLAG_NONE

CLSCRS_FLAG_WHATIF_VERBOSE

CLSCRS_FLAG_WHYIF: Displays reasoned command evaluation information
 CLSCRS_FLAG_FORCE
 CLSCRS_FLAG_REG_UPDATE (to modify the server pool)
 qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.
 alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.
 CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.
 CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
 CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.
 CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.
 CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_unregister_serverpool

Determines the actions that Oracle Clusterware takes if you unregister a specific server pool.

Syntax

```
clscrs_whatif_unregister_serverpool(const oratext *poolname, uword flags,
                                     clscrs_querylist *qlist,
                                     clscrs_crsentitylist *status,
                                     clscrs_actionlist *alist);
```

Input

name: Name of the server pool.
 flags: CLSCRS_FLAG_NONE
 CLSCRS_FLAG_WHATIF_VERBOSE
 CLSCRS_FLAG_WHYIF: Displays reasoned command evaluation information
 CLSCRS_FLAG_FORCE
 qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.
 alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.
CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.
CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.
CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_add_server

Determines the actions that Oracle Clusterware takes if you add a server.

Syntax

```
clscrs_whatif_add_server(const oratext *name, clscrs_splist *attrs, uword
flags,
                                clscrs_querylist *qlist, clscrs_crsentitylist
*status,
                                clscrs_actionlist *alist);
```

Input

name: Name of the server.
attrs: The attributes of the specified server.
flags: CLSCRS_FLAG_NONE
CLSCRS_FLAG_WHATIF_VERBOSE
CLSCRS_FLAG_WHYIF: Displays reasoned command evaluation information
CLSCRS_FLAG_FORCE
qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.
alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.
CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.
CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.
CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.
CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_delete_server

Determines the actions that Oracle Clusterware takes if you delete a server.

Syntax

```
clscrs_whatif_delete_server(const oratext *name, uword flags,  
                           clscrs_querylist *qlist,  
                           clscrs_crentitylist *status,  
                           clscrs_actionlist *alist);
```

Input

name: Name of the server.

flags: CLSCRS_FLAG_NONE

CLSCRS_FLAG_WHATIF_VERBOSE

CLSCRS_FLAG_WHYIF: Displays reasoned command evaluation information

CLSCRS_FLAG_FORCE

qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.

alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.

CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

clscrs_whatif_relocate_server

Determines the actions that Oracle Clusterware takes if you relocate a server to a different server pool.

Syntax

```
clscrs_whatif_relocate_server(const oratext *name, const oratext *topool,  
                              uword flags, clscrs_querylist *qlist,  
                              clscrs_crentitylist *status,  
                              clscrs_actionlist *alist);
```

Input

name: Name of the server.
 topool: The name of the server pool to which you want to relocate the server.
 flags: CLSCRS_FLAG_NONE
 CLSCRS_FLAG_WHATIF_VERBOSE
 CLSCRS_FLAG_FORCE
 qlist: Specifies the client query on Oracle Clusterware entity status. NULL indicates no query specified.

Output

status: List containing the returned entity information.
 alist: Action list containing the actions that Oracle Clusterware will perform.

Returns

CLSCRS_STAT_SUCCESS: Indicates that Oracle Clusterware completed the request successfully or that no entities matched a filter.
 CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.
 CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
 CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.
 CLSCRS_STAT_INTERNAL_ERROR: Displays if an unexpected, non-user error occurs.
 CLSCRS_STAT_INVALID_ARGS: Displays if you provide incorrect arguments.

Server Categorization APIs

Oracle Clusterware includes an entity called `clsrcs_entity_server_category`.

Parameters for the APIs listed in this section are separated into those for which you provide input and those that display information when the function completes successfully.

`clsrcs_servercategory_create`

Creates a server category.

Syntax

```
clsrcs_servercategory_create(clsrcs_ctx *ctx, const oratext *sc_name,
                             clsrcs_crsentity **sc);
```

Input

ctx: CLSCRS context.
 sc_name: Name of the server category.

Output

sc: The newly created server category.

Returns

`clscrsretSUCC`: Indicates that Oracle Clusterware completed the request successfully.
`clscrsretNOMEM`: Displays if no memory can be allocated.
`clscrsretBADCTX`: Displays if the context is NULL.
`clscrsretBADARG`: Displays if the server name is NULL.

clscrs_servercategory_destroy

Frees memory for a server category.

Syntax

```
clscrs_servercategory_destroy(clscrs_crsentity **sc);
```

Input

`sc_name`: Name of the server category you want to destroy to free up memory

Returns

`clscrsretSUCC`: Indicates that Oracle Clusterware completed the request successfully

clscrs_register_servercategory

Registers the server categories that you specify in the input server category list.

Syntax

```
clscrs_register_servercategory(clscrs_crsentitylist *in_entitylist, uword
flags,
                                clscrs_msgf2 msgf, void *msgarg,
                                clscrs_crsentitylist *op_status);
```

Input

`in_entitylist`: The list of server categories you want to register.
`flags`: `CLSCRS_FLAG_REG_UPDATE`
`CLSCRS_FLAG_QUEUE`
`CLSCRS_FLAG_FORCE`
`CLSCRS_FLAG_NONE`
`msgf`: User message callback, which can be NULL.
`msgarg`: User callback argument, which can be NULL.

Output

`op_status`: The entity list that holds the status of the register operation for each server category.

Returns

CLSCRS_STAT_SUCCESS: Indicates that all input server categories are successfully registered.

CLSCRS_STAT_FAILURE: Displays if at least one server category cannot be registered.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INVALID_ARGS: Displays if any of the server categories in the input entity list do not have attributes.

Usage Notes

- The attributes for the server category are contained in the input server category list.
- The `op_status` list contains the results of the register operation for each server category and contains no valid attributes.
- The caller must create and populate the `in_entitylist` and must create the `op_status` list. Both of these lists must be destroyed by the caller.
- The `op_status` list cannot be reused with another API call. It must be created and destroyed for each API call.
- One or more attributes of an already registered server category can be modified by passing the `CLSCRS_FLAG_REG_UPDATE` flag.
- The flags apply to all server categories in the input entity list.

clscrs_unregister_servercategory

Unregisters the server categories that you specify in the input list.

Syntax

```
clscrs_unregister_servercategory(clscrs_splist *sclist, uword flags,  
                                clscrs_msgf2 msgf, void *msgarg,  
                                clscrs_crsentitylist *op_status);
```

Input

`sclist`: The list of server categories you want to unregister.

`flags`: Specify option flags.

`msgf`: User message callback, which can be NULL.

`msgarg`: User callback argument, which can be NULL.

Output

`op_status`: The entity list that holds the status of the unregister operation for each server category.

Returns

- CLSCRS_STAT_SUCCESS: Indicates that all input server categories are successfully registered.
- CLSCRS_STAT_FAILURE: Displays if at least one server category cannot be unregistered.
- CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
- CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

Usage Notes

- The `op_status` list contains the results of the unregister operation for each server category.
- The caller must create and populate the `sclist` and must create the `op_status` list. Both of these lists must be destroyed by the caller.
- The `op_status` list cannot be reused with another API call and must be created and destroyed for each API call.

clscrs_get_server_by_category

Obtains a list of servers that match a particular server category.

Syntax

```
clscrs_get_server_by_category(clscrs_splist *in_list,
                             clscrs_crsentitylist *out_entitylist);
```

Input

`in_list`: The list of server categories or a filter.

Output

`out_entitylist`: Lists the servers matching the server category.

Returns

- CLSCRS_STAT_SUCCESS: Indicates that all input server categories are successfully registered.
- CLSCRS_STAT_FAILURE: Displays if at least one server category cannot be unregistered.
- CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

clscrs_register_server

Modifies the server attributes.

Syntax

```
clscrs_register_server(clscrs_crseentitylist *in_entitylist, uword flags,  
                      clscrs_msgf2 msgf, void *msgarg,  
                      clscrs_crseentitylist *op_status);
```

Input

`in_entitylist`: The list of server categories you want to register.

`flags`: CLSCRS_FLAG_QUEUE

CLSCRS_FLAG_FORCE

CLSCRS_FLAG_REG_UPDATE

CLSCRS_FLAG_NONE

`msgf`: User message callback, which can be NULL.

`msgarg`: User callback argument, which can be NULL.

Output

`op_status`: The entity list that holds the status of the register operation for each server.

Returns

CLSCRS_STAT_SUCCESS: Indicates that all input server categories are successfully registered.

CLSCRS_STAT_FAILURE: Displays if at least one server category cannot be registered.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

CLSCRS_STAT_INVALID_ARGS: Displays if any of the server categories in the input entity list do not have attributes.

Usage Notes

- The attributes for the server are contained in the input server list.
- The `op_status` list contains the results of the modify operation for each server but contains no valid attributes.
- The caller must create and populate the `in_entitylist` and must create the `op_status` list. Both of these lists must be destroyed by the caller.
- The `op_status` list cannot be reused with another API call and it must be created and destroyed for each API call.

Note:

Since Oracle currently supports, only, the `CLSCRS_FLAG_REG_UPDATE` flag must always be passed. The flags apply to all servers in the input entity list.

(Optional) Provide detailed information about using the API or subprogram here.

STAT3 API

Oracle Clusterware 11g release 2 (11.2) manages several entities, such as resources, server pools, and so on. However, the interfaces in that release only allowed retrieving (reading) entities by type, which meant that a single retrieval could only return entities of a single type. Therefore, clients that needed to get different types of entities and needed to have a consistent view of the data structures to make further decisions needed to rely on a work around using a special event sequence ID and, if necessary, reissue query requests several times (in theory, in a system with ever changing state/ configuration such a solution is time-unbounded).

Oracle Clusterware 12c provides a mechanism to perform a consistent read of entities of several kinds. The mechanism works on entities transparently, such that addition of new managed entities do not require any changes to the mechanism.

This is achieved by the `clscrs_stat3` API.

`clscrs_stat3`

Obtains information about the Oracle Clusterware entities identified in `qlist`.

Syntax

```
clscrs_stat3(clscrs_querylist *qlist,  
            uword flags,  
            clscrs_crsentitylist *out_entitylist);
```

Input

`qlist`: The list of Oracle Clusterware entities you want to query.

Output

`out_entitylist`: The entity list that holds the returned entity information.

Returns

- `CLSCRS_STAT_SUCCESS`: Indicates that the API successfully queried Oracle Clusterware.
- `CLSCRS_STAT_FAILURE`: Displays if there is an error querying Oracle Clusterware.
- `CLSCRS_STAT_CONNECTION`: Displays if there is a communication error.

Usage Notes

- Information is returned in `out_entitylist`. Queries are executed such that the subsequent output set is consistent.
- You create `clscrs_query` objects and append them to the `clscrs_querylist` object before passing the object to this function. Any attributes required for each entity type must be present in the `clscrs_query` object. If the attribute list is empty, then Oracle Clusterware returns all attributes.
- The `out_entitylist` must be created and passed as an empty list. Any errors for an entity are returned in the `out_entitylist`.

- The output is returned as a nested entity list. Results for individual queries are returned as respective entities in the `out_entitylist`. The output for individual queries is returned as an entity list for that specific entity. The type of results in the entity can be verified by calling `clscrs_crseentity_get_type`, to get the Oracle Clusterware entity type of the entity that is part of the `out_entitylist`.

For example, if you have two queries, one for resources and the other for resource types, then `out_entitylist` will contain two entity objects, one for each of the aforementioned Oracle Clusterware entity types. The entity list for each of these entity objects will contain the results of the queries. To determine what type of entity object a particular object is, you must call the `clscrs_crseentity_get_type` function on that entity object. If the query returns zero matches, then the size of the entity list for that entity object will be zero.

 **Note:**

Oracle supports only one `clscrs_query` object per entity type. If more than one `clscrs_query` object is created for the same entity type, then Oracle does not guarantee the `stat3` API behavior.

Miscellaneous APIs

Parameters for the APIs listed in this section are separated into those for which you provide input and those that display information when the function completes successfully.

`clscrs_get_error_details`

Returns the `clsk` exception stack if there are any failures while invoking other CLSCRS APIs.

Syntax

```
clscrs_get_error_details(oratext* error_buf, size_t* buf_size);
```

Input

`error_buf`: The buffer that will be populated with the error stack.

`buf_size`: The size of the buffer for `error_buf`. If the size of the buffer given is smaller than what is required, then the API returns a value for the necessary buffer size.

Returns

`clscrsretSUCC`: Indicates that the error stack printed successfully.

`clscrsretEMPTY`: Displays if the error stack is empty.

`clscrsretBADARG`: Displays if either `error_buf` or `buf_size` is NULL.

`clscrsretBUFFSMALL`: Displays if the buffer size that you specify is smaller than what is required.

Usage Notes

The caller is responsible for allocating memory for `error_buf`.

clscrs_request_action

Instructs Oracle Clusterware to run an action on a named set of resources.

Syntax

```
clscrs_request_action(oratext *action_name, clscrs_splist *ridlist,  
                    clscrs_env env, clscrs_msgf2 msgf,  
                    void *msgarg, uword flag, clscrs_reslist *op_status);
```

Input

`action_name`: The name of the action to be performed.
`ridlist`: The list of resources or resource instance IDs to stop, or a filter.
`env`: Specify environment arguments to the stop operation.
`msgf`: User message callback, which can be NULL.
`msgarg`: User callback argument, which can be NULL.
`flag`: Either `async` or `queue` options.

Output

`op_status`: The resource list that holds the status of the action operation for each resource.

Returns

`CLSCRS_STAT_INVALID_RESNAME`: Displays if `ridlist` is empty.
`CLSCRS_STAT_AUTHORIZATION_FAILURE`: Displays when authorization fails.
`CLSCRS_STAT_SUCCESS`: Displays if the request completes successfully for all requested entities or if no entities match a filter.
`CLSCRS_STAT_FAILURE`: Displays if at least one resource or resource ID does not stop successfully.
`CLSCRS_STAT_CONNECTION`: Displays if there is a communication error.

Usage Notes

- In the first argument, you can specify either a list of resource IDs or a filter.
- Specifying a filter enables the query and action to take place in a single call. The filter searches all registered resources.

clscrs_restart_resource

Instructs Oracle Clusterware to restart a named set of resources.

Syntax

```
clscrs_restart_resource(clscrs_splist *ridlist,  
                       clscrs_env env, clscrs_msgf2 msgf, void *msgarg,  
                       uword flag, clscrs_reslist *op_status);
```

Input

ridlist: The list of resources or resource instance IDs to restart, or a filter.
env: Specify environment arguments to the restart operation.
msgf: User message callback, which can be NULL.
msgarg: User callback argument, which can be NULL.
flag: Either `async`, `force`, or `event` options.

Output

op_status: The resource list that holds the status of the restart operation for each resource.

Returns

CLSCRS_STAT_INVALID_RESNAME: Displays if `ridlist` is empty.
CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.
CLSCRS_STAT_SUCCESS: Displays if the request completes successfully for all requested entities or if no entities match a filter.
CLSCRS_STAT_FAILURE: Displays if at least one resource or resource ID does not start successfully.
CLSCRS_STAT_CONNECTION: Displays if there is a communication error.
CLSCRS_STAT_WRONG_ACTIVE_VERSION: Displays if you use the API before you have upgraded the cluster to Oracle Clusterware 12c.

Usage Notes

- If the flag is `async`, then the `msgf` callback function you specify is never called. The API returns an `OK` status after initiating the call to Oracle Clusterware, and asynchronously executes the restarts.
- If the flag is not `async`, and `msgf` is not NULL, then the API drives `msgf` one line at a time with collected output from the restart programs. An optional `event` flag maybe passed to indicate that this is not a request to perform an action, but rather a notification that the action has already started. The flag should only be used for a narrow set of co-managed resources.
- In the first argument, you can specify either a list of resource IDs or a filter. Specifying a filter enables the query and action to take place in a single call. The filter searches all registered resources.

`clscrs_start_resource_in_pools`

Instructs Oracle Clusterware to start a named set of resources in server pools.

Syntax

```

clscrs_start_resource_in_pools(clscrs_splist *ridlist,clscrs_splist
*spoollist,
                                clscrs_env env,clscrs_msgf2 msgf2, void
*msgarg,
                                uword flags, clscrs_reslist *op_status);

```

Input

ridlist: The list of resources or resource instance IDs to start, or a filter.

spoollist: The list of server pool names where a resource that you want to start is running, or a filter.

env: Specify environment arguments to the start operation.

msgf: User message callback, which can be NULL.

msgarg: User callback argument, which can be NULL.

flag: Either `async`, `force`, or `event` options.

Output

op_status: The resource list that holds the status of the start operation for each resource.

Returns

CLSCRS_STAT_INVALID_RESNAME: Displays if `ridlist` is empty.

CLSCRS_STAT_AUTHORIZATION_FAILURE: Displays when authorization fails.

CLSCRS_STAT_SUCCESS: Displays if the request completes successfully for all requested entities or if no entities match a filter.

CLSCRS_STAT_FAILURE: Displays if at least one resource or resource ID does not start successfully.

CLSCRS_STAT_CONNECTION: Displays if there is a communication error.

Usage Notes

- Functionality is similar to `clscrs_start_resource2`, except that this function takes an extra argument—`spoollist`—which is a list of server pool names or a filter based on which a list of server pools is generated.
- This function *does not* take `node` as an argument.

clscrs_stop_resource_in_pools

Instructs Oracle Clusterware to stop a named set of resources in server pools.

Syntax

```

clscrs_stop_resource_in_pools(clscrs_splist *ridlist,clscrs_splist
*spoollist,
                                clscrs_env env,clscrs_msgf2 msgf2, void
*msgarg,
                                uword flags, clscrs_reslist *op_status);

```

Input

`ridlist`: The list of resources or resource instance IDs to stop, or a filter.
`spoolist`: The list of server pool names where a resource that you want to stop is running, or a filter.
`env`: Specify environment arguments to the stop operation.
`msgf`: User message callback, which can be NULL.
`msgarg`: User callback argument, which can be NULL.
`flag`: Either `async`, `force`, or `event` options.

Output

`op_status`: The resource list that holds the status of the stop operation for each resource.

Returns

`CLSCRS_STAT_INVALID_RESNAME`: Displays if `ridlist` is empty.
`CLSCRS_STAT_AUTHORIZATION_FAILURE`: Displays when authorization fails.
`CLSCRS_STAT_SUCCESS`: Displays if the request completes successfully for all requested entities or if no entities match a filter.
`CLSCRS_STAT_FAILURE`: Displays if at least one resource or resource ID does not stop successfully.
`CLSCRS_STAT_CONNECTION`: Displays if there is a communication error.

Usage Notes

- Functionality is similar to `clscrs_stop_resource2`, except that this function takes an extra argument—`spoolist`—which is a list of server pool names or a filter based on which a list of server pools is generated.
- This function *does not* take `node` as an argument.

J

Oracle Cluster Registry Utility Reference

This appendix describes the syntax of the Oracle Cluster Registry (OCR) configuration utility, OCRCONFIG, and troubleshooting OCR, which includes descriptions and usage information for OCRCHECK and OCRDUMP.

This appendix contains the following topics:

- [About OCRCONFIG](#)
- [OCRCONFIG Command Reference](#)
- [Troubleshooting Oracle Cluster Registry and Diagnostic Output](#)

About OCRCONFIG

Use the `ocrconfig` command to manage OCR. Using this utility you can import, export, add, delete, restore, overwrite, backup, repair, replace, move, upgrade, or downgrade OCR.

Usage Information

- The OCRCONFIG executable is located in the `Grid_home/bin` directory
- The `ocrconfig` command syntax is as follows:

```
ocrconfig -option
```

Using Utility Help

To display the help output for the OCRCONFIG utility:

```
ocrconfig -help
```

Privileges and Security

To use the OCRCONFIG utility you must be logged into the operating system as a user with administrative privileges.

Log Files

The OCRCONFIG utility creates a log file in `ORACLE_BASE/diag/crs/host_name/crs`.

To change the amount of logging, edit the path in the `ORACLE_BASE/crsdata/host_name/crsdiag/ocrconfig.ini` file. The `ORACLE_BASE`, in this case, is for the Oracle Grid Infrastructure software home.

OCRCONFIG Command Reference

This section describes the following OCRCONFIG commands:

- `ocrconfig -add`
- `ocrconfig -backuploc`
- `ocrconfig -copy`
- `ocrconfig -delete`
- `ocrconfig -downgrade`
- `ocrconfig -export`
- `ocrconfig -import`
- `ocrconfig -manualbackup`
- `ocrconfig -overwrite`
- `ocrconfig -repair`
- `ocrconfig -replace`
- `ocrconfig -restore`
- `ocrconfig -showbackup`
- `ocrconfig -showbackuploc`
- `ocrconfig -upgrade`

ocrconfig -add

Adds an OCR location to an Oracle Automatic Storage Management (Oracle ASM) disk group.

Syntax

```
ocrconfig -add +diskgroup_name
```

Usage Notes

- OCR locations that you add must exist, have sufficient permissions, and must be mounted before you can add them.
- You must run this command as `root`.
- The *diskgroup_name* variable is the name of an Oracle ASM disk group. For example:
 - `+newdg`: Ensure that the disk group exists and is mountedThe name of the disk group must be preceded by a plus sign (+).

 **See Also:**

- *Oracle Grid Infrastructure Installation and Upgrade Guide* for information about creating OCR locations and setting correct permissions
- *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle ASM disk group management

Example

To add an OCR location to the default location in Oracle ASM, data:

```
# ocrconfig -add +data
```

ocrconfig -backuploc

Specify an OCR backup directory location.

Syntax

```
ocrconfig -backuploc +disk_group_name  
ocrconfig -local -backuploc dir_name
```

Usage Notes

- You must run this command as `root`.
- Use the `-local` option to specify an OLR backup directory location.
- For an OCR backup location, the *disk_group_name* variable must be an Oracle ASM disk group. For example:

```
+bkupdg: Ensure that the Oracle ASM disk group exists and is mounted on all nodes
```

An Oracle ASM disk group name must be preceded by a plus sign (+).

- For an OLR backup location, the *dir_name* variable can be a full directory path name. For example:

```
– Grid_home/cdata/cluster3/: Ensure that the location exists
```

```
– d:\cdata\cluster3: Ensure that the location exists
```

The default location for generating OLR backups on Linux or UNIX systems is *Grid_home/cdata/host_name*, where *host_name* is the name of the node on which the OLR resides that you want to back up. The Windows default location for generating OLR backups uses the same path structure.

Examples

To specify an OCR backup location in an Oracle ASM disk group:

```
# ocrconfig -backuploc +bkupdg
```

To specify an OLR backup location:

```
# ocrconfig -local -backuploc full_directory_path
```

ocrconfig -copy

Copies an OCR backup file from one location to another.

Syntax

```
ocrconfig -copy source_file_name destination_file_name
```

Usage Notes

- You must run this command as `root`.
- The *source_file_name* must be a valid OCR backup file. For example:

- `/oradbocfs/crs/BACKUP00.ocr`
- `d:\oracle\BACKUP01.ocr`
- `+bkupdg:/aimev1/ocrbackup/day.ocr.260.739756845`

If you specify an Oracle ASM disk group, then ensure that the disk group exists and is mounted.

Examples of valid *destination_file_name* values are:

- `/oradbocfs/crs/mybkp01.ocr`
- `d:\oracle\mybkp01.ocr`
- `+bkupdg:mybkp01`

Oracle ASM disk group names must be preceded by a plus sign (+).

Examples

To copy an OCR backup file from a file system to an Oracle ASM disk group:

```
# ocrconfig -copy /oradbocfs/crs/BACKUP00.ocr +bkupdg:mybkp01.ocr
```

To copy an OCR backup file from an Oracle ASM disk group to a file system:

```
# ocrconfig -copy +bkupdg:/aimev1/ocrbackup/day.ocr.260.739756845 /  
oradbocfs/crs/mybkp01.ocr
```

To copy an OCR backup file from one Oracle ASM disk group to another Oracle ASM disk group:

```
# ocrconfig -copy +bkupdg01:/aimev1/ocrbackup/day.ocr.260.739756845  
+bkupdg02:mybkp01.ocr
```

ocrconfig -delete

Removes an OCR device or file.

Syntax

```
ocrconfig -delete +diskgroup_name
```

Usage Notes

- You must run this command as `root`.
- The `+diskgroup_name` variable is the name of an Oracle ASM disk group. For example: `+olddg`

The name of the disk group must be preceded by a plus sign (+).

Example

To remove an OCR location:

```
# ocrconfig -delete +olddg
```

ocrconfig -downgrade

Only root scripts use this command to downgrade OCR to an earlier specified version.

Syntax

```
ocrconfig -downgrade
```

Usage Notes

This command has no options.

ocrconfig -export

Exports the contents of OCR to a target file.

Syntax

```
ocrconfig [-local] -export file_name
```

Usage Notes

- You must run this command as `root`.
- Use the `-local` option to export the contents of OLR.
- The `file_name` variable can be a file name or the name of an Oracle ASM disk group (preceded by a plus sign (+)). For example:

- `/oradbocfs/crs/data.ocr`
- `d:\oracle\data.ocr`

```
– +bkupdg:data.ocr
```

Example

To export the contents of OCR to a file:

```
# ocrconfig -export d:\tmp\a
```

ocrconfig -import

Imports the contents of a target file into which you exported the contents of OCR back into OCR.

Syntax

```
ocrconfig [-local] -import file_name
```

Usage Notes

- You must run this command as `root`.
- Use the `-local` option to import the contents of OLR from a file.
- The `file_name` variable can be a file name or the name of an Oracle ASM disk group (preceded by a plus sign (+)). For example:

```
/oradbocfs/crs/data.ocr  
d:\oracle\data.ocr  
+bkupdg:/aimev1/ocrbackup/data.ocr.260.739756845
```

- You must shut down Oracle Clusterware before running this command.

Example

To import the contents a file back into OCR:

```
# ocrconfig -import d:\tmp\a
```

ocrconfig -manualbackup

Backs up OCR on demand to a specific location.

Prerequisites

Determine the location of the on-demand backup using [ocrconfig -backuploc](#).

Syntax

```
ocrconfig [-local] -manualbackup
```

Usage Notes

- You must run this command as `root`.
- Use the `-local` option to perform a manual backup of OLR.

ocrconfig -overwrite

Overwrites an OCR configuration in the OCR metadata with the current OCR configuration information that is found on the node from which you run this command.

Syntax

```
ocrconfig -overwrite
```

Usage Notes

You must run this command as `root`.

ocrconfig -repair

Repairs an OCR configuration on the node from which you run this command. Use this command to add, delete, or replace an OCR location on a node that may have been stopped while you made changes to the OCR configuration in the cluster. OCR locations that you add must exist, have sufficient permissions, and must be mounted before you can add them.

Prerequisites

- Oracle High Availability Services must be started to successfully complete the repair.
- The Cluster Ready Services daemon must be stopped before running `ocrconfig -repair`.

Syntax

```
ocrconfig -repair -add +diskgroup_name | -delete +diskgroup_name | -replace  
+current_diskgroup_name -replacement +new_diskgroup_name
```

Usage Notes

- You must run this command as `root`.
- The `+diskgroup_name` variable is the name of an Oracle ASM disk group. For example: `+newdg/mycluster/OCRFILE/registry.255.842574125`

If you specify an Oracle ASM disk group, then the name of the disk group must be preceded by a plus sign (+).

To obtain the name of an Oracle ASM disk group, run the `ocrcheck -config -details` command on a node where the OCR configuration *does not* need repair.

 **See Also:**

- *Oracle Grid Infrastructure Installation and Upgrade Guide* for information about creating OCRs and setting correct permissions
- *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle ASM disk group management
- "[Using the OCRCHECK Utility](#)" for information about the `ocrcheck -config -details` command

- You can only use one option with `ocrconfig -repair` at a time.
- Running this command only modifies the local configuration and it and only affects the current node.

Example

To repair an OCR configuration:

```
# ocrconfig -repair -delete +olddg
```

ocrconfig -replace

Replaces an OCR location to an Oracle Automatic Storage Management (Oracle ASM) disk group.

Prerequisites

- OCR locations that you add must exist, have sufficient permissions, and must be mounted before you can add them.
- You must have at least two OCR devices to use this command. If you do not have at least two OCR devices, then you must run the `ocrconfig -add` command to add a new OCR device followed by the `ocrconfig -delete` command to delete the OCR device you want to replace.

Syntax

```
ocrconfig -replace +current_diskgroup_name -replacement +new_diskgroup_name
```

Usage Notes

- You must run this command as `root`.
- The `+diskgroup_name` variable is the name of an Oracle ASM disk group, preceded by a plus sign (+). For example: `+newdg`. The disk group must exist and be mounted.

 **See Also:**

- *Oracle Grid Infrastructure Installation and Upgrade Guide* for information about creating OCRs and setting correct permissions
- *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle ASM disk group management

Example

To replace an OCR device or file:

```
# ocrconfig -replace +olddg -replacement +newdg
```

ocrconfig -restore

Restores OCR from an automatically created OCR backup file.

Prerequisites

- Before running this command, ensure that the original OCR or OLR files exist. If the original file does not exist, then you must create an empty file.

 **Note:**

You cannot use a file generated by the `ocrconfig -export` command to restore OCR. The file must be a manual or automatic backup.

- If OCR is located on an Oracle ASM disk group, then ensure that the disk group exists and is mounted.

Syntax

```
ocrconfig [-local] -restore file_name
```

Usage Notes

- You must run this command as `root`.
- If the OCR backup file is located in an Oracle ASM disk group, then ensure that the disk group exists and is mounted.
- The name of an Oracle ASM disk group must be preceded by a plus sign (+).
- Use the `-local` option to restore a backup of OLR.
- Example file names are:
 - `/oradbocfs/crs/BACKUP00.ocr`
 - `d:\oracle\BACKUP01.ocr`
 - `+bkupdg:/aimev1/ocrbackup/day.ocr.260.739756845`

Example

To restore OCR from a file:

```
# ocrconfig -restore /oradbcfs/crs/BACKUP00.ocr
```

ocrconfig -showbackup

Displays all available backups.

Syntax

```
ocrconfig [-local] -showbackup [auto | manual]
```

Usage Notes

- Use the `-local` option to show manual OLR backup information. The `-local` flag functions only with the `manual` option.
- You can optionally specify `auto` or `manual` to display information about only automatic backups or only manual backups, respectively:
 - `auto`: Displays information about automatic backups that Oracle Clusterware created in the past 4 hours, 8 hours, 12 hours, and in the last day and week.
 - `manual`: Displays information for, at most, the last five manual backups that you invoke using the `ocrconfig -manualbackup` command.

Example

This command displays manual backup information for OLR, similar to the following:

```
$ ocrconfig -local -showbackup manual  
  
cc40524 2017/05/08 15:06:57 /mjk_has5/oracle/log/srvm/  
backup_20170508_150657.olr 0
```

ocrconfig -showbackuploc

Displays the backup location of the OCR and OLR backup files.

Syntax

```
ocrconfig [-local] -showbackuploc
```

Usage Notes

- You must run this command as `root`
- Use the `-local` option to display the OLR backup location

Example

This command returns output similar to the following:

```
# ocrconfig -showbackuploc
```

The Oracle Cluster Registry backup location is [+bkupdg]

```
# ocrconfig -local -showbackuploc
```

The Oracle Local Registry backup location is [/gridhome/oracle/log/srvm]

ocrconfig -upgrade

Upgrades OCR from a previous version.

Syntax

```
ocrconfig -upgrade
```

Usage Notes

This command has no options.

Troubleshooting Oracle Cluster Registry and Diagnostic Output

This section describes various methods for troubleshooting problems with OCR, and obtaining diagnostic information from the utilities used to manage OCR. You can use these utilities to troubleshoot OLR.

This section contains the following topics:

- [Troubleshooting Oracle Cluster Registry](#)
- [Using the OCRCHECK Utility](#)
- [Using the OCRDUMP Utility to View Oracle Cluster Registry Content](#)

Troubleshooting Oracle Cluster Registry

[Table J-1](#) describes common OCR problems with corresponding resolution suggestions.

Table J-1 Common Oracle Cluster Registry Problems and Solutions

Problem	Solution
Not currently using OCR mirroring and would like to enable it.	Run the ocrconfig command with the -replace option.

Table J-1 (Cont.) Common Oracle Cluster Registry Problems and Solutions

Problem	Solution
OCR failed and you must replace it. Error messages in Oracle Enterprise Manager or OCR log file.	Run the <code>ocrconfig</code> command with the <code>-replace</code> option.
OCR has a misconfiguration.	Run the <code>ocrconfig</code> command with the <code>-repair</code> option as described.
You are experiencing a severe performance effect from OCR processing or you want to remove OCR for other reasons.	Run the <code>ocrconfig</code> command with the <code>-replace</code> option as described.
OCR has failed and before you can fix it, the node must be rebooted with only one OCR.	Run the <code>ocrconfig</code> with the <code>-repair</code> option to remove the bad OCR location. Oracle Clusterware cannot start if it cannot find all OCRs defined.

Using the OCRCHECK Utility

The OCRCHECK utility displays the version of the OCR's block format, total space available and used space, OCRID, and the OCR locations that you have configured.

OCRCHECK performs a block-by-block `checksum` operation for all of the blocks in all of the OCRs that you have configured. It also returns an individual status for each file and a result for the overall OCR integrity check.

Note:

Oracle supports using the `ocrcheck` command when, at a minimum, the Oracle Cluster Ready Services stack is OFFLINE on all nodes in the cluster because the command will run even if the stack is ONLINE but it can falsely indicate that the OCR is corrupt if the check happens while an update to OCR is underway.

See Also:

"[The Oracle Clusterware Technology Stack](#)" for more information about the Oracle Cluster Ready Services stack

Syntax

```
ocrcheck [-local] [-config | -backupfile <backup_file_name>] [-details] [-help]
```

You can run the `ocrcheck -help` command to display usage information about this utility.

The `-details` option displays detailed OCR configuration information, including names for Oracle ASM disk groups.

Examples

The following examples show sample OCRCHECK command output and command usage.

Example J-1 OCRCHECK Utility Sample Output

```
# ocrcheck

Status of Oracle Cluster Registry is as follows :
  Version                :          4
  Total space (kbytes)   :    409568
  Used space (kbytes)    :     17012
  Available space (kbytes) :    392556
  ID                     :    467263828
  Device/File Name       : +ocrdg1
                        : Device/File integrity check succeeded
  Device/File Name       : +ocrdg2
                        : Device/File integrity check succeeded
                        : Device/File not configured
                        : Device/File not configured
                        : Device/File not configured

Cluster registry integrity check succeeded
Logical corruption check succeeded
```

Note:

The logical corruption check is only performed if you run the `ocrcheck` command as `root`.

[Example J-2](#) shows using `ocrcheck -local -config` command to obtain OLR information.

Note:

Oracle supports using the `ocrcheck -local` command when both the Oracle Cluster Ready Services and Oracle High Availability Services stacks are OFFLINE on the local node.

See Also:

["The Oracle Clusterware Technology Stack"](#) for more information about the Oracle Cluster Ready Services and Oracle High Availability Services stacks

Example J-2 Using OCRCHECK to Obtain OLR Information

```
$ ocrcheck -local -config

Oracle Local Registry configuration is :
    Device/File Name      : /scratch/oracle/crshome/cdata/
rwsbg08.olr
```

[Example J-3](#) shows using the `ocrcheck -config -details` command to obtain the location of OCR in an Oracle ASM disk group.

Example J-3 Using OCRCHECK to Obtain OCR Information

```
$ ocrcheck -config -details

Oracle Cluster Registry configuration is :
    Device/File Name      : +ocrvd11/mjk_1/OCRFILE/registry.
255.842574113
    Device/File Name      : +ocrdv2/mjk_1/OCRFILE/registry.
255.842574125
```

[Example J-4](#) shows using the `ocrcheck -details` command to obtain a greater level of detail for the OCR configuration.

Example J-4 Using OCRCHECK to Obtain More OCR Information

```
$ ocrcheck -details

Status of Oracle Cluster Registry is as follows :
    Version              :          4
    Total space (kbytes) :      409568
    Used space (kbytes)  :       17012
    Available space (kbytes) :    392556
    ID                   :   467263828
    Device/File Name     : +OCRVD11/mycluster/OCRFILE/registry.
255.842574113
                                Device/File integrity check succeeded
    Device/File Name     : +OCRVD2/mycluster/OCRFILE/registry.
255.842574125
                                Device/File integrity check succeeded
                                Device/File not configured
                                Device/File not configured
                                Device/File not configured
    Cluster registry integrity check succeeded
    Logical corruption check succeeded
```

[Example J-5](#) shows using the `ocrcheck -details` command to obtain backup file information.

Example J-5 Using OCRCHECK to Obtain Backup File Information

```
% ocrcheck -backupfile +bkupdg:/sbezawadhtest/OCRBACKUP/backup00.ocr.
258.919047805
```

```
Status of backup file is as follows:  
Version : 4  
Used space (kbytes) : 17012  
Device/File Name : +bkupdg:/sbezawadhtest/OCRBACKUP/backup00.ocr.  
258.919047805  
Device/File integrity check succeeded  
Backup file integrity check succeeded  
Logical corruption check succeeded  
Consistency check of entities managed by Cluster Ready Services succeeded.
```

OCRCHECK creates a log file in the `ORACLE_BASE/diag/crs/host_name/crs` directory. To change the log level, edit the `ORACLE_BASE/crsdata/host_name/crsdiag/ocrcheck.ini` file. The `ORACLE_BASE`, in this case, is for the Oracle Grid Infrastructure software home.

Using the OCRDUMP Utility to View Oracle Cluster Registry Content

The OCRDUMP utility enables you to view OCR and OLR contents by writing the content to a file or `stdout` in a readable format.

This section explains how to use the OCRDUMP utility to view OCR and Oracle Local Registry (OLR) content for troubleshooting.

You can use several options for OCRDUMP. For example, you can limit the output to a key and its descendents. You can also write the contents to an XML file that you can view using a browser. OCRDUMP writes the OCR keys as ASCII strings and values in a data type format. OCRDUMP retrieves header information based on a best effort basis.

OCRDUMP creates a log file in `ORACLE_BASE/diag/crs/host_name/crs`. To change the log level, edit the `ORACLE_BASE/crsdata/host_name/crsdiag/ocrlog.ini` file. The `ORACLE_BASE`, in this case, is for the Oracle Grid Infrastructure software home.

To change the logging component, edit the entry containing the `comploglvl=` entry. For example, to change the log level of the `OCRAPI` component to 3 and to change the log level of the `OCRRAW` component to 5, make the following entry in the `ocrlog.ini` file:

```
comploglvl="OCRAPI:3;OCRRAW:5"
```

Note:

Make sure that you have file creation privileges in the `Grid_home` directory before using the OCRDUMP utility.

This section includes the following topics:

- [OCRDUMP Utility Syntax and Options](#)
- [OCRDUMP Utility Examples](#)
- [Sample OCRDUMP Utility Output](#)

OCRDUMP Utility Syntax and Options

This section describes the OCRDUMP utility command syntax and usage. Run the `ocrdump` command with the following syntax where *file_name* is the name of a target file to which you want Oracle Database to write the Oracle Cluster Registry output and where *key_name* is the name of a key from which you want Oracle Database to write Oracle Cluster Registry subtree content:

```
$ ocrdump [file_name | -stdout] [-local] [-backupfile backup_file_name
[-keyname key_name] [-xml] [-noheader]
]
```

[Table J-2](#) describes the OCRDUMP utility options and option descriptions.

Table J-2 OCRDUMP Options and Option Descriptions

Options	Description
<i>file_name</i>	The name of a file to which you want OCRDUMP to write output. By default, OCRDUMP writes output to a predefined output file named OCRDUMPFIL. The <i>file_name</i> option redirects OCRDUMP output to a file that you specify.
-stdout	Use this option to redirect the OCRDUMP output to the text terminal that initiated the program. If you do not redirect the output, OCRDUMP writes output to a predefined output file named OCRDUMPFIL.
-local	Use this option to dump the contents of OLR.
-backupfile	Use this option to view the contents of an OCR backup file. Use the <code>-local</code> option with this option to view the contents of an OLR backup file.
<i>backup_file_name</i>	The name of the backup file with the content you want to view. You can query the backups using the <code>ocrconfig -showbackup</code> command. If the backup file is located in an Oracle ASM disk group, then ensure that the disk group exists and is mounted.
-keyname <i>key_name</i>	The name of an Oracle Cluster Registry key whose subtree is to be dumped.
-xml	Use this option to write the output in XML format.
-noheader	Does not print the time at which you ran the command and when the Oracle Cluster Registry configuration occurred.

OCRDUMP Utility Examples

The following `ocrdump` utility examples extract various types of OCR information and write it to various targets:

```
ocrdump
```

Writes OCR content to a file called `OCRDUMPFIL` in the current directory.

```
ocrdump MYFILE
```

Writes OCR content to a file called `MYFILE` in the current directory.

```
ocrdump -stdout -keyname SYSTEM
```

Displays OCR content from the subtree of the key `SYSTEM` in the terminal window.

```
ocrdump -stdout -xml
```

Displays OCR content in the terminal window in XML format.

```
ocrdump -stdout -backupfile Grid_home/cdata/cluster_name/file_name
```

Displays the content of the in the `Grid_home/cdata/cluster_name/file_name` directory. You must run this command as `root` to be able to view all of the keys. Be sure to name the file appropriately so that it can be recognized by anyone as an OCR backup file, such as `BACKUPOO.ocr`.

Sample OCRDUMP Utility Output

The following OCRDUMP examples show the `KEYNAME`, `VALUE TYPE`, `VALUE`, permission set (`user`, `group`, `world`) and access rights for two sample runs of the `ocrdump` command. The following shows the output for the `SYSTEM.language` key that has a text value of `AMERICAN_AMERICA.WE8ASCII37`.

```
[SYSTEM.language]
ORATEXT : AMERICAN_AMERICA.WE8ASCII37
SECURITY : {USER_PERMISSION : PROC_ALL_ACCESS, GROUP_PERMISSION : PROC_READ,
OTHER_PERMISSION : PROC_READ, USER_NAME : user, GROUP_NAME : group}
```

The following shows the output for the `SYSTEM.version` key that has integer value of 3:

```
[SYSTEM.version]
UB4 (10) : 3
SECURITY : {USER_PERMISSION : PROC_ALL_ACCESS, GROUP_PERMISSION : PROC_READ,
OTHER_PERMISSION : PROC_READ, USER_NAME : user, GROUP_NAME : group}
```


K

REST APIs for Oracle Clusterware

This appendix describes the REST APIs for Oracle Clusterware.

The following topics are discussed:

- [About REST APIs for Oracle Clusterware](#)

About REST APIs for Oracle Clusterware

The REST application programming interfaces (APIs) for Oracle Clusterware enable you to remotely execute commands on your cluster, whether in the Oracle Cloud, at remote physical locations, or locally provisioned. With the remote execution of REST interface commands, you are able to retrieve information about that execution, including output, error codes, and execution duration. The REST interface provides secure support for Oracle Clusterware SRVCTL and CRSCTL command line interfaces (CLIs).

Note:

The REST APIs for Oracle Clusterware are not supported in Windows environments.

Getting Started

REST APIs for Oracle Clusterware requires that the CDP Cross Cluster Domain Protocol (CDP) daemon is running on all of the SCAN VIPs of the cluster. REST endpoints exist for you to execute CLIs as asynchronous jobs. You can access the output of a command from a REST endpoint at later point of time, and get return codes and execution time.

To make the REST APIs secure by default, all requests that are not from IP addresses of nodes in the cluster are rejected. The CDP daemon monitors the nodes in the cluster and rejects any request to the CLI endpoint that do not match one of the IPs. To support the ability to make requests from outside the cluster, you can run the `srvctl modify cdp` command to provide a list of IPs or networks in CIDR format.

To change the admin password, run the following commands.

```
$ srvctl stop cdp
$ echo "myAdminPasswd" > /tmp/a
$ srvctl modify cdp -passfile_admin /tmp/a
```

To enable connections from outside the cluster, run the following commands.

```
$ srvctl start cdp
$ srvctl modify cdp -allow "ip/network1,ip/network2,.."
```

You can view the configuration information with the following command:

```
$ srvctl config cdp
```

The password is not visible when you display the configuration information.

See Also:

- [srvctl modify cdp](#)
- [srvctl config cdp](#)

REST Endpoints

- Delete a Specified Job
Method: Delete
Path: `/grid/cmd/v1/cmd/jobs/jobId`
- Execute a CLI
Method: Post
Path: `/grid/cmd/v1/cmd/exec`
- Return a List of the Job Resources
Method: Get
Path: `/grid/cmd/v1/cmd/jobs`
- Return the Common Resource
Method: Get
Path: `/grid/cmd/v1/cmd`
- Return the Status of the Current Job
Method: Get
Path: `/grid/cmd/v1/cmd/jobs/jobId`

Examples

The following are some examples of the use of REST APIs for Oracle Clusterware.

Example K-1 Get the list of all homes

```
curl -k -X GET https://scan-name:port/grid/cmd/v1/cmd/ --user  
admin:myAdminPasswd
```

Example K-2 Create a job and monitor the status #1

```
curl -k -X POST \  
https://scan-name:port/grid/cmd/v1/cmd/exec \  
'-H "accept: text/plain,text/javascript,application/json" \  
'-H "content-type: application/vnd.oracle.resource
```

```
+json;type=singular"' \
  --user admin:myAdminPasswd \
  -d {"command" : ["crsctl", "stat", "res", "-t"],
"runAsUser": "osUser", "userPassword": "osPasswd"}
```

```
curl -k -X GET https://scan-name:port/grid/cmd/v1/cmd/jobs/myJobId --user
admin:myAdminPasswd
```

Example K-3 Create a job and monitor the status #2

```
curl -k -X POST \
  https://scan-name:port/grid/cmd/v1/cmd/exec \
  -H "accept: text/plain,text/javascript,application/json" \
  -H "content-type: application/vnd.oracle.resource
+json;type=singular"' \
  --user admin:myAdminPasswd \
  -d {"command" : [ "srvctl","config", "cdp" ], "runAsUser" :
"osUser", "userPassword" : "osPasswd"}
```

```
curl -k -X GET https://scan-name:port/grid/cmd/v1/cmd/jobs/myJobId --user
admin:myAdminPasswd
```

Example K-4 Create a job and monitor the status #3

```
curl -k -X POST \
  https://scan-name:port/grid/cmd/v1/cmd/exec \
  -H "accept: text/plain,text/javascript,application/json" \
  -H "content-type: application/vnd.oracle.resource
+json;type=singular"' \
  --user admin:myAdminPasswd \
  -d {"command" : ["srvctl", "config", "scan"], "runAsUser": "osUser",
"userPassword": "osPasswd", "jobExpiry": 300}
```

```
curl -k -X GET https://scan-name:port/grid/cmd/v1/cmd/jobs/myJobId --user
admin:myAdminPasswd
```

Example K-5 Monitor the status of all jobs

```
curl -k -X GET \
  https://scan-name:port/grid/cmd/v1/cmd/jobs/ --user admin:myAdminPasswd
```

Example K-6 Delete a job

```
curl -k -X DELETE \
  https://scan-name:port/grid/cmd/v1/cmd/jobs/myJobId --user
admin:myAdminPasswd
```

Example K-7 Verify that a deleted job has been removed

```
curl -k -X GET \  
    https://scan-name:port/grid/cmd/v1/cmd/jobs/myJobId --user  
admin:myAdminPasswd
```

L

Troubleshooting Oracle Clusterware

This appendix introduces monitoring the Oracle Clusterware environment and explains how you can enable dynamic debugging to troubleshoot Oracle Clusterware processing, and enable debugging and tracing for specific components and specific Oracle Clusterware resources to focus your troubleshooting efforts.

This appendix includes the following topics:

- [Troubleshooting an Incompatible Fleet Patching and Provisioning Client Resource](#)
- [Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures](#)
- [Oracle Clusterware Diagnostic and Alert Log Data](#)
- [Diagnostics Collection Script](#)
- [Storage Split in Oracle Extended Clusters](#)
- [Rolling Upgrade and Driver Installation Issues](#)
- [Testing Zone Delegation](#)
- [Oracle Clusterware Alerts](#)

Troubleshooting an Incompatible Fleet Patching and Provisioning Client Resource

If you manually upgrade an Oracle Clusterware 12c (12.1) target that is registered as a Fleet Patching and Provisioning target to a later version, then the result will be an incompatible Fleet Patching and Provisioning Client resource.

To register a Fleet Patching and Provisioning Client resource with Fleet Patching and Provisioning, you must perform the following steps on the Fleet Patching and Provisioning Server:

Note:

These same steps apply to upgrading an Oracle Clusterware 12c (12.2) Fleet Patching and Provisioning Client cluster to a later version that results in connectivity issues.

1. Stop the Fleet Patching and Provisioning Client resource on the target if it is running.
2. Run the following commands on the Fleet Patching and Provisioning Server:

Update the Fleet Patching and Provisioning Client credential on the Fleet Patching and Provisioning Server:

```
$ rhpctl modify client -client client_name -password
```

Export data from the repository to the Fleet Patching and Provisioning Client data file:

```
$ rhpctl export client -client client_name -clientdata file_name.xml
```

3. Copy the `xml` file you created in the previous step to a node in the Fleet Patching and Provisioning Client cluster, and run the following command to update the client credential:

```
$ srvctl modify rhpclient -clientdata file_name.xml
```

4. Start the Fleet Patching and Provisioning Client:

```
$ srvctl start rhpclient
```

5. Verify that the Fleet Patching and Provisioning Client started successfully:

```
$ srvctl status rhpclient
```

6. To verify that the Fleet Patching and Provisioning Client is properly registered with the Fleet Patching and Provisioning Server, run the following command and search for the Fleet Patching and Provisioning Client-registered port reported in the command output:

```
$ rhpctl query client -client client_cluster
```

Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

If an Oracle Clusterware-managed resource fails, then Oracle Clusterware logs messages about the failure in the **cluster resource activity log** located in the Grid Infrastructure Management Repository. Failures can occur as a result of a problem with a resource, a hosting node, or the network. The cluster resource activity log provides a unified view of the cause of resource failure.

Writes to the cluster resource activity log are tagged with an activity ID and any related data gets the same parent activity ID, and is nested under the parent data. For example, if Oracle Clusterware is running and you run the `crsctl stop clusterware -all` command, then all activities get activity IDs, and related activities are tagged with the same parent activity ID. On each node, the command creates sub-IDs under the parent IDs, and tags each of the respective activities with their corresponding activity ID. Further, each resource on the individual nodes creates sub-IDs based on the parent ID, creating a hierarchy of activity IDs. The hierarchy of activity IDs enables you to analyze the data to find specific activities.

For example, you may have many resources with complicated dependencies among each other, and with a database service. On Friday, you see that all of the resources are running on one node but when you return on Monday, every resource is on a different node, and you want to know why. Using the `crsctl query calog` command, you can query the cluster resource activity log for all activities involving those resources and the database service. The output provides a complete flow and you can query each sub-ID within the parent service failover ID, and see, specifically, what happened and why.

You can query any number of fields in the cluster resource activity log using filters. For example, you can query all the activities written by specific operating system users such as `root`. The output produced by the `crsctl query calog` command can be displayed in either a tabular format or in XML format.

The cluster resource activity log is an adjunct to current Oracle Clusterware logging and alert log messages.

 **Note:**

Oracle Clusterware does not write messages that contain security-related information, such as log-in credentials, to the cluster activity log.

Use the following commands to manage and view the contents of the cluster resource activity log:

crsctl query calog

Query the cluster resource activity logs matching specific criteria.

Syntax

```
crsctl query calog [-aftertime "timestamp"] [-beforetime "timestamp"]  
  [-duration "time_interval" | -follow] [-filter "filter_expression"]  
  [-fullfmt | -xmlfmt]
```

Parameters

Table L-1 crsctl query calog Command Parameters

Parameter	Description
-aftertime "timestamp"	<p>Displays the activities logged after a specific time.</p> <p>Specify the timestamp in the YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM] or YYYY-MM-DD or HH24:MI:SS[.FF] [TZH:TZM] format.</p> <p>TZH and TZM stands for time zone hour and minute, and FF stands for microseconds.</p> <p>If you specify [TZH:TZM], then the crsctl command assumes UTC as time zone. If you do not specify [TZH:TZM], then the crsctl command assumes the local time zone of the cluster node from where the crsctl command is run.</p> <p>Use this parameter with -beforetime to query the activities logged at a specific time interval.</p>
-beforetime "timestamp"	<p>Displays the activities logged before a specific time.</p> <p>Specify the timestamp in the YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM] or YYYY-MM-DD or HH24:MI:SS[.FF] [TZH:TZM] format.</p> <p>TZH and TZM stands for time zone hour and minute, and FF stands for microseconds.</p> <p>If you specify [TZH:TZM], then the crsctl command assumes UTC as time zone. If you do not specify [TZH:TZM], then the crsctl command assumes the local time zone of the cluster node from where the crsctl command is run.</p> <p>Use this parameter with -aftertime to query the activities logged at a specific time interval.</p>
-duration "time_interval" - follow	<p>Use -duration to specify a time interval that you want to query when you use the -aftertime parameter.</p> <p>Specify the timestamp in the DD HH:MM:SS format.</p> <p>Use -follow to display a continuous stream of activities as they occur.</p>
-filter "filter_expression"	<p>Query any number of fields in the cluster resource activity log using the -filter parameter.</p> <p>To specify multiple filters, use a comma-delimited list of filter expressions surrounded by double quotation marks (" ").</p>
-fullfmt -xmlfmt	<p>To display cluster resource activity log data, choose full or XML format.</p>

Cluster Resource Activity Log Fields

Query any number of fields in the cluster resource activity log using the -filter parameter.

Table L-2 Cluster Resource Activity Log Fields

Field	Description	Use Case
timestamp	The time when the cluster resource activities were logged.	Use this filter to query all the activities logged at a specific time. This is an alternative to - aftertime, -beforetime, and -duration command parameters.
writer_process_id	The ID of the process that is writing to the cluster resource activity log.	Query only the activities spawned by a specific process.
writer_process_name	The name of the process that is writing to the cluster resource activity log.	When you query a specific process, CR\$CTL returns all the activities for a specific process.
writer_user	The name of the user who is writing to the cluster resource activity log.	Query all the activities written by a specific user.
writer_group	The name of the group to which a user belongs who is writing to the cluster resource activity log.	Query all the activities written by users belonging to a specific user group.
writer_hostname	The name of the host on which the cluster resource activity log is written.	Query all the activities written by a specific host.
writer_clustername	The name of the cluster on which the cluster resource activity log is written.	Query all the activities written by a specific cluster.
nls_product	The product of the NLS message, for example, CRS, ORA, or srvm.	Query all the activities that have a specific product name.
nls_facility	The facility of the NLS message, for example, CRS or PROC.	Query all the activities that have a specific facility name.
nls_id	The ID of the NLS message, for example 42008.	Query all the activities that have a specific message ID.
nls_field_count	The number of fields in the NLS message.	Query all the activities that correspond to NLS messages with more than, less than, or equal to nls_field_count command parameters.
nls_field1	The first field of the NLS message.	Query all the activities that match the first parameter of an NLS message.
nls_field1_type	The type of the first field in the NLS message.	Query all the activities that match a specific type of the first parameter of an NLS message.
nls_format	The format of the NLS message, for example, Resource '%S' has been modified.	Query all the activities that match a specific format of an NLS message.

Table L-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
nls_message	The entire NLS message that was written to the cluster resource activity log, for example, Resource 'ora.cvu' has been modified.	Query all the activities that match a specific NLS message.
actid	The unique activity ID of every cluster activity log.	Query all the activities that match a specific ID. Also, specify only partial actid and list all activities where the actid is a subset of the activity ID.
is_planned	Confirms if the activity is planned or not. For example, if a user issues the command <code>crsctl stop crs</code> on a node, then the stack stops and resources bounce. Running the <code>crsctl stop crs</code> command generates activities and logged in the calog. Since this is a planned action, the <code>is_planned</code> field is set to true (1). Otherwise, the <code>is_planned</code> field is set to false (0).	Query all the planned or unplanned activities.
onbehalfof_user	The name of the user on behalf of whom the cluster activity log is written.	Query all the activities written on behalf of a specific user.
entity_isoraentity	Confirms if the entity for which the calog activities are being logged is an oracle entity or not. If a resource, such as <code>ora.***</code> , is started or stopped, for example, then all those activities are logged in the cluster resource activity log. Since <code>ora.***</code> is an Oracle entity, the <code>entity_isoraentity</code> field is set to true (1). Otherwise the <code>entity_isoraentity</code> field is set to false (0).	Query all the activities logged by Oracle or non-Oracle entities.
entity_type	The type of the entity, such as <code>server</code> , for which the cluster activity log is written.	Query all the activities that match a specific entity.

Table L-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
entity_name	The name of the entity, for example, <i>foo</i> for which the cluster activity log is written.	Query all the cluster activities that match a specific entity name.
entity_hostname	The name of the host, for example, <i>node1</i> , associated with the entity for which the cluster activity log is written.	Query all the cluster activities that match a specific host name.
entity_clustername	The name of the cluster, for example, <i>cluster1</i> associated with the entity for which the cluster activity log is written.	Query all the cluster activities that match a specific cluster name.

Usage Notes

Combine simple filters into expressions called expression filters using Boolean operators.

Enclose timestamps and time intervals in double quotation marks ("").

Enclose the filter expressions in double quotation marks ("").

Enclose the values that contain parentheses or spaces in single quotation marks (').

If no matching records are found, then the Oracle Clusterware Control (CRSCTL) utility displays the following message:

```
CRS-40002: No activities match the query.
```

Examples

Examples of filters include:

- "writer_user==root": Limits the display to only root user.
- "customer_data=='GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~'": Limits the display to *customer_data* that has the specified value *GEN_RESTART@SERVERNAME(node1)=StartCompleted~*.

To query all the resource activities and display the output in full format:

```
$ crsctl query calog -fullfmt
```

```
----ACTIVITY START----
timestamp           : 2016-09-27 17:55:43.152000
writer_process_id   : 6538
writer_process_name : crsd.bin
writer_user         : root
writer_group        : root
writer_hostname     : node1
writer_clustername  : cluster1-mbl
customer_data       : CHECK_RESULTS=-408040060~
nls_product         : CRS
nls_facility        : CRS
```

```

nls_id           : 2938
nls_field_count  : 1
nls_field1       : ora.cvu
nls_field1_type  : 25
nls_field1_len   : 0
nls_format       : Resource '%s' has been modified.
nls_message      : Resource 'ora.cvu' has been modified.
actid           : 14732093665106538/1816699/1
is_planned       : 1
onbehalfof_user  : grid
onbehalfof_hostname : node1
entity_isoraentity : 1
entity_type      : resource
entity_name      : ora.cvu
entity_hostname  : node1
entity_clustername : cluster1-mbl
----ACTIVITY END----

```

To query all the resource activities and display the output in XML format:

```
$ crsctl query calog -xmlfmt
```

```

<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-27 17:55:43.152000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustername>cluster1-mbl</writer_clustername>
    <customer_data>CHECK_RESULTS=-408040060~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
    <nls_field1_len>0</nls_field1_len>
    <nls_format>Resource '%s' has been modified.</nls_format>
    <nls_message>Resource 'ora.cvu' has been modified.</nls_message>
    <actid>14732093665106538/1816699/1</actid>
    <is_planned>1</is_planned>
    <onbehalfof_user>grid</onbehalfof_user>
    <onbehalfof_hostname>node1</onbehalfof_hostname>
    <entity_isoraentity>1</entity_isoraentity>
    <entity_type>resource</entity_type>
    <entity_name>ora.cvu</entity_name>
    <entity_hostname>node1</entity_hostname>
    <entity_clustername>cluster1-mbl</entity_clustername>
  </activity>
</activities>

```

To query resource activities for a two-hour interval after a specific time and display the output in XML format:

```
$ crsctl query calog -aftertime "2016-09-28 17:55:43" -duration "0
02:00:00" -xmlfmt
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-28 17:55:45.992000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustername>cluster1-mbl</writer_clustername>
    <customer_data>CHECK_RESULTS=1718139884~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
    <nls_field1_len>0</nls_field1_len>
    <nls_format>Resource '%s' has been modified.</nls_format>
    <nls_message>Resource 'ora.cvu' has been modified.</nls_message>
    <actid>14732093665106538/1942009/1</actid>
    <is_planned>1</is_planned>
    <onbehalfof_user>grid</onbehalfof_user>
    <onbehalfof_hostname>node1</onbehalfof_hostname>
    <entity_isoraentity>1</entity_isoraentity>
    <entity_type>resource</entity_type>
    <entity_name>ora.cvu</entity_name>
    <entity_hostname>node1</entity_hostname>
    <entity_clustername>cluster1-mbl</entity_clustername>
  </activity>
</activities>
```

To query resource activities at a specific time:

```
$ crsctl query calog -filter "timestamp=='2016-09-28 17:55:45.992000'"
2016-09-28 17:55:45.992000 : Resource 'ora.cvu' has been modified. :
14732093665106538/1942009/1 :
```

To query resource activities using filters `writer_user` and `customer_data`:

```
$ crsctl query calog -filter "writer_user==root AND customer_data==
'GEN_RESTART@SERVERNAME(node1)=StartCompleted~'" -fullfmt
```

Or

```
$ crsctl query calog -filter "(writer_user==root) AND (customer_data==
'GEN_RESTART@SERVERNAME(nodel)=StartCompleted~')" -fullfmt

----ACTIVITY START----
timestamp           : 2016-09-15 17:42:57.517000
writer_process_id   : 6538
writer_process_name : crsd.bin
writer_user         : root
writer_group        : root
writer_hostname     : nodel
writer_clustername  : cluster1-mb1
customer_data       : GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~
nls_product         : CRS
nls_facility        : CRS
nls_id              : 2938
nls_field_count     : 1
nls_field1         : ora.testdb.db
nls_field1_type     : 25
nls_field1_len      : 0
nls_format          : Resource '%s' has been modified.
nls_message         : Resource 'ora.devdb.db' has been modified.
actid               : 14732093665106538/659678/1
is_planned          : 1
onbehalfof_user     : oracle
onbehalfof_hostname : nodel
entity_isoraentity  : 1
entity_type         : resource
entity_name         : ora.testdb.db
entity_hostname     : nodel
entity_clustername  : cluster1-mb1
----ACTIVITY END----
```

To query all the calogs that were generated after UTC+08:00 time "2016-11-15 22:53:08":

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08+08:00"
```

To query all the calogs that were generated after UTC-08:00 time "2016-11-15 22:53:08":

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08-08:00"
```

To query all the calogs by specifying the timestamp with microseconds:

```
$ crsctl query calog -aftertime "2016-11-16 01:07:53.063000"
```

```
2016-11-16 01:07:53.558000 : Resource 'ora.cvu' has been modified. :
14792791129816600/2580/7 :
```

```
2016-11-16 01:07:53.562000 : Clean of 'ora.cvu' on 'rwsam02' succeeded :  
14792791129816600/2580/8 :
```

crsctl get calog maxsize

To store Oracle Clusterware-managed resource activity information, query the maximum space allotted to the cluster resource activity log.

Syntax

```
crsctl get calog maxsize
```

Parameters

The `crsctl get calog maxsize` command has no parameters.

Example

The following example returns the maximum space allotted to the cluster resource activity log to store activities:

```
$ crsctl get calog maxsize
```

```
CRS-6760: The maximum size of the Oracle cluster activity log is 1024 MB.
```

crsctl get calog retentiontime

Query the retention time of the cluster resource activity log.

Syntax

```
crsctl get calog retentiontime
```

Parameters

The `crsctl get calog retentiontime` command has no parameters.

Examples

The following example returns the retention time of the cluster activity log, in number of hours:

```
$ crsctl get calog retentiontime
```

```
CRS-6781: The retention time of the cluster activity log is 73 hours.
```

crsctl set calog maxsize

Configure the maximum amount of space allotted to store Oracle Clusterware-managed resource activity information.

Syntax

```
crsctl set calog maxsize maximum_size
```

Usage Notes

Specify a value, in MB, for the maximum size of the storage space that you want to allot to the cluster resource activity log.

**Note:**

If you reduce the amount of storage space, then the contents of the storage are lost.

Example

The following example sets maximum amount of space, to store Oracle Clusterware-managed resource activity information, to 1024 MB:

```
$ crsctl set calog maxsize 1024
```

crsctl set calog retentiontime

Configure the retention time of the cluster resource activity log.

Syntax

```
crsctl set calog retentiontime hours
```

Parameters

The `crsctl set calog retentiontime` command takes a number of hours as a parameter.

Usage Notes

Specify a value, in hours, for the retention time of the cluster resource activity log.

Examples

The following example sets the retention time of the cluster resource activity log to 72 hours:

```
$ crsctl set calog retentiontime 72
```

Oracle Clusterware Diagnostic and Alert Log Data

Review this content to understand clusterware-specific aspects of how Oracle Clusterware uses ADR.

Oracle Clusterware uses Oracle Database fault diagnosability infrastructure to manage diagnostic data and its alert log. As a result, most diagnostic data resides in the Automatic Diagnostic Repository (ADR), a collection of directories and files located under a base directory that you specify during installation.

ADR Directory Structure

Oracle Clusterware ADR data is written under a root directory known as the ADR base. Because components other than ADR use this directory, it may also be referred to as the Oracle base. You specify the file system path to use as the base during Oracle Grid Infrastructure installation and can only be changed if you reinstall the Oracle Grid Infrastructure.

ADR files reside in an ADR home directory. The ADR home for Oracle Clusterware running on a given host always has this structure:

```
ORACLE_BASE/diag/crs/host_name/crs
```

In the preceding example, *ORACLE_BASE* is the Oracle base path you specified when you installed the Oracle Grid Infrastructure and *host_name* is the name of the host. On a Windows platform, this path uses backslashes (\) to separate directory names.

Under the ADR home are various directories for specific types of ADR data. The directories of greatest interest are *incident*. The *trace* directory contains all normal (non-incident) trace files written by Oracle Clusterware daemons and command-line programs as well as the simple text version of the Oracle Clusterware alert log. This organization differs significantly from versions prior to Oracle Clusterware 12c release 1 (12.1.0.2), where diagnostic log files were written under distinct directories per daemon.

To change the log level, edit the *ORACLE_BASE/crsdata/host_name/crsdiag/ocrcheck.ini* file.

Files in the Trace Directory

Starting with Oracle Clusterware 12c release 1 (12.1.0.2), diagnostic data files written by Oracle Clusterware programs are known as *trace files* and have a *.trc* file extension, and appear together in the *trace* subdirectory of the ADR home. The naming convention for these files generally uses the executable program name as the file name, possibly augmented with other data depending on the type of program.

Trace files written by Oracle Clusterware command-line programs incorporate the Operating System process ID (PID) in the trace file name to distinguish data from multiple invocations of the same command program. For example, trace data written by CRCTL uses this name structure: *crctl_PID.trc*. In this example, *PID* is the operating system process ID displayed as decimal digits.

Trace files written by Oracle Clusterware daemon programs do not include a PID in the file name, and they also are subject to a file rotation mechanism that affects naming. Rotation means that when the current daemon trace file reaches a certain size, the file is closed, renamed, and a new trace file is opened. This occurs a fixed number of times, and then the oldest trace file from the daemon is discarded, keeping the rotation set at a fixed size.

Most Oracle Clusterware daemons use a file size limit of 25 MB and a rotation set size of 10 files, thus maintaining a total of 250 MB of trace data. The current trace file for a given daemon simply uses the program name as the file name; older files in the rotation append a number to the file name. For example, the trace file currently being

written by the Oracle High Availability Services daemon (OHASD) is named `ohasd.trc`; the most recently rotated-out trace file is named `ohasd_n.trc`, where *n* is an ever-increasing decimal integer. The file with the highest *n* is actually the most recently archived trace, and the file with the lowest *n* is the oldest.

Oracle Clusterware agents are daemon programs whose trace files are subject to special naming conventions that indicate the origin of the agent (whether it was spawned by the OHASD or the Cluster Ready Services daemon (CRSD)) and the Operating System user name with which the agent runs. Thus, the name structure for agents is:

`origin_executable_user_name`

 **Note:**

The first two underscores (`_`) in the name structure are literal and are included in the trace file name. The underscore in `user_name` is not part of the file naming convention.

In the previous example, *origin* is either `ohasd` or `crsd`, *executable* is the executable program name, and *user_name* is the operating system user name. In addition, because they are daemons, agent trace files are subject to the rotation mechanism previously described, so files with an additional `_n` suffix are present after rotation occurs.

The Oracle Clusterware Alert Log

Besides trace files, the `trace` subdirectory in the Oracle Clusterware ADR home contains the simple text Oracle Clusterware alert log. It always has the name `alert.log`. The alert log is also written as an XML file in the `alert` subdirectory of the ADR home, but the text alert log is most easily read.

The alert log is the first place to look when a problem or issue arises with Oracle Clusterware. Unlike the Oracle Database instance alert log, messages in the Oracle Clusterware alert log are identified, documented, and localized (translated). Oracle Clusterware alert messages are written for most significant events or errors that occur.

 **Note:**

Messages and data written to Oracle Clusterware trace files generally are not documented and translated and are used mainly by My Oracle Support for problem diagnosis.

Incident Trace Files

Certain errors occur in Oracle Clusterware programs that will raise an ADR incident. In most cases, these errors should be reported to My Oracle Support for diagnosis. The occurrence of an incident normally produces one or more descriptive messages in the Oracle Clusterware alert log.

In addition to alert messages, incidents also cause the affected program to produce a special, separate trace file containing diagnostic data related to the error. These

incident-specific trace files are collected in the `incident` subdirectory of the ADR home rather than the `trace` subdirectory. Both the normal trace files and incident trace files are collected and submitted to Oracle when reporting the error.

 **See Also:**

Oracle Database Administrator's Guide for more information on incidents and data collection

Other Diagnostic Data

Besides ADR data, Oracle Clusterware collects or uses other data related to problem diagnosis. Starting with Oracle Clusterware 12c release 1 (12.1.0.2), this data resides under the same base path used by ADR, but in a separate directory structure with this form: `ORACLE_BASE/crsdata/host_name`. In this example, `ORACLE_BASE` is the Oracle base path you specified when you installed the Grid Infrastructure and `host_name` is the name of the host.

In this directory, on a given host, are several subdirectories. The two subdirectories of greatest interest if a problem occurs are named `core` and `output`. The `core` directory is where Oracle Clusterware daemon core files are written when the normal ADR location used for core files is not available (for example, before ADR services are initialized in a program). The `output` directory is where Oracle Clusterware daemons redirect their C standard output and standard error files. These files generally use a name structure consisting of the executable name with the characters `OUT` appended to a `.trc` file extension (like trace files). For example, the redirected standard output from the Cluster Time Synchronization Service daemon is named `octssdOUT.trc`. Typically, daemons write very little to these files, but in certain failure scenarios important data may be written there.

Related Topics

- *Oracle Database Administrator's Guide*
- *Oracle Database Utilities*

Diagnostics Collection Script

When an Oracle Clusterware error occurs, run the `diagcollection.pl` diagnostics collection script to collect diagnostic information from Oracle Clusterware into trace files. The diagnostics provide additional information so My Oracle Support can resolve problems. Run this script as `root` from the `Grid_home/bin` directory.

Syntax

Use the `diagcollection.pl` script with the following syntax:

```
diagcollection.pl {--collect [--crs | --acfs | -all] [--chmos [--incidenttime time
[--incidentduration time]]
  [--adr location [--aftertime time [--beforetime time]]]
  [--crshome path | --clean | --coreanalyze]}
```

**Note:**

The `diagcollection.pl` script arguments are all preceded by two dashes (`--`).

Parameters

Table L-3 lists and describes the parameters used with the `diagcollection.pl` script.

Table L-3 `diagcollection.pl` Script Parameters

Parameter	Description
<code>--collect</code>	<p>Use this parameter with any of the following arguments:</p> <ul style="list-style-type: none"> • <code>--crs</code>: Use this argument to collect Oracle Clusterware diagnostic information • <code>--acfs</code>: Use this argument to collect Oracle ACFS diagnostic information Note: You can only use this argument on UNIX systems. • <code>--all</code>: Use this argument to collect all diagnostic information except CHM (OS) data Note: This is the default. • <code>--chmos</code>: Use this argument to collect the following CHM diagnostic information <ul style="list-style-type: none"> <code>--incidenttime time</code>: Use this argument to collect CHM (OS) data from the specified time Note: The time format is MM/DD/YYYYHH24:MM:SS. <code>--incidentduration time</code>: Use this argument with <code>--incidenttime</code> to collect CHM (OS) data for the duration after the specified time Note: The time format is HH:MM. If you do not use <code>--incidentduration</code>, then all CHM (OS) data after the time you specify in <code>--incidenttime</code> is collected. • <code>--adr location</code>: The Automatic Diagnostic Repository Command Interpreter (ADRCI) uses this argument to specify a location in which to collect diagnostic information for ADR See Also: <i>Oracle Database Utilities</i> for more information about ADRCI • <code>--aftertime time</code>: Use this argument with the <code>--adr</code> argument to collect archives after the specified time Note: The time format is YYYYMMDDHHMISS24. • <code>--beforetime time</code>: Use this argument with the <code>--adr</code> argument to collect archives before the specified time Note: The time format is YYYYMMDDHHMISS24. • <code>--crshome path</code>: Use this argument to override the location of the Oracle Clusterware home Note: The <code>diagcollection.pl</code> script typically derives the location of the Oracle Clusterware home from the system configuration (either the <code>olr.loc</code> file or the Windows registry), so this argument is not required.

Table L-3 (Cont.) diagcollection.pl Script Parameters

Parameter	Description
<code>--clean</code>	Use this parameter to clean up the diagnostic information gathered by the <code>diagcollection.pl</code> script. Note: You cannot use this parameter with <code>--collect</code> .
<code>--coreanalyze</code>	Use this parameter to extract information from core files and store it in a text file. Note: You can only use this parameter on UNIX systems.

Storage Split in Oracle Extended Clusters

A storage split occurs when the private network between two or more disparate sites is available and online, but the storage network has failed.

When Oracle Automatic Storage Management (Oracle ASM) detects a storage split in a typical extended cluster configuration with three sites (two data sites and a quorum site), one of the data sites terminates and quarantines itself and the nodes it contains from the rest of the cluster. If Oracle ASM attempts to start on the quarantined site, then error CRS-2971 occurs.

Resolve the issue, as follows:

1. Resolve the inter-site connectivity issue that resulted in the storage split.
2. Ensure that all Oracle ASM disk groups are mounted on the site that is not quarantined, as follows:

```
SELECT group_number, name, state FROM v$asm_diskgroup_stat;
```

3. Obtain a list of online disks belonging to these disk groups by running the following command for each disk group:

```
SELECT path FROM v$asm_disk_stat WHERE group_number=group_number AND state = 'NORMAL' AND mode_status = 'ONLINE';
```

4. For each of the paths from you obtained in the previous step, ensure that the disk is accessible from the quarantined site, as follows:

```
asmcmd lsdsk -I --member 'path'
```

5. If the preceding verification succeeds, then rejuvenate the quarantined site, as follows:

```
crsctl modify cluster site site_name -s rejuvenate
```

Rolling Upgrade and Driver Installation Issues

During an upgrade, while running the Oracle Clusterware `root.sh` script, you may see the following messages:

- ACFS-9427 Failed to unload ADVN/ACFS drivers. A system restart is recommended.
- ACFS-9428 Failed to load ADVN/ACFS drivers. A system restart is recommended.

If you see these error messages during the upgrade of the initial (first) node, then do the following:

1. Complete the upgrade of all other nodes in the cluster.
2. Restart the initial node.
3. Run the `root.sh` script on initial node again.
4. Run the `Grid_home/gridSetup -executeConfigTools` script as the user that installed Oracle Grid Infrastructure to complete the upgrade.

For nodes other than the initial node (the node on which you started the installation):

1. Restart the node where the error occurs.
2. Run the `oraInstRoot.sh` script as `root` on the node where the error occurs.
3. Change directory to the Grid home, and run the `root.sh` script on the node where the error occurs.

Testing Zone Delegation



See Also:

[Oracle Clusterware Control \(CRSCTL\) Utility Reference](#) for information about using the CRSCTL commands referred to in this procedure

Use the following procedure to test zone delegation:

1. Start the GNS VIP by running the following command as `root`:

```
# crsctl start ip -A IP_name/netmask/interface_name
```

The `interface_name` should be the public interface and netmask of the public network.

2. Start the test DNS server on the GNS VIP by running the following command (you must run this command as `root` if the port number is less than 1024):

```
# crsctl start testdns -address address [-port port]
```

This command starts the test DNS server to listen for DNS forwarded packets at the specified IP and port.

3. Ensure that the GNS VIP is reachable from other nodes by running the following command as `root`:

```
crsctl status ip -A IP_name
```

4. Query the DNS server directly by running the following command:

```
crsctl query dns -name name -dnsserver DNS_server_address
```

This command fails with the following error:

CRS-10023: Domain name look up for name asdf.foo.com failed. Operating system error: Host name lookup failure

Look at *Grid_home/log/host_name/client/odnsd_*.log* to see if the query was received by the test DNS server. This validates that the DNS queries are not being blocked by a firewall.

5. Query the DNS delegation of GNS domain queries by running the following command:

```
crsctl query dns -name name
```

Note:

The only difference between this step and the previous step is that you are not giving the `-dnsserver DNS_server_address` option. This causes the command to query name servers configured in `/etc/resolv.conf`. As in the previous step, the command fails with same error. Again, look at `odnsd*.log` to ensure that `odnsd` received the queries. If step 5 succeeds but step 6 does not, then you must check the DNS configuration.

6. Stop the test DNS server by running the following command:

```
crsctl stop testdns -address address
```

7. Stop the GNS VIP by running the following command as `root`:

```
crsctl stop ip -A IP_name/netmask/interface_name
```

Oracle Clusterware Alerts

Oracle Clusterware writes messages to the ADR alert log file (as previously described) for various important events. Alert log messages generally are localized (translated) and carry a message identifier that can be used to look up additional information about the message. The alert log is the first place to look if there appears to be problems with Oracle Clusterware.

The following is an example of alert log messages from two different CRS daemon processes:

```
2014-07-16 00:27:22.074 [CTSSD(12817)]CRS-2403:The Cluster Time Synchronization
Service on host stnsp014 is in observer mode.
2014-07-16 00:27:22.146 [CTSSD(12817)]CRS-2407:The new Cluster Time Synchronization
Service reference node is host stnsp013.
2014-07-16 00:27:22.753 [CTSSD(12817)]CRS-2401:The Cluster Time Synchronization
Service started on host stnsp014.
2014-07-16 00:27:43.754 [CRSD(12975)]CRS-1012:The OCR service started on node
stnsp014.
2014-07-16 00:27:46.339 [CRSD(12975)]CRS-1201:CRSD started on node stnsp014.
```

The following example shows the start of the Oracle Cluster Time Synchronization Service (OCTSS) after a cluster reconfiguration:

```
2014-07-15 23:51:17.532 [CTSSD(12813)]CRS-2403:The Cluster Time Synchronization
Service on host stnsp014 is in observer mode.
```

```
2014-07-15 23:51:18.292 [CTSSD(12813)]CRS-2407:The new Cluster Time Synchronization
Service reference node is host stnsp013.
2014-07-15 23:51:18.961 [CTSSD(12813)]CRS-2401:The Cluster Time Synchronization
Service started on host stnsp014.
```

Alert Messages Using Diagnostic Record Unique IDs

Beginning with Oracle Database 11g release 2 (11.2), certain Oracle Clusterware messages contain a text identifier surrounded by "(:" and ":)". Usually, the identifier is part of the message text that begins with "Details in..." and includes an Oracle Clusterware diagnostic log file path and name similar to the following example. The identifier is called a DRUID, or diagnostic record unique ID:

```
2014-07-16 00:18:44.472 [ORAROOTAGENT(13098)]CRS-5822:Agent
'/scratch/12.1/grid/bin/orarootagent_root' disconnected from server. Details at
(:CRSAGF00117:) in /scratch/12.1/grid/log/stnsp014/agent/crsd/orarootagent_
root/orarootagent_root.log.
```

DRUIDs are used to relate external product messages to entries in a diagnostic log file and to internal Oracle Clusterware program code locations. They are not directly meaningful to customers and are used primarily by My Oracle Support when diagnosing problems.

Glossary

action script

A script that defines the start, stop and check actions for a resource. The start action is invoked while starting the resource, the stop action for stopping the resource, and the check action while checking the running status of a resource.

administrator managed

Database administrators define on which servers a database resource should run, and place resources manually as needed. This is the management strategy used in previous releases.

agent

A program that contains the agent framework and user code to manage resources.

agent framework

A C library that enables users to plug in application-specific code to manage customized applications.

availability directive

Instructions to Oracle Clusterware to reconfigure the system when a server leaves or joins a cluster.

cardinality

The number of servers on which a [resource](#) can run, simultaneously.

client cluster

In a multicluster environment, a client cluster advertises its names with the [server cluster](#).

cluster

Multiple interconnected computers or servers that appear as if they are one server to end users and applications.

cluster-aware

Any application designed to be deployed using clusterware.

cluster administrator

An administrator who can manage a certain part of a cluster based on set policies and privileges.

cluster configuration policy

A document for policy-managed clusters, which contains exactly one definition for each server pool defined in the system.

cluster configuration policy set

A document that defines the names of all server pools configured in the cluster, which contains one or more configuration policies. Only one policy can be in effect at any one time, but administrators can set different policies to be in effect at different dates or times of the day in accordance with business needs and system demands.

Cluster Health Monitor

Detects and analyzes operating system and cluster resource-related degradation and failures.

cluster resource

A resource that is aware of the cluster environment and subject to cross-node switchover and failover.

Cluster Time Synchronization Service

A time synchronization mechanism that ensures that all internal clocks of all nodes in a cluster are synchronized.

Cluster Verification Utility (CVU)

A tool that verifies a wide range of Oracle RAC components such as shared storage devices, networking configurations, system requirements, Oracle Clusterware, groups, and users.

database cloud

A set of databases integrated by the global service and load management framework into a single virtual server that offers one or more global services, while ensuring high performance, availability, and optimal utilization of resources.

dependency

The relationship between two or more resources. and the interaction expressed between them.

disk group

An Oracle ASM disk group is a collection of disks that Oracle ASM manages as a unit. Within a disk group, Oracle ASM exposes a file system interface for Oracle Database files. The content of files that is stored in a disk group is evenly distributed, or striped, to eliminate hot spots and to provide uniform performance across the disks. Oracle ASM files may also be optionally mirrored within a disk group. The performance of disks in a disk group is comparable to the performance of raw devices.

Dynamic Host Configuration Protocol (DHCP)

A network application protocol used by devices (DHCP clients) to obtain configuration information for operation in an Internet Protocol network. This protocol reduces system

administration workload, allowing devices to be added to the network with little or no manual intervention.

fixup script

Oracle Universal Installer detects when minimum requirements for installation are not completed, and creates shell script programs, called fixup scripts, to resolve many incomplete system configuration requirements. If Oracle Universal Installer detects an incomplete task, it prompts you to create a fixup script and then to run the fixup script in a separate terminal session. You can also generate fixup scripts with certain CVU commands by using the `-fixup` flag.

gold image

A copy of a software only, installed Oracle home, where the home-specific details are unavailable. This enables you to easily copy an image of an Oracle home to a new host on a new file system to serve as an active usable Oracle home.

A gold image can correspond, generally, to any application but in the context of this document, gold images correspond to Oracle Databases. Also, a gold image is *not* a ship home, which means that you do not have to run Oracle Universal Installer when deploying a gold image.

Grid Home client

The Grid Home client is a remote [Oracle Grid Infrastructure](#) cluster that subscribes to the [Fleet Patching and Provisioning Server](#) for provisioning of [gold images](#).

Oracle Grid Infrastructure

The software that provides the infrastructure for an enterprise grid architecture. In a cluster this software includes Oracle Clusterware and Oracle ASM. For a standalone server, this software includes Oracle Restart and Oracle ASM. Oracle Database 12c combines these infrastructure products into one software installation called the *Oracle Grid Infrastructure home* (`Grid_home`).

Grid Naming Service (GNS)

A generic service which resolves the names of hosts in a delegated normal DNS zone by mapping them to IP addresses within the zone. GNS enables the use of Dynamic Host Configuration Protocol (DHCP) address for Oracle RAC database nodes, simplifying deployment. GNS also resolves host names passed back from a SCAN listener.

Hub anchor

A Hub Node in an Oracle Flex Cluster that acts as the connection point for purposes of cluster membership for one or more other supported nodes. Those nodes exchange heartbeats with a single Hub anchor.

Hub Node

A node in an [Oracle Flex Cluster](#) that is tightly connected with other servers and has direct access to shared storage.

image

Master copy of Oracle database home software which resides on the Fleet Patching and Provisioning Server.

IPv4

Internet Protocol Version 4. IPv4 is the current standard for the IP protocol. IPv4 uses 32-bit (four-byte) addresses, which are typically represented in dotted-decimal notation. The decimal value of each octet is separated by a period, as in 192.168.2.22.

IPv6

Internet Protocol Version 6. The protocol designed to replace IPv4. In IPv6, an IP address is typically represented in eight fields of hexadecimal values separated by colons, as in 2001:0DB8:0000:0000:0000:0000:1428:57AB. In some cases, fields with 0 values can be compressed, as in 2001:DB8::1428:57AB.

local resource

A resource that runs on the nodes of a cluster but is unaware of anything outside of the scope of the node.

OCR writer

Each CRSD process also acts as an OCR server. One of the CRSD processes in the cluster is the OCR server that performs I/O to the disk group or file, or block or raw device.

Oracle Automatic Storage Management (Oracle ASM)

Oracle ASM manages the storage disks used by Oracle Clusterware and Oracle RAC in disk groups. By default, Oracle Clusterware uses Oracle ASM to store OCR and voting files.

Oracle Cluster Registry (OCR)

The Oracle RAC configuration information repository that manages information about the cluster node list and instance-to-node mapping information. OCR also manages information about Oracle Clusterware resource profiles for customized applications.

Oracle Clusterware

Software that allows groups (clusters) of connected servers to operate or be controlled as a unit.

Oracle Clusterware stack

The Oracle Clusterware stack includes Oracle Cluster Ready Services, Event Manager, Cluster Synchronization Services, and Oracle ASM (if used).

Oracle Flex Cluster

Large clusters that are made of up of [Hub Nodes](#) and other supported nodes, where the Hub Nodes form a cluster using current membership algorithms and the other nodes connect for membership to a single Hub Node.

policy managed

Database administrators specify the server pool (excluding Generic or Free) in which the database resource runs. Oracle Clusterware places the database resource on a server.

Fleet Patching and Provisioning Server

The Fleet Patching and Provisioning Server is one server on the server cluster that can hold a repository of [gold images](#).

reader node

A reader node is a nodes on which an Oracle RAC database instance runs.

resource

A database, application, or process managed by Oracle Clusterware.

resource group

A container for an application that is comprised of various logically related resources that Oracle Clusterware manages and monitors as a single entity.

resource state

The state of a particular resource at any given time that determines its availability to the cluster.

resource type

Defines whether a resource is either a [cluster resource](#) or a [local resource](#).

server categorization

A method of identifying servers in the cluster into different categories by using a set of server attributes. Administrators can configure server pools to restrict which categories they accept. Server categories are created by providing attributes for the `SERVER_CATEGORY` parameter.

server cluster

In a multicluster environment, a server cluster is a cluster in which the Grid Naming Service (GNS) process runs.

server pool

A logical division of servers in a cluster into a group that hosts applications, databases, or both.

Single Client Access Name (SCAN)

A single name that resolves to three IP addresses in the public network.

start effort evaluation

The process that Oracle Clusterware goes through when it starts a resource. During this process, Oracle Clusterware considers resource dependencies contained in the profile of the resource.

state

A set of values that describes what the condition of a particular resource.

working copy

An Oracle ACFS snapshot of a [gold image](#) that can either be a software-only installed Oracle home or an instantiated, configured Oracle home.

zone data

The GNS daemon records saved in the Oracle Cluster Registry.

Index

A

access control list (ACL), [2-2](#)

ACL

resource attribute, [B-3](#)

action entry points

defined, [9-23](#)

action scripts, [9-20](#)

actions, [9-23](#)

ACTION_FAILURE_EVENT_TEMPLATE

resource attribute, [B-19](#)

ACTION_SCRIPT

resource attribute, [B-4](#)

ACTION_TIMEOUT

resource attribute, [B-4](#)

ACTIONS

resource attribute, [B-4](#)

ACTIVE_PLACEMENT

resource attribute, [B-5](#)

add nodes to a cluster, [7-4](#)

addresses, configuring manually, [1-9](#)

administrative tools

overview and concepts, [1-21](#)

ADR, [L-12](#)

directory structure, [L-12](#)

ADR home, [L-12](#)

agent

defined, [9-16](#)

agent framework

defined, [9-16](#)

agent programs, [9-19](#)

AGENT_FILENAME

resource attribute, [B-5](#)

agents, [9-19](#)

appagent, [1-11](#), [9-19](#)

appagent.exe, [1-11](#), [9-19](#)

scriptagent, [1-11](#), [9-19](#)

scriptagent.exe, [1-11](#), [9-19](#)

alert messages

CRSD, [L-19](#)

ALERT_TEMPLATE

resource attribute, [B-5](#)

APIs

agent framework

clsagfw_add_type(), [H-5](#)

APIs (*continued*)

agent framework (*continued*)

clsagfw_check_resource(), [H-5](#)

clsagfw_create_attr_iterator(), [H-6](#)

clsagfw_delete_cookie(), [H-6](#)

clsagfw_exit2(), [H-6](#)

clsagfw_get_attr_from_iterator(), [H-7](#)

clsagfw_get_attrvalue(), [H-7](#)

clsagfw_get_check_type(), [H-8](#)

clsagfw_get_cmdid(), [H-9](#)

clsagfw_get_cookie(), [H-9](#)

clsagfw_get_request_action_name(), [H-9](#)

clsagfw_get_resource_id(), [H-10](#)

clsagfw_get_resource_name(), [H-10](#)

clsagfw_get_retry_count(), [H-10](#)

clsagfw_get_type_name(), [H-11](#)

clsagfw_init(), [H-11](#)

clsagfw_is_cmd_timedout(), [H-12](#)

clsagfw_log(), [H-12](#)

clsagfw_reset_attr_iterator(), [H-13](#)

clsagfw_send_status2(), [H-13](#)

clsagfw_set_cookie(), [H-14](#)

clsagfw_set_entrypoint(), [H-14](#)

clsagfw_set_exitcb(), [H-15](#)

clsagfw_set_resource_state_label(),
[H-15](#)

clsagfw_startup(), [H-16](#)

clscrs_stat2, [I-10](#)

command evaluation

clscrs_whatif_add_server, [I-23](#)

clscrs_whatif_delete_server, [I-24](#)

clscrs_whatif_register_serverpool, [I-21](#)

clscrs_whatif_set_activepolicy, [I-17](#)

clscrs_whatif_unregister_serverpool, [I-22](#)

miscellaneous

clscrs_get_error_details, [I-31](#)

clscrs_request_action, [I-32](#)

clscrs_restart_resource, [I-32](#)

clscrs_start_resource_in_pools, [I-33](#)

clscrs_stat3, [I-30](#)

clscrs_stop_resource_in_pools, [I-34](#)

server categorization

clscrs_get_server_by_category, [I-28](#)

clscrs_register_server, [I-28](#)

clscrs_register_servercategory, [I-26](#)

APIs (*continued*)

- server categorization (*continued*)
 - clscrs_servercategory_create, [I-25](#)
 - clscrs_servercategory_destroy, [I-26](#)
 - clscrs_unregister_servercategory, [I-27](#)

What-If

- clscrs_whatif_fail_resource, [I-18](#)
- clscrs_whatif_register_resource, [I-18](#)
- clscrs_whatif_relocate_resource, [I-19](#)
- clscrs_whatif_relocate_server, [I-24](#)
- clscrs_whatif_start_resource, [I-20](#)
- clscrs_whatif_stop_resource, [I-20](#)

application agent, [1-11](#), [9-19](#)

Application Member Cluster, [1-19](#)

application resources

- starting, [9-44](#)
- stopping, [9-45](#)

application VIP

- deleting, [9-33](#)

applications

- defining a VIP address, [1-24](#)
- highly available, [1-24](#)
- managing with CLSCRS commands, [I-1](#)
- placement policies, [9-42](#)
- registering as a resource, [9-33](#)
- relocating, [9-45](#)

appvipcfg create, [9-33](#)

attraction start dependency, [9-27](#)

- modifiers, [B-14](#)

AUTO_START

- resource attribute, [B-5](#)

autoconfig, [1-8](#), [2-10](#)

automatic address configuration, [2-12](#)

automatic cluster address configuration, [2-10](#)

Automatic Diagnostic Repository

See [ADR](#)

B

background processes, [1-10](#)

Baseboard Management Controller (BMC)

See [Intelligent Platform Management Interface \(IPMI\)](#)

black-box virtual machine, [9-2](#)

block-by-block checksum operation, [J-12](#)

built-in agents, [9-19](#)

C

Cache Fusion

- communication, [D-3](#)

CARDINALITY

- resource attribute, [B-6](#)
- resource group attribute, [B-6](#)

CARDINALITY_ID

- resource attribute, [B-6](#)
- resource group attribute, [B-6](#)

changing a network interface

- network interface
 - changing, [2-37](#)

changing interface names, consequences, [2-36](#)

changing the cluster mode, [4-3](#)

changing VIP addresses, [2-32](#)

CHECK_INTERVAL

- resource attribute, [B-6](#)

CHECK_TIMEOUT

- resource attribute, [B-6](#)

checksum operation

- block-by-block, [J-12](#)

CHM

See [Cluster Health Monitor \(CHM\)](#)

cipher suites, [1-15](#)

CLEAN_TIMEOUT

- resource attribute, [B-7](#)

client cluster, [1-8](#)

Client Data File

Using to configure shared GNS, [2-16](#)

client-side diagnosability infrastructure (CRSD)

- alerts, [L-19](#)

cloning

Oracle Clusterware, [8-1](#)

Oracle Grid Infrastructure, [8-1](#)

cloning, overview for clusterware, [1-24](#)

CLSCRS

command evaluation APIs, [I-15](#)

described, [I-15](#)

reasoned command evaluation APIs, [I-15](#)

described, [I-15](#)

CLSCRS APIs

callback mechanism, [I-2](#)

data structures, [I-2](#)

clscrs_crntity, [I-2](#)

clscrs_crntitylist, [I-2](#)

clscrs_entity_type, [I-2](#)

clscrs_sp, [I-2](#)

clscrs_splist, [I-2](#)

deprecated, [I-8](#)

error handling and tracing, [I-2](#)

filters, [E-2](#), [I-2](#)

comparison filter, [I-7](#)

expression filter, [I-2](#)

initialization and persistence, [I-2](#)

interactive, [I-10](#)

memory management, [I-2](#)

non-interactive, [I-12](#)

overview, [I-1](#)

threading support, [I-2](#)

CLSCRS commands

overview, [I-1](#)

- clscrs_crsentity, [1-2](#)
- clscrs_crsentitylist, [1-2](#)
- clscrs_entity_type, [1-2](#)
- clscrs_sp, [1-2](#)
- clscrs_splist, [1-2](#)
- CLSD-1009 message
 - resolving, [6-10](#)
- CLSD-1011 message
 - resolving, [6-10](#)
- cluster
 - adding a node, [5-76](#)
 - converting to GNS client cluster, [2-17](#)
- cluster configuration policy, [3-9](#)
- Cluster Health Monitor (CHM), [1-21](#)
- Cluster Health Monitor (CRF)
 - debugging, [E-150](#)
- cluster interconnect
 - Cache Fusion communication, [D-3](#)
 - changing private network addresses, [2-35](#)
- cluster mode
 - changing, [4-3](#)
- cluster nodes
 - adding and deleting, [7-1](#)
 - deleting from Windows systems, [7-13](#)
- Cluster Ready Services (CRS), [1-10](#)
 - debugging, [E-150](#)
 - defined, [1-10](#)
- cluster resource, [9-13](#)
- cluster resource activity log, [L-2](#)
- cluster resource failures
 - monitoring, [L-2](#)
- cluster storage
 - recording with OCR, [1-7](#)
- Cluster Synchronization Services (CSS), [1-10](#)
 - debugging, [E-150](#)
 - defined, [1-10](#)
- cluster time management, [1-25](#)
- Cluster Time Synchronization Service (CTSS), [1-25](#)
 - debugging, [E-150](#)
- Cluster Verification Utility, [A-7](#)
 - Cluster Verification Utility, [A-7](#)
 - See also CVU
 - deprecated and unsupported commands, [A-7](#)
 - See also cluvfy
- Cluster Verification Utility (CVU)
 - See CVU
- CLUSTER_INTERCONNECT interface
 - specifying with OIFCFG, [D-3](#)
- CLUSTER_INTERCONNECTS initialization
 - parameter, [2-37](#)
- cluster_resource, [9-13](#)
- cluster-aware resource, [9-13](#)
- clusters
 - administrator-managed, [2-1](#)
- clusters (*continued*)
 - converting, [2-15](#)
 - converting to an Oracle Flex Cluster, [4-3](#)
 - policy-managed, [2-1](#)
- clusterware, cloning overview, [1-24](#)
- cluvfy
 - See CVU
- comma-delimited lists with SRVCTL, [G-1](#)
- command evaluation APIs, [1-15](#)
 - sequence ID, [1-15](#)
- compatibility
 - Oracle Clusterware, Oracle ASM, and database, [1-16](#)
- Configuration Wizard
 - configuring multiple nodes, [2-5](#)
- configurations
 - reinitializing OCR, [6-19](#)
- configuring
 - voting files, [6-1](#)
- converting clusters, [2-15](#)
- CRITICAL_RESOURCES
 - resource group attribute, [B-7](#)
- cross-cluster dependency proxies, [2-44](#)
- CRS
 - See Cluster Ready Services (CRS)
- CRSCTL, [1-21](#)
 - checking the Oracle Clusterware status, [6-8](#)
 - command evaluation, [1-23](#)
 - commands
 - add category, [E-62](#)
 - add cluster site, [E-70](#)
 - add crs administrator, [E-79](#)
 - add policy, [E-103](#)
 - add resource, [E-7](#)
 - add resourcegroup, [E-33](#)
 - add resourcegroupstype, [E-47](#)
 - add serverpool, [E-125](#)
 - add type, [E-50](#)
 - add wallet, [E-57](#)
 - check cluster, [E-65](#)
 - check crs, [E-76](#)
 - check css, [E-6](#)
 - check ctss, [E-94](#)
 - check evm, [E-6](#)
 - check has, [E-146](#)
 - check resource, [E-111](#)
 - check resourcegroup, [E-33](#)
 - config crs, [E-76](#)
 - config has, [E-146](#)
 - create member_cluster_configuration, [E-100](#)
 - create policysset, [E-108](#)
 - delete category, [E-63](#)
 - delete cluster site, [E-71](#)
 - delete crs administrator, [E-80](#)

CRSCTL (*continued*)

commands (*continued*)
 delete css votedisk, [E-92](#)
 delete member_cluster_configuration,
 [E-101](#)
 delete node, [E-102](#)
 delete policy, [E-104](#)
 delete resource, [E-11](#)
 delete resourcegroup, [E-34](#)
 delete resourcegrouptype, [E-48](#)
 delete serverpool, [E-127](#)
 delete type, [E-53](#)
 delete wallet, [E-58](#)
 disable crs, [E-76](#)
 disable has, [E-146](#)
 discover dhcp, [E-94](#)
 enable crs, [E-77](#)
 enable has, [E-147](#)
 eval activate policy, [E-105](#)
 eval add resource, [E-12](#)
 eval add resourcegroup, [E-35](#)
 eval add server, [E-116](#)
 eval add serverpool, [E-127](#)
 eval delete server, [E-118](#)
 eval delete serverpool, [E-130](#)
 eval fail resource, [E-14](#)
 eval fail resourcegroup, [E-36](#)
 eval modify resource, [E-16](#)
 eval modify serverpool, [E-132](#)
 eval relocate resource, [E-15](#)
 eval relocate resourcegroup, [E-36](#)
 eval relocate server, [E-119](#)
 eval start resource, [E-18](#)
 eval start resourcegroup, [E-37](#)
 eval stop resource, [E-18](#)
 eval stop resourcegroup, [E-38](#)
 export resourcegroup, [E-38](#)
 get calog maxsize, [L-11](#)
 get calog retentiontime, [L-11](#)
 get clientid dhcp, [E-95](#)
 get cluster class, [E-68](#), [E-92](#)
 get cluster configuration, [E-68](#)
 get cluster extended, [E-68](#)
 get cluster hubsize, [E-69](#)
 get cluster mode, [E-69](#)
 get cluster name, [E-70](#)
 get cluster tlsciphersuite, [E-73](#)
 get cluster type, [E-73](#)
 get cpu equivalency, [E-73](#)
 get css, [E-85](#)
 get css ipmiaddr, [E-88](#)
 get hostname, [E-7](#)
 get ipmi binaryloc, [E-87](#)
 get node role, [E-103](#)
 get nodename, [E-102](#)

CRSCTL (*continued*)

commands (*continued*)
 get resource use, [E-114](#)
 get restricted placement, [E-74](#)
 get server css_critical, [E-124](#)
 get server label, [E-124](#)
 get tracefileopts, [E-49](#)
 getperm resource, [E-19](#)
 getperm serverpool, [E-135](#)
 getperm type, [E-53](#)
 lsmodules, [E-99](#)
 modify category, [E-64](#)
 modify cluster site, [E-71](#)
 modify policy, [E-106](#)
 modify policysset, [E-108](#)
 modify resource, [E-20](#)
 modify resourcegroup, [E-39](#)
 modify resourcegrouptype, [E-49](#)
 modify server, [E-120](#)
 modify serverpool, [E-136](#)
 modify type, [E-54](#)
 modify wallet, [E-59](#)
 pin css, [E-85](#)
 query calog, [L-3](#)
 query cluster site, [E-72](#)
 query crs activeversion, [E-79](#)
 query crs administrator, [E-81](#)
 query crs autostart, [E-82](#)
 query crs releasepatch, [E-83](#)
 query crs releaseversion, [E-83](#)
 query crs site, [E-83](#)
 query crs softwarepatch, [E-84](#)
 query crs softwareversion, [E-84](#)
 query css ipmicfg, [E-90](#)
 query css ipmiddevice, [E-91](#)
 query css votedisk, [E-93](#)
 query dns, [E-97](#)
 query has releaseversion, [E-147](#)
 query has softwareversion, [E-147](#)
 query member_cluster_configuration,
 [E-101](#)
 query socket udp, [E-141](#)
 query wallet, [E-60](#)
 release dhcp, [E-95](#)
 relocate resource, [E-22](#), [E-112](#)
 relocate resourcegroup, [E-41](#)
 relocate server, [E-121](#)
 replace discoverystring, [E-96](#)
 replace votedisk, [E-144](#)
 request action, [E-61](#)
 request dhcp, [E-96](#)
 restart resource, [E-24](#)
 restart resourcegroup, [E-42](#)
 set calog maxsize, [L-11](#)
 set calog retentiontime, [L-12](#)

CRSCTL (*continued*)

- commands (*continued*)
- set cluster disabledtlsciphersuite, [E-68](#)
- set cluster hubsize, [E-69](#)
- set cluster mode, [E-70](#)
- set cluster type, [E-73](#)
- set cpu equivalency, [E-74](#)
- set crs autostart, [E-82](#)
- set css, [E-86](#)
- set css ipmiaddr, [E-89](#)
- set css ipmiadmin, [E-89](#)
- set css noautorestart, [E-92](#)
- set ipmi binaryloc, [E-88](#)
- set log, [E-149](#)
- set resource use, [E-114](#)
- set restricted placement, [E-75](#)
- set server css_critical, [E-124](#)
- set server label, [E-124](#)
- set tracefileopts, [E-49](#)
- setperm resource, [E-25](#)
- setperm serverpool, [E-138](#)
- setperm type, [E-55](#)
- start cluster, [E-66](#)
- start crs, [E-77](#)
- start has, [E-148](#)
- start ip, [E-98](#)
- start resource, [E-27](#)
- start resourcegroup, [E-43](#)
- start rollingpatch, [E-115](#)
- start rollingupgrade, [E-116](#)
- start testdns, [E-142](#)
- status category, [E-65](#)
- status ip, [E-98](#)
- status policy, [E-107](#)
- status policyset, [E-109](#)
- status resource, [E-28](#)
- status resourcegroup, [E-44](#)
- status server, [E-122](#)
- status serverpool, [E-139](#)
- status testdns, [E-143](#)
- status type, [E-56](#)
- stop cluster, [E-67](#)
- stop crs, [E-78](#)
- stop has, [E-148](#)
- stop ip, [E-99](#)
- stop resource, [E-31](#)
- stop resourcegroup, [E-46](#)
- stop rollingpatch, [E-115](#)
- stop testdns, [E-144](#)
- unpin css, [E-86](#)
- unset css, [E-87](#)
- unset css ipmiconfig, [E-90](#)
- dual environment commands, [E-6](#)
- Oracle RAC environment commands, [E-61](#)

CRSCTL (*continued*)

- Oracle Restart environment commands, [E-145](#)
- CRSCTL commands
 - cluster aware, [1-21](#)
 - debug log, [E-150](#)
 - lsmodules, [E-150](#)
 - module_name parameter, [E-150](#)
 - overview, [1-21](#)
 - resource_name parameter, [E-153](#)
 - set log, [E-153](#)
- crsctl set log, [E-149](#)
- crsctl set trace, [E-149](#)
- CRSD background process
 - alert messages, [L-19](#)
- CSS
 - See Cluster Synchronization Services (CSS)
- CVU, [1-21](#), [A-7](#)
 - about, [A-1](#)
 - baseline creation, [A-14](#)
 - commands
 - cluvfy comp acfs, [A-11](#)
 - cluvfy comp admprv, [A-11](#)
 - cluvfy comp asm, [A-13](#)
 - cluvfy comp baseline, [A-14](#)
 - cluvfy comp clocksync, [A-16](#)
 - cluvfy comp clumgr, [A-16](#)
 - cluvfy comp crs, [A-17](#)
 - cluvfy comp DHCP, [A-17](#)
 - cluvfy comp dns, [A-19](#)
 - cluvfy comp freespace, [A-20](#)
 - cluvfy comp gns, [A-20](#)
 - cluvfy comp gnpnp, [A-21](#)
 - cluvfy comp ha, [A-22](#)
 - cluvfy comp healthcheck, [A-22](#)
 - cluvfy comp nodeapp, [A-23](#)
 - cluvfy comp nodecon, [2-32](#), [A-24](#)
 - cluvfy comp nodereach, [A-25](#)
 - cluvfy comp ocr, [A-26](#)
 - cluvfy comp ohasd, [A-27](#)
 - cluvfy comp olr, [A-28](#)
 - cluvfy comp peer, [A-29](#)
 - cluvfy comp scan, [A-30](#)
 - cluvfy comp software, [A-30](#)
 - cluvfy comp space, [A-31](#)
 - cluvfy comp ssa, [A-32](#)
 - cluvfy comp sys, [A-34](#)
 - cluvfy comp vdisk, [A-36](#)
 - cluvfy stage -post acfscfg, [A-36](#)
 - cluvfy stage -post appcluster, [A-37](#)
 - cluvfy stage -post cfs, [A-37](#)
 - cluvfy stage -post crsinst, [A-38](#)
 - cluvfy stage -post hacfg, [A-43](#)
 - cluvfy stage -post hwos, [A-44](#)
 - cluvfy stage -post nodeadd, [A-45](#)

CVU (continued)

- commands (continued)
- cluvfy stage -post nodedel, [A-46](#)
- cluvfy stage -pre acfsfg, [A-36](#)
- cluvfy stage -pre cfs, [A-37](#)
- cluvfy stage -pre crsinst, [A-38](#)
- cluvfy stage -pre dbcfg, [A-40](#)
- cluvfy stage -pre dbinst, [A-41](#)
- cluvfy stage -pre hacfg, [A-43](#)
- cluvfy stage -pre nodeadd, [A-45](#)
- component verifications
 - checking Oracle Clusterware and Oracle Database installations, [A-22](#)
 - Cluster Manager subcomponent, [A-16](#)
 - connectivity between cluster nodes, [A-24](#)
 - CTSS integrity, [A-16](#)
 - Domain Name Service (DNS), [A-19](#)
 - existence of DHCP, [A-17](#)
 - free space, [A-31](#)
 - Grid Naming Service (GNS), [A-20](#)
 - Grid Plug and Play service and profile, [A-21](#)
 - integrity of high availability, [A-22](#)
 - integrity of OCR, [A-26](#)
 - integrity of ohasd, [A-27](#)
 - integrity of OLR, [A-28](#)
 - integrity of Oracle ACFS, [A-11](#)
 - integrity of Oracle ASM, [A-13](#)
 - integrity of voting files, [A-36](#)
 - node applications, [A-23](#)
 - node comparison, [A-29](#)
 - Oracle Clusterware component, [A-17](#)
 - reachability of nodes, [A-25](#)
 - SCAN configuration, [A-30](#)
 - software distribution across nodes, [A-30](#)
 - storage, [A-32](#)
 - system requirements, [A-34](#)
 - user and permissions, [A-11](#)
- difference between runcluvfy.sh and cluvfy, [A-4](#)
- installation requirements, [A-3](#)
- installation verifications
 - application clusters, [A-37](#)
- known issues, [A-47](#)
- node list shortcuts, [A-10](#)
- online Help system, [A-7](#)
- overview and concepts, [1-21](#)
- performing verifications, [A-2](#)
- runcluvfy.sh, [A-4](#)
- stage verifications
 - database configuration, [A-40](#)
 - high availability installation, [A-43](#)
 - network and storage on all nodes, [A-44](#)
 - node deletion, [A-46](#)
 - node installation, [A-45](#)

CVU (continued)

- stage verifications (continued)
 - Oracle ACFS, [A-37](#)
 - Oracle ACFS configuration, [A-36](#)
 - Oracle Clusterware installation, [A-38](#)
 - Oracle RAC installation, [A-41](#)
- UNKNOWN output, [A-10](#)
- verbose mode, [A-9](#)

D

-
- database home
 - provisioning a working copy of, [5-41](#)
 - Database Management Repository, [1-19](#)
 - Database Member Cluster, [1-19](#)
 - debugging
 - CRS, CSS, and EVM modules, [E-150](#)
 - Oracle Clusterware resources, [E-153](#)
 - default application VIP
 - creating
 - appvipcfg create, [9-33](#)
 - defining network interfaces
 - OIFCFG command-line interface, [D-1](#)
 - DELETE_TIMEOUT
 - resource attribute, [B-7](#)
 - delif command
 - OIFCFG command-line interface, [D-5](#)
 - deployment scheme
 - deciding, [9-36](#)
 - DESCRIPTION
 - resource attribute, [B-7](#)
 - DHCP configuration, [2-12](#)
 - diagcollection.pl, [L-15](#)
 - diagnostic directories
 - alert, [L-12](#)
 - core, [L-15](#)
 - incident, [L-15](#)
 - output, [L-15](#)
 - trace, [L-13](#)
 - diagnostics collection script, [L-15](#)
 - disk group redundancy, [6-26](#)
 - external, [6-26](#)
 - high, [6-26](#)
 - normal, [6-26](#)
 - dispersion start dependency, [9-27](#)
 - modifiers, [B-14](#)
 - DNS, entries example for GNS and SCAN, [2-9](#)
 - Domain Services Cluster, [1-19](#)
 - DRUID
 - diagnostic record unique ID, [L-20](#)
 - Dynamic Host Configuration Protocol (DHCP), [1-8](#)

E

ENABLED

resource attribute, [B-7](#)

enabling debugging for Oracle Clusterware

resources, [E-153](#)

enabling debugging for the CRS, CSS, and EVM

modules, [E-150](#)

entry points

ABORT, [9-16](#)

ACTION, [9-16](#)

CHECK, [9-16](#)

CLSAGFW_FAILED state, [9-16](#)

CLSAGFW_ONLINE state, [9-16](#)

CLSAGFW_PARTIAL state, [9-16](#)

CLSAGFW_PLANNED_OFFLINE state,
[9-16](#)

CLSAGFW_UNKNOWN state, [9-16](#)

CLSAGFW_UNPLANNED_OFFLINE
state, [9-16](#)

CLEAN, [9-16](#)

defined, [9-16](#)

DELETE, [9-16](#)

MODIFY, [9-16](#)

monitor, [9-16](#)

START, [9-16](#)

STOP, [9-16](#)

Event Management (EVM), [1-10](#)

defined, [1-10](#)

Event Manager (EVM)

debugging, [E-150](#)

EVM, [1-10](#)

overview, [1-10](#)

See also Event Management (EVM)

exclusion start dependency, [9-28](#)

modifiers, [B-14](#)

extending Oracle database home

on non-shared storage, [7-4](#)

on shared storage

network-attached storage, [7-4](#)

Oracle ACFS, [7-4](#)

F

failure groups, [6-26](#)

quorum, [6-26](#)

failure isolation

configuring IPMI for, [2-23](#)

FAILURE_INTERVAL

resource attribute, [B-8](#)

FAILURE_THRESHOLD

resource attribute, [B-8](#)

Fast Application Notification (FAN), [1-10](#)

Fleet Patching and Provisioning, [5-1](#), [F-1](#)

adding cluster nodes with, [7-4](#)

Fleet Patching and Provisioning (*continued*)

architecture, [5-7](#)

components

Fleet Patching and Provisioning Client,
[5-10](#)

Fleet Patching and Provisioning Server,
[5-9](#)

Fleet Patching and Provisioning targets,
[5-10](#)

NFS home client, [5-10](#)

creating databases on a working copy, [5-42](#)

features, [5-1](#)

gold image distribution, [5-15](#)

image series, [5-20](#)

image state, [5-20](#)

image type, [5-20](#)

images, [5-14](#)

job scheduler, [5-66](#)

managing Clients, [5-55](#)

moving databases between working copies,
[5-43](#)

notifications, [5-17](#)

Oracle Database software management,
[5-41](#)

patching Oracle Exadata software, [5-46](#)

patching Oracle Grid Infrastructure (non-
rolling), [5-32](#)

patching Oracle Grid Infrastructure (rolling),
[5-31](#)

persistent home path, [5-54](#)

provisioned software

storage options, [5-24](#)

provisioning a working copy of a database
home, [5-41](#)

provisioning for a different user, [5-25](#)

roles, [5-12](#)

basic built-in, [5-13](#)

composite built-in, [5-14](#)

troubleshooting, [L-1](#)

upgrading Oracle Database software, [5-48](#)

user actions

creating web server, [5-77](#)

user groups management, [5-22](#)

user-defined actions, [5-60](#)

Fleet Patching and Provisioning Client, [5-10](#)

assigning roles to users, [5-59](#)

creating, [5-55](#)

creating users, [5-59](#)

enabling and disabling, [5-57](#)

managing, [5-55](#)

managing the password, [5-59](#)

troubleshooting incompatible, [L-1](#)

Fleet Patching and Provisioning commands

See RHPCTL

Fleet Patching and Provisioning Server, [5-1](#)
 creating, [5-18](#)
 Oracle Grid Infrastructure software management, [5-27](#)
 propagating images between, [5-26](#)
 Free server pool, [3-3](#)
 described, [3-3](#)

G

generic server pool
 described, [3-3](#)
 Generic server pool, [3-3](#)
 generic_application, [9-13](#)
 resource type
 creating resources of, [9-39](#)
 getif command
 OIFCFG command-line interface, [D-5](#)
 global interface
 network interface stored as, [D-2](#)
 GNS, [1-8–1-10](#)
 administering, [2-13](#)
 changing the subdomain, [2-19](#)
 converting non-GNS cluster to GNS cluster, [2-16](#)
 highly available
 removing secondary GNS instances, [2-14](#)
 highly-available, [2-11](#)
 configuring, [2-13](#)
 removing primary GNS instances, [2-14](#)
 primary GNS instance, [2-11](#)
 secondary GNS instance, [2-11](#)
 starting, [2-15](#)
 stopping, [2-15](#)
 zone data, [2-11](#)
 zone transfer, [2-11](#)
 See also Grid Naming Service (GNS)
 GNS daemon
 and GNS VIP, [2-9](#)
 port number for, [2-9](#)
 GNS, GNS VIP, [2-9](#)
 gold image, [5-1](#)
 gold images
 adding, [5-76](#)
 adding to Fleet Patching and Provisioning server, [5-19](#)
 Grid Infrastructure Management Repository, [1-18](#)
 attributes and requirements, [1-18](#)
 Grid Interprocess Communication (gipc)
 debugging, [E-150](#)
 Grid Interprocess Communication (GIPC), [1-11](#)
 Grid Naming Service (GNS), [1-10](#)
 defined, [1-10](#)
 See GNS, [1-8](#)

Grid Plug and Play (gpnpp)
 debugging, [E-150](#)
 Grid Plug and Play (GPNPD), [1-11](#)
 gridSetup.sh
 using to add nodes, [7-4](#)

H

HAIP
 highly available IP address, [2-35](#)
 hard start dependency, [9-28](#)
 modifiers, [B-14](#)
 hard stop dependency, [9-31](#)
 modifiers, [B-17](#)
 hardware requirements, [1-5](#)
 high availability
 and Oracle Clusterware, [1-10](#)
 application programming interface, [1-24](#)
 framework, [1-24](#)
 highly-available GNS, [2-11](#)
 configuring, [2-13](#)
 HOSTING_MEMBERS
 resource attribute, [B-8](#)
 Hub Node, [4-1](#)

I

iflist command
 OIFCFG command-line interface, [D-5](#)
 importing
 OCR, [6-19](#)
 incident trace files, [L-14](#)
 independent automaton, [5-38, 5-45](#)
 initialization parameters
 CLUSTER_INTERCONNECTS, [2-37](#)
 installation
 introduction, [1-16](#)
 installations
 configuring voting files, [6-1](#)
 INSTANCE_COUNT
 resource attribute, [B-19](#)
 INSTANCE_FAILOVER
 resource attribute, [B-9](#)
 Intelligent Management Platform Interface (IPMI), [2-21, 2-24](#)
 configuring for failure isolation, [2-23](#)
 removing IPMI configuration, [2-25](#)
 Intelligent Platform Management Interface (IPMI), [2-21](#)
 CRSTCTL commands
 get css ipmiaddr, [E-88](#)
 get ipmi binaryloc, [E-87](#)
 query css ipmicfg, [E-90](#)
 query css ipmidevice, [E-91](#)
 set css ipmiaddr, [E-89](#)

Intelligent Platform Management Interface (IPMI) (*continued*)CRSCTL commands (*continued*)set css ipmiadmin, [E-89](#)set ipmi binaryloc, [E-88](#)unset css ipmicfg, [E-90](#)interface names, consequences of changing,
[2-36](#)

INTERMEDIATE_TIMEOUT

resource attribute, [B-9](#)

INTERNAL_STATE

resource attribute, [B-19](#)IPMI, [2-21](#)modifying administrator, [E-89](#)obtaining IP address, [E-88](#)storing IP address, [E-89](#)See also Intelligent Management Platform
Interface (IPMI)IPv4, [2-26](#)changing to a static IPv6 address, [2-41](#)changing to dynamic IPv6 addresses, [2-43](#)

network configuration

adding an IPv6 network to, [2-44](#)

networks

transitioning to IPv6 networks, [2-44](#)IPv6, [2-26](#)name resolution, [2-27](#)

IPv6 Stateless Address Autoconfiguration

Protocol, [2-12](#)

J

job scheduler

Fleet Patching and Provisioning, [5-66](#)

L

LAST_SERVER

resource attribute, [B-20](#)

LAST_STATE_CHANGE

resource attribute, [B-20](#)

listeners

in OCR, [1-7](#)

LOAD

resource attribute, [B-9](#)load-aware resource placement, [2-7](#), [3-11](#)local resource, [9-13](#)local_resource, [9-13](#)

log levels

setting for Oracle Clusterware, [E-149](#)

lsmodule parameter

with the CRSCTL command, [E-150](#)

managing applications

CLSCRS commands, [1-1](#)

managing Oracle Clusterware

with CRSCTL, [1-21](#)manual address configuration, [1-9](#)

mDNS

See Multicast Domain Name Service (mDNS)

mDNSResponder

purpose, [1-15](#)memory pressure, [3-15](#)

mirroring

OCR (Oracle Cluster Registry), [6-5](#)

MODIFY_TIMEOUT

resource attribute, [B-9](#)modifying IPMI configuration, [2-24](#)

module_name parameter

supplying to CRSCTL commands, [E-150](#)

modules

debugging

CRF, [E-150](#)CRS, [E-150](#)CSS, [E-150](#)CTSS, [E-150](#)EVM, [E-150](#)gipc, [E-150](#)gpnv, [E-150](#)Multicast Domain Name Service (mDNS), [1-11](#)defined, [1-11](#)

N

NAME

resource attribute, [B-10](#)

network file system

See NFS

network interface

configuration, [2-26](#)global, [D-2](#)node-specific, [D-2](#)OIFCFG syntax, [D-3](#)network interface card, [2-35](#)

network interfaces

defining with OIFCFG, [D-1](#)types, [D-3](#)updating subnet classification, [D-1](#)

networks

creating, [2-39](#)for Oracle Flex Clusters, [4-1](#)NFS home client, [5-10](#)

node-specific interface

network interface stored as, [D-2](#)

nodes

- adding to a cluster
 - on Linux or UNIX, [7-4](#)
 - on Windows, [7-10](#)
- configuring multiple, [2-5](#)
- deleting from a cluster
 - on Linux or UNIX, [7-8](#)
- VIP address, [1-8](#)

non-default application VIP

- creating, [9-33](#)

O

OCR (Oracle Cluster Registry), [1-7](#)

- adding, [6-5](#), [6-7](#)
- automatic backups, [6-11](#)
- backing up, [6-11](#)
- changing backup file location, [6-12](#)
- contents, [6-2](#)
- diagnosing problems with OCRDUMP, [6-19](#)
- downgrading, [6-25](#)
- exporting, [6-19](#)
- importing
 - on Windows systems, [6-22](#)
- importing content
 - on Linux and UNIX systems, [6-20](#)
- listing backup files, [6-11](#)
- managing, [6-2](#)
- manual backups, [6-11](#)
- migrating from Oracle ASM, [6-4](#)
- migrating to Oracle ASM, [6-2](#)
- ocr.loc file, [6-5](#)
- OCRDUMP utility command examples, [J-16](#)
- overriding data loss protection mechanism, [6-10](#)
- recording cluster configuration information, [1-3](#)
- recording cluster storage, [1-7](#)
- removing, [6-5](#), [6-7](#)
- repairing, [6-5](#), [6-9](#)
- replacing, [6-5](#), [6-8](#)
- restoring, [6-12](#)
 - in Oracle Restart, [6-18](#)
 - on Linux and UNIX systems, [6-13](#)
 - on Windows systems, [6-16](#)
 - using automatically generated OCR backups, [6-12](#)
- troubleshooting, [6-19](#), [J-11](#)
- upgrading, [6-25](#)
- viewing content with OCRDUMP, [J-15](#)

OCR configuration tool

See OCRCONFIG utility

ocr.loc file, [6-5](#)

OCRCHECK

- commands
 - config, [J-13](#)
 - local, [J-13](#)

OCRCHECK utility

- check status of OLR, [6-24](#)
- diagnosing OCR problems with, [6-19](#)
- log files, [J-13](#)
- sample output, [J-13](#)

OCRCONFIG utility, [1-21](#)

- administering OLR, [6-24](#)

commands

- add, [J-2](#)
- backuploc, [J-3](#)
- copy, [J-4](#)
- delete, [J-5](#)
- downgrade, [J-5](#)
- export, [J-5](#)
- import, [6-20](#), [J-6](#)
- manualbackup, [J-6](#)
- overwrite, [J-7](#)
- repair, [J-7](#)
- replace, [J-8](#)
- restore, [J-9](#)
- showbackup, [J-10](#)
- showbackuploc, [J-10](#)
- upgrade, [J-11](#)

log files, [J-1](#)overview and concepts, [1-21](#)syntax, [J-1](#)

OCRDUMP utility

- changing the amount of logging, [J-15](#)
- command examples, [J-16](#)
- commands, [J-16](#)
 - backup, [J-17](#)
- diagnosing OCR problems with, [6-19](#), [J-15](#)
- dump content of OLR, [6-24](#)
- log files, [J-15](#)
- sample output, [J-17](#)
- syntax and options, [J-16](#)
- SYSTEM.language key output, [J-17](#)
- SYSTEM.version key output, [J-17](#)

ocrlog.ini file

- editing, [J-15](#)

OFFLINE_CHECK_INTERVAL

- resource attribute, [B-10](#)

OIFCFG command-line interface

- commands, [D-2](#)
- interface types, [D-3](#)
- invoking, [D-1](#)
- overview and concepts, [1-21](#), [D-1](#)
- syntax, [D-2](#)

OLR (Oracle Local Registry),

- administering, [6-24](#)
- backing up, [6-25](#)

- OLR (Oracle Local Registry) (*continued*)
 - check status of, [6-24](#)
 - defined, [6-24](#)
 - dump content of, [6-24](#)
 - exporting to a file, [6-24](#)
 - importing a file, [6-24](#)
 - restoring, [6-25](#)
 - viewing backup files, [6-25](#)
 - viewing content with OCRDUMP, [J-15](#)
 - OLSNODES command
 - reference, [C-1](#)
 - ONLINE_RELOCATION_TIMEOUT
 - resource attribute, [B-10](#)
 - ONS
 - See Oracle Notification Service (ONS)
 - operating systems
 - requirements for Oracle Clusterware, [1-3](#)
 - oraagent
 - defined, [1-10](#), [1-11](#)
 - Oracle agent, [1-10](#), [1-11](#)
 - Oracle ASM
 - disk groups
 - redundancy, [6-26](#)
 - failure group, [6-26](#)
 - migrating OCR locations to, [6-2](#)
 - Oracle Cluster Registry
 - See OCR (Oracle Cluster Registry)
 - Oracle Clusterware
 - adding a home to a new node, [7-4](#)
 - alert log, [L-12](#)
 - background processes
 - on Windows, [1-15](#)
 - debugging
 - component level, [E-149](#)
 - dynamic, [E-149](#)
 - defined, [1-1](#)
 - new features, [1-1](#)
 - OCR
 - migrating from Oracle ASM, [6-4](#)
 - migrating to Oracle ASM, [6-2](#)
 - overview of administration, [1-1](#)
 - processes
 - Cluster Ready Services (CRS), [1-10](#)
 - Cluster Synchronization Services (CSS), [1-10](#)
 - Event Management (EVM), [1-10](#)
 - Grid Interprocess Communication (GIPC), [1-11](#)
 - Grid Naming Service (GNS), [1-10](#)
 - Multicast Domain Name Service (mDNS), [1-11](#)
 - oraagent, [1-10](#), [1-11](#)
 - Oracle Notification Service (ONS), [1-10](#)
 - orarootagent, [1-10](#), [1-11](#)
 - REST APIs, [K-1](#)
 - Oracle Clusterware (*continued*)
 - upgrade
 - out-of-place, [1-17](#)
 - Oracle Clusterware Control (CRSCTL)
 - See CRSCTL
 - Oracle Clusterware home
 - adding, [7-4](#), [7-10](#)
 - deleting manually, [7-8](#)
 - Oracle Database
 - creating on a working copy, [5-42](#)
 - fault diagnosability infrastructure, [L-12](#)
 - moving between working copies, [5-43](#)
 - patching, [5-45](#), [5-72](#), [5-73](#)
 - scaling, [5-76](#)
 - upgrading, [5-74](#)
 - Oracle Database QoS Management, [2-8](#)
 - Oracle Database Quality of Service Management
 - See Oracle Database QoS Management
 - Oracle Enterprise Manager
 - adding resources with, [9-41](#)
 - adding VIPs with, [9-35](#)
 - overview and concepts, [1-21](#)
 - Oracle Exadata
 - patching, [5-46](#)
 - Oracle Extended Clusters, [4-4](#)
 - configuring, [4-5](#)
 - storage split in, [L-17](#)
 - Oracle Fleet Patching and Provisioning, [5-27](#)
 - Oracle Flex Clusters, [1-3](#)
 - and Oracle Flex ASM cluster, [4-1](#)
 - changing from Oracle Clusterware standard Clusters to, [4-3](#)
 - CRSCTL commands
 - get cluster hubsize, [E-69](#)
 - get cluster mode, [E-69](#)
 - set cluster hubsize, [E-69](#)
 - set cluster mode, [E-70](#)
 - Hub Node, [4-1](#)
 - managing, [4-3](#)
 - reader nodes, [4-1](#)
- Oracle FPP, [5-27](#)
- Oracle Grid Infrastructure
 - cloning, [8-1](#)
 - cloning Oracle Clusterware in, [8-1](#)
 - creating, [5-68](#)
 - Grid Setup Wizard, [2-4](#)
 - in batches, [5-32](#)
 - patching, [5-38](#), [5-71](#), [5-72](#)
 - patching using Fleet Patching and Provisioning, [5-31](#), [5-32](#)
 - patching using the non-rolling method, [5-32](#)
 - patching using the rolling method, [5-31](#)
 - provisioning, [5-29](#)
 - upgrading, [5-70](#)

Oracle Grid Infrastructure (*continued*)
 upgrading using Fleet Patching and Provisioning, [5-40](#)
 using installer to add nodes, [7-4](#)
 zero-downtime patching, [5-38](#)

Oracle Interface Configuration tool
 see OIFCFG, [D-1](#)

Oracle Layered File System, [5-27](#)

Oracle Local Registry
 See OLR (Oracle Local Registry)

Oracle Notification Service (ONS), [1-10](#)
 defined, [1-10](#)

Oracle Restart
 patching, upgrading, [5-67](#)
 restoring OCR, [6-18](#)

Oracle root agent, [1-10](#), [1-11](#)

Oracle Services for Microsoft Transaction Server,
[7-10](#)
 creating the OraMTS service, [7-10](#)

Oracle Universal Installer
 Client Data File for shared GNS clients, [2-16](#)
 Oracle Clusterware installation, [1-16](#)

OraMTS
 See Oracle Services for Microsoft Transaction Server

orarootagent
 defined, [1-10](#), [1-11](#)

out-of-place upgrade, [1-17](#)

P

patching
 automated recovery options, [5-39](#)
 combined Oracle Grid Infrastructure and Oracle Database, [5-36](#)
 error prevention, [5-39](#)

patching software
 Fleet Patching and Provisioning-defined, [5-32](#)
 in batches, [5-32](#)
 user-defined, [5-32](#)

persistent home path, [5-54](#)

PLACEMENT
 resource attribute, [B-10](#)

pluggable database
 provisioning, [5-70](#)

policy-based management, [3-1](#)
 adding resources to server pools, [9-37](#)

primary GNS instance, [2-11](#)
 configuring, [2-13](#)
 removing, [2-14](#)

private network address
 changing, [2-35](#)

PROFILE_CHANGE_EVENT_TEMPLATE
 resource attribute, [B-20](#)

provisioning software, [5-21](#)

public interface
 specifying with OIFCFG, [D-3](#)

pullup start dependency, [9-29](#)
 modifiers, [B-15](#)

Q

quorum failure groups, [6-26](#)

R

reader nodes, [4-1](#)

reasoned command evaluation, [1-23](#)

reasoned command evaluation APIs, [I-15](#)
 sequence ID, [I-15](#)

recording cluster configuration information, [1-3](#)

recording node membership information
 voting file, [1-3](#)

redundancy
 voting file, [1-7](#)

Redundant Interconnect Usage, [2-35](#)

RELOCATE_BY_DEPENDENCY
 resource attribute, [B-11](#)

RELOCATE_KIND
 resource attribute, [B-11](#)

resource attributes
 ACL, [B-3](#)
 ACTION_FAILURE_EVENT_TEMPLATE,
[B-19](#)
 ACTION_SCRIPT, [B-4](#)
 ACTION_TIMEOUT, [B-4](#)
 ACTIONS, [B-4](#)
 ACTIVE_PLACEMENT, [B-5](#)
 AGENT_FILENAME, [B-5](#)
 ALERT_TEMPLATE, [B-5](#)
 AUTO_START, [B-5](#)
 CARDINALITY, [B-6](#)
 CARDINALITY_ID, [B-6](#)
 CHECK_INTERVAL, [B-6](#)
 CHECK_TIMEOUT, [B-6](#)
 CLEAN_TIMEOUT, [B-7](#)
 DELETE_TIMEOUT, [B-7](#)
 DESCRIPTION, [B-7](#)
 ENABLED, [B-7](#)
 FAILURE_INTERVAL, [B-8](#)
 FAILURE_THRESHOLD, [B-8](#)
 HOSTING_MEMBERS, [B-8](#)
 INSTANCE_COUNT, [B-19](#)
 INSTANCE_FAILOVER, [B-9](#)
 INTERMEDIATE_TIMEOUT, [B-9](#)
 INTERNAL_STATE, [B-19](#)
 LAST_SERVER, [B-20](#)
 LAST_STATE_CHANGE, [B-20](#)
 LOAD, [B-9](#)

- resource attributes (*continued*)
 - MODIFY_TIMEOUT, [B-9](#)
 - NAME, [B-10](#)
 - OFFLINE_CHECK_INTERVAL, [B-10](#)
 - ONLINE_RELOCATION_TIMEOUT, [B-10](#)
 - PLACEMENT, [B-10](#)
 - PROFILE_CHANGE_EVENT_TEMPLATE, [B-20](#)
 - RELOCATE_BY_DEPENDENCY, [B-11](#)
 - RELOCATE_KIND, [B-11](#)
 - RESTART_ATTEMPTS, [B-11](#)
 - RESTART_COUNT, [B-20](#)
 - SCRIPT_TIMEOUT, [B-11](#)
 - SERVER_CATEGORY, [B-12](#)
 - SERVER_POOLS, [B-12](#)
 - START_CONCURRENCY, [B-13](#)
 - START_DEPENDENCIES, [B-13](#)
 - START_TIMEOUT, [B-16](#)
 - STATE_CHANGE_EVENT_TEMPLATE, [B-21](#)
 - STATE_DETAILS, [B-21](#)
 - STOP_CONCURRENCY, [B-16](#)
 - STOP_DEPENDENCIES, [B-16](#)
 - STOP_TIMEOUT, [B-17](#)
 - TARGET, [B-21](#)
 - TARGET_SERVER, [B-21](#)
 - TYPE, [B-21](#)
 - UPTIME_THRESHOLD, [B-18](#)
 - USE_STICKINESS, [B-19](#)
 - USER_WORKLOAD, [B-18](#)
- resource dependencies
 - defined, [9-26](#)
 - start dependencies, [9-26](#)
 - attraction, [9-27](#), [B-14](#)
 - dispersion, [9-27](#), [B-14](#)
 - exclusion, [9-28](#), [B-14](#)
 - hard, [9-28](#), [B-14](#)
 - pullup, [9-29](#), [B-15](#)
 - weak, [9-30](#), [B-15](#)
 - stop dependencies, [9-31](#)
 - hard, [9-31](#), [B-17](#)
- resource group, [9-2](#)
- resource group attributes
 - CARDINALITY, [B-6](#)
 - CARDINALITY_ID, [B-6](#)
 - CRITICAL_RESOURCES, [B-7](#)
 - RESOURCE_LIST, [B-20](#)
- resource groups
 - about, [9-5](#)
 - adding resources to, [9-44](#)
 - automatic resource groups, [9-5](#)
 - cluster resource group type, [9-11](#)
 - creating, [9-12](#)
 - dependencies, [9-5](#)
 - local resource group type, [9-11](#)
 - resource groups (*continued*)
 - privileges, [9-5](#)
 - resource group types, [9-11](#)
- resource permissions
 - changing, [9-42](#)
- resource type
 - cluster_resource, [9-13](#)
 - defined, [9-13](#)
 - generic_application, [9-13](#)
 - local_resource, [9-13](#)
- RESOURCE_LIST
 - resource group attribute, [B-20](#)
- resource_name parameter
 - supplying to CRSCTL commands, [E-153](#)
- resources
 - action scripts, [9-20](#)
 - adding, [9-36](#)
 - with Oracle Enterprise Manager, [9-41](#)
 - adding to a server pool, [9-37](#)
 - adding using a server-specific deployment, [9-38](#)
 - creating with the generic_application resource type, [9-39](#)
 - critical, [9-5](#)
 - defined, [9-2](#)
 - non-critical, [9-5](#)
 - placement, [9-32](#)
 - registering, [9-22](#), [9-44](#)
 - registering an application as, [9-33](#)
 - relocating, [9-45](#)
 - resource group, [9-2](#)
- REST APIs for Oracle Clusterware, [K-1](#)
- RESTART_ATTEMPTS
 - resource attribute, [B-11](#)
- RESTART_COUNT
 - resource attribute, [B-20](#)
- restarts
 - managing automatic, [9-47](#)
- restoring
 - OCR, [6-19](#)
- restricting service registration, [2-29](#)
- RHPCTL, [F-2](#)
 - authentication options, [5-11](#)
 - collect osconfig, [F-50](#)
 - command syntax, [F-1](#)
 - commands, [F-50–F-52](#), [F-61](#), [F-62](#)
 - add client, [F-4](#)
 - add credentials, [F-11](#)
 - add database, [F-12](#)
 - add image, [F-35](#)
 - add imagetype, [F-44](#)
 - add role, [F-53](#)
 - add series, [F-57](#)
 - add useraction, [F-65](#)
 - add workingcopy, [F-68](#)

RHPCTL (*continued*)

commands (*continued*)
 addnode database, [F-14](#)
 addnode gihome, [F-29](#)
 addnode workingcopy, [F-74](#)
 addpdb database, [F-16](#)
 allow image, [F-36](#)
 allow imagetype, [F-45](#)
 delete audit, [F-2](#)
 delete client, [F-5](#)
 delete credentials, [F-12](#)
 delete database, [F-18](#)
 delete image, [F-37](#)
 delete imagetype, [F-45](#)
 delete job, [F-47](#)
 delete role, [F-54](#)
 delete series, [F-57](#)
 delete user, [F-63](#)
 delete useraction, [F-66](#)
 delete workingcopy, [F-75](#)
 deleteimage series, [F-58](#)
 deletenode database, [F-19](#)
 deletenode gihome, [F-31](#)
 deletepdb database, [F-17](#)
 deploy image, [F-37](#)
 disallow image, [F-38](#)
 disallow imagetype, [F-45](#)
 discover client, [F-6](#)
 export client, [F-7](#)
 grant role, [F-55](#)
 import image, [F-39](#)
 insertimage series, [F-58](#)
 instantiate image, [F-40](#)
 modify audit, [F-2](#)
 modify client, [F-7](#)
 modify image, [F-41](#)
 modify imagetype, [F-46](#)
 modify user, [F-63](#)
 modify useraction, [F-66](#)
 move database, [F-21](#)
 move gihome, [F-32](#)
 movepdb database, [F-24](#)
 promote image, [F-43](#)
 query audit, [F-2](#)
 query client, [F-8](#)
 query image, [F-42](#)
 query imagetype, [F-46](#)
 query job, [F-48](#)
 query role, [F-56](#)
 query series, [F-59](#)
 query server, [F-61](#)
 query useraction, [F-68](#)
 query workingcopy, [F-76](#)
 register user, [F-64](#)
 revoke role, [F-56](#)

RHPCTL (*continued*)

commands (*continued*)
 subscribe series, [F-60](#)
 unstantiate image, [F-43](#)
 unsubscribe series, [F-61](#)
 update client, [F-9](#)
 update workingcopy, [F-77](#)
 upgrade database, [F-26](#)
 upgrade gihome, [F-34](#)
 verify client, [F-10](#)
 zdtupgrade database, [F-28](#)
 compare osconfig, [F-50](#)
 disable osconfig, [F-51](#)
 enable osconfig, [F-51](#)
 export server, [F-61](#)
 help, [F-1](#)
 query osconfig, [F-52](#)
 query peerserver, [F-52](#)
 register server, [F-62](#)
 unregister server, [F-62](#)
 role-separated management, [2-1](#)
 configuring, [2-2](#)
 horizontal implementation, [2-1](#)
 vertical implementation, [2-1](#)
 runcluvfy.sh, [A-4](#)

S

scalability
 adding nodes and instances, quick-start
 format, [7-4](#)
 SCAN, [1-9](#), [2-30](#)
 about, [2-28](#)
 See also single client access names (SCAN)
 SCAN listeners, [2-29](#)
 SCRIPT_TIMEOUT
 resource attribute, [B-11](#)
 secondary GNS instance, [2-11](#)
 configuring, [2-13](#)
 removing, [2-14](#)
 sequence ID, [1-15](#)
 server categorization, [3-10](#)
 using attributes to create, [3-9](#)
 server category attributes, [3-16](#)
 server cluster, [1-8](#)
 Server Control Utility (SRVCTL), [1-7](#), [1-21](#)
 See also SRVCTL
 server pools
 attributes, [3-4](#)
 creating, [9-38](#)
 described, [3-3](#)
 Free, [3-3](#)
 Generic, [3-3](#)
 SERVER_CATEGORY, [3-9](#)
 resource attribute, [B-12](#)

- SERVER_POOLS
 - resource attribute, [B-12](#)
- server-centric resource, [9-13](#)
- servers
 - described, [3-12](#)
 - how Oracle Clusterware assigns, [3-8](#)
 - Oracle Clusterware requirements, [1-3](#)
 - states, [3-12](#)
- servers weight-based node eviction, [2-7](#)
- services
 - restricting registration with listeners, [2-29](#)
- setif command
 - OIFCFG command-line interface, [D-5](#)
- shared GNS, [1-8](#), [2-13](#)
 - administering, [2-13](#)
 - generating Client Data file for configuring, [2-16](#)
 - starting, [2-15](#)
 - stopping, [2-15](#)
 - See also GNS
- shared single client access names (SCAN)
 - about, [2-30](#)
 - configuring on dedicated cluster, [2-31](#)
- Single Client Access Name (SCAN)
 - See SCAN
- single client access names (SCAN), shared
 - about, [2-30](#)
 - configuring on dedicated cluster, [2-31](#)
- singleton, [3-3](#)
- slew time synchronization, [1-25](#)
- software requirements, [1-7](#)
- SRVCTL, [1-21](#)
 - command feedback, [G-1](#)
 - commands, [G-76](#), [G-105](#), [G-106](#), [G-125–G-131](#)
 - eval parameter, [G-1](#)
 - add asm, [G-16](#)
 - add asmnetwork, [G-27](#)
 - add cdp, [G-28](#)
 - add cdproxy, [G-32](#)
 - add cvu, [G-38](#), [G-40](#)
 - add exportfs, [G-42](#)
 - add filesystem, [G-49](#)
 - add gns, [G-58](#)
 - add havip, [G-69](#)
 - add ioserver, [G-75](#)
 - add mgmtldb, [G-81](#)
 - add mgmtlsnr, [G-89](#)
 - add mountfs, [G-96](#)
 - add netstorageservice, [G-100](#)
 - add ovmm, [G-104](#)
 - add qosmsserver, [G-107](#)
 - add rhpclient, [G-112](#)
 - add rhpsrvr, [G-118](#)
 - add vm, [G-124](#)
- SRVCTL (*continued*)
 - commands (*continued*)
 - check vm, [G-125](#)
 - config asm, [G-17](#)
 - config asmnetwork, [G-27](#)
 - config cdp, [G-29](#)
 - config cdproxy, [G-33](#)
 - config cvu, [G-38](#)
 - config exportfs, [G-43](#)
 - config filesystem, [G-51](#)
 - config gns, [G-60](#)
 - config havip, [G-70](#)
 - config mgmtldb, [G-81](#)
 - config mgmtlsnr, [G-90](#)
 - config mountfs, [G-96](#)
 - config netstorageservice, [G-101](#)
 - config rhpclient, [G-113](#)
 - config rhpsrvr, [G-119](#)
 - disable asm, [G-18](#)
 - disable cdp, [G-29](#)
 - disable cdproxy, [G-33](#)
 - disable cvu, [G-38](#)
 - disable exportfs, [G-44](#)
 - disable filesystem, [G-52](#)
 - disable gns, [G-61](#)
 - disable havip, [G-70](#)
 - disable ioserver, [G-76](#)
 - disable mgmtldb, [G-82](#)
 - disable mgmtlsnr, [G-90](#)
 - disable mountfs, [G-97](#)
 - disable netstorageservice, [G-101](#)
 - disable qosmsserver, [G-108](#)
 - disable rhpclient, [G-113](#)
 - disable rhpsrvr, [G-119](#)
 - enable asm, [G-19](#)
 - enable cdp, [G-30](#)
 - enable cdproxy, [G-34](#)
 - enable cvu, [G-39](#)
 - enable exportfs, [G-44](#)
 - enable filesystem, [G-52](#)
 - enable gns, [G-61](#)
 - enable havip, [G-71](#)
 - enable ioserver, [G-77](#)
 - enable mgmtldb, [G-82](#)
 - enable mgmtlsnr, [G-91](#)
 - enable mountfs, [G-97](#)
 - enable netstorageservice, [G-102](#)
 - enable qosmsserver, [G-108](#)
 - enable rhpclient, [G-114](#)
 - enable rhpsrvr, [G-120](#)
 - export gns, [G-62](#)
 - getenv asm, [G-19](#)
 - getenv ioserver, [G-77](#)
 - getenv mgmtldb, [G-83](#)
 - getenv mgmtlsnr, [G-91](#)

SRVCTL (continued)

commands (continued)
 help, [G-4](#)
 modify asm, [G-20](#)
 modify asmnetwork, [G-28](#)
 modify cdp, [G-30](#)
 modify cdpproxy, [G-34](#)
 modify cvu, [G-39](#)
 modify exportfs, [G-45](#)
 modify filesystem, [G-53](#)
 modify gns, [G-63](#)
 modify havip, [G-71](#)
 modify ioserver, [G-77](#)
 modify mgmtdb, [G-83](#)
 modify mgmtlsnr, [G-92](#)
 modify mountfs, [G-98](#)
 modify qosmsserver, [G-109](#)
 modify rhpclient, [G-114](#)
 modify rhpserver, [G-120](#)
 predict asm, [G-21](#)
 predict filesystem, [G-54](#)
 predict qosmsserver, [G-109](#)
 relocate asm, [G-21](#)
 relocate cdp, [G-30](#)
 relocate cdpproxy, [G-35](#)
 relocate cvu, [G-40](#)
 relocate gns, [G-64](#)
 relocate havip, [G-72](#)
 relocate ioserver, [G-78](#)
 relocate mgmtdb, [G-84](#)
 relocate qosmsserver, [G-109](#)
 relocate rhpclient, [G-115](#)
 relocate rhpserver, [G-121](#)
 remove asm, [G-22](#)
 remove asmnetwork, [G-28](#)
 remove cdp, [G-31](#)
 remove cdpproxy, [G-35](#)
 remove exportfs, [G-46](#)
 remove filesystem, [G-55](#)
 remove gns, [G-65](#)
 remove havip, [G-73](#)
 remove ioserver, [G-78](#)
 remove mgmtdb, [G-85](#)
 remove mgmtlsnr, [G-92](#)
 remove mountfs, [G-99](#)
 remove netstorageservice, [G-102](#)
 remove qosmsserver, [G-110](#)
 remove rhpclient, [G-116](#)
 remove rhpserver, [G-122](#)
 setenv asm, [G-23](#)
 setenv ioserver, [G-79](#)
 setenv mgmtdb, [G-86](#)
 setenv mgmtlsnr, [G-93](#)
 start asm, [G-24](#)
 start cdp, [G-31](#)

SRVCTL (continued)

commands (continued)
 start cdpproxy, [G-36](#), [G-37](#)
 start cvu, [G-41](#)
 start exportfs, [G-47](#)
 start filesystem, [G-55](#)
 start gns, [G-65](#)
 start havip, [G-73](#)
 start ioserver, [G-79](#)
 start mgmtdb, [G-86](#), [G-94](#)
 start mountfs, [G-99](#)
 start netstorageservice, [G-103](#)
 start qosmsserver, [G-110](#)
 start rhpclient, [G-116](#)
 start rhpserver, [G-122](#)
 status asm, [G-25](#)
 status cdp, [G-32](#)
 status cdpproxy, [G-37](#)
 status cvu, [G-41](#)
 status exportfs, [G-47](#)
 status filesystem, [G-56](#)
 status gns, [G-66](#)
 status havip, [G-74](#)
 status ioserver, [G-79](#)
 status mgmtdb, [G-87](#)
 status mgmtlsnr, [G-94](#)
 status mountfs, [G-100](#)
 status netstorageservice, [G-103](#)
 status qosmsserver, [G-111](#)
 status rhpclient, [G-117](#)
 status rhpserver, [G-123](#)
 stop asm, [G-26](#)
 stop cdp, [G-32](#)
 stop cvu, [G-42](#)
 stop database, [G-88](#)
 stop exportfs, [G-48](#)
 stop filesystem, [G-57](#)
 stop gns, [G-66](#)
 stop havip, [G-74](#)
 stop ioserver, [G-80](#)
 stop mgmtlsnr, [G-95](#)
 stop mountfs, [G-100](#)
 stop netstorageservice, [G-104](#)
 stop qosmsserver, [G-111](#)
 stop rhpclient, [G-117](#)
 stop rhpserver, [G-123](#)
 unsetenv asm, [G-26](#)
 unsetenv ioserver, [G-81](#)
 unsetenv mgmtdb, [G-88](#)
 unsetenv mgmtlsnr, [G-95](#)
 update gns, [G-67](#)
 concurrent commands, [G-1](#)
 config ioserver, [G-76](#)
 config ovmm, [G-105](#)
 config qosmsserver, [G-107](#)

SRVCTL (*continued*)

- config vm, [G-125](#)
- deprecated commands and options, [G-6](#)
- deprecated commands and parameters, [G-13](#)
- difference between SRVCTL and CRSCTL, [G-5](#)
- disable vm, [G-126](#)
- enable vm, [G-127](#)
- modify ovmm, [G-106](#)
- modify vm, [G-127](#)
- overview and concepts, [1-21](#)
- relocate vm, [G-128](#)
- remove ovmm, [G-106](#)
- remove vm, [G-129](#)
- single-character parameters, [G-6](#)
- specifying a continuation of command line entries, [G-1](#)
- start vm, [G-129](#)
- status vm, [G-130](#)
- stop vm, [G-131](#)
- stopping active commands, [G-5](#)
- using comma-delimited lists, [G-1](#)

SRVCTL command reference, [G-1](#)

srvctl stop nodeapps command, [2-32](#)

standard operating environment, [5-27](#)

start effort evaluation

- described, [9-32](#)

START_CONCURRENCY

- resource attribute, [B-13](#)

START_DEPENDENCIES

- resource attribute, [B-13](#)

START_TIMEOUT

- resource attribute, [B-16](#)

starting the OIFCFG interface, [D-1](#)

STATE_CHANGE_EVENT_TEMPLATE

- resource attribute, [B-21](#)

STATE_DETAILS

- resource attribute, [B-21](#)

Stateless Address Autoconfiguration Protocol

- See [autoconfig](#)

step time synchronization, [1-25](#)

STOP_CONCURRENCY

- resource attribute, [B-16](#)

STOP_DEPENDENCIES

- resource attribute, [B-16](#)

STOP_TIMEOUT

- resource attribute, [B-17](#)

subnet

- changing, [2-37](#)
- configuring for VIP address, [1-8](#)

syntax

- OCRDUMP utility, [J-16](#)

SYSRAC administrative privilege, [9-16](#)

SYSTEM.language key

- output, [J-17](#)

SYSTEM.version key

- output, [J-17](#)

T

TARGET

- resource attribute, [B-21](#)

TARGET_SERVER

- resource attribute, [B-21](#)

thread safety, [1-12](#)

trace files, [L-12](#)

tracing

- enabling for Oracle Clusterware, [E-149](#)

transport layer security, [1-15](#)

troubleshooting

- OCR, [J-11](#)

TYPE

- resource attribute, [B-21](#)

U

uniform, [3-3](#)

upgrade

- migrating storage after, [6-2](#), [6-4](#)
- out-of-place, [1-17](#)

upgrades

- and SCANS, [2-32](#)

UPTIME_THRESHOLD

- resource attribute, [B-18](#)

USE_STICKINESS

- resource attribute, [B-19](#)

USER_WORKLOAD

- resource attribute, [B-18](#)

V

valid node checking, [2-29](#)

versions

- compatibility for Oracle Clusterware, Oracle ASM, and Oracle Database software, [1-16](#)

VIP, [1-8](#), [9-33](#)

- adding
 - with Oracle Enterprise Manager, [9-35](#)
- address
 - changing, [2-32](#)
 - defining for applications, [1-24](#)
 - requirements, [1-8](#)

virtual internet protocol address (VIP), [9-33](#)

virtual IP

- See [VIP](#)

virtual machine, [9-2](#)

- black-box, [9-2](#)

virtual machine (*continued*)
 manager, [9-2](#)
 server, [9-2](#)
voting file location
 locating, [6-31](#)
voting files, [1-7](#)
 adding, [6-30](#)
 adding to non-Oracle ASM storage, [6-30](#)
 adding to Oracle ASM, [6-30](#)
 administering, [6-1](#)
 backing up, [6-28](#)
 deleting, [6-30](#), [6-31](#)
 file universal identifier (FUID)
 obtaining, [6-30](#)
 managing, [6-25](#)
 migrating, [6-30](#)
 migrating to Oracle ASM, [6-30](#), [6-31](#)
 modifying not stored on Oracle ASM, [6-30](#)
 modifying stored on Oracle ASM, [6-30](#)
 replacing in non-Oracle ASM storage, [6-30](#)
 restoring, [6-28](#)
 storing on Oracle ASM, [6-26](#)

W

weak start dependency, [9-30](#)
 modifiers, [B-15](#)
What-If APIs
 See command evaluation APIs
Why-If APIs
 See reasoned command evaluation APIs
Windows systems
 services for clusterware, [1-15](#)
working copy
 creating on the Fleet Patching and
 Provisioning client, [5-21](#)
 creating on the Fleet Patching and
 Provisioning Server, [5-21](#)

Z

zero-downtime database upgrade, [5-1](#)
zero-downtime upgrade, [5-49](#)
 customizing, [5-54](#)
 running with Oracle Data Guard, [5-52](#)
 running with Oracle GoldenGate, [5-50](#)