Oracle® Clusterware

Clusterware Administration and Deployment Guide

23ai
F46761-02
October 2024

Oracle Clusterware Clusterware Administration and Deployment Guide, 23ai

F46761-02

Copyright © 2007, 2024, Oracle and/or its affiliates.

Primary Authors: Subhash Chandra, Eric Belden

Contributors: David Jimenez Alvarez, Troy Anthony, Ram Avudaiappan, Mark Bauer, Devang Bagaria, Suman Bezawada, Gajanan Bhat, Burt Clouse, Ian Cookson, Jonathan Creighton, Mark Fuller, Apostolos Giannakidis, Angad Gokakkar, Donald Graves, 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, Lars Mortimer, Anil Nair, Siva Nandan, Philip Newlan, Gia-Khanh Nguyen, Balaji Pagadala, Srinivas Poovala, Kevin Reardon, Dipak Saggi, Luv Sharma, Duane Smith, Janet Stern, Su Tang, Manasvi Nishith Vohra, James Warnes, Douglas Williams

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, software documentation, data (as defined in the Federal Acquisition Regulation), 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 embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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 xxii
Documentation Accessibility xxii
Diversity and Inclusion xxii
Set Up Java Access Bridge to Implement Java Accessibility xxiii
Command Syntax xxiii
Conventions xxiv

1 Introduction to Oracle Clusterware

Overview of Oracle Clusterware 1-1
Understanding System Requirements for Oracle Clusterware 1-3
- Oracle Clusterware Hardware Concepts and Requirements 1-3
- Oracle Clusterware Operating System Concepts and Requirements 1-4
- Oracle Clusterware Software Concepts and Requirements 1-4
- Oracle Clusterware Network Configuration Concepts 1-5
  - Single Client Access Name (SCAN) 1-6
  - Manual Addresses Configuration 1-6

Overview of Oracle Clusterware Platform-Specific Software Components 1-7
- The Oracle Clusterware Technology Stack 1-7
  - The Cluster Ready Services Technology Stack 1-7
  - The Oracle High Availability Services Technology Stack 1-8
- Oracle Clusterware Processes on Windows Systems 1-12

High Availability Options for Oracle Database 1-13
Oracle Clusterware Version Compatibility 1-14
Overview of Upgrading and Patching Oracle Clusterware 1-14
Overview of Managing Oracle Clusterware Environments 1-15
Overview of Command Evaluation 1-16
Overview of Cloning and Extending Oracle Clusterware in Grid Environments 1-18
Overview of the Oracle Clusterware High Availability Framework and APIs 1-18

2 Oracle Clusterware Configuration and Administration

Role-Separated Management 2-1

Configuring Role Separation 2-2
Configuring Oracle Grid Infrastructure Using Grid Setup Wizard 2-3
Configuring a Single Node 2-4
Configuring Multiple Nodes 2-4
Upgrading Oracle Grid Infrastructure 2-5
Running the Configuration Wizard in Silent Mode 2-5
Moving and Patching an Oracle Grid Infrastructure Home 2-6
Server Weight-Based Node Eviction 2-6
Overview of Grid Naming Service 2-7
Network Administration Tasks for GNS and GNS Virtual IP Address 2-7
Understanding Grid Naming Service Configuration Options 2-8
Automatic Configuration Option for Addresses 2-8
Static Configuration Option for Addresses 2-9
Administering Grid Naming Service 2-9
Starting and Stopping GNS with SRVCTL 2-9
Changing the GNS Subdomain when Moving from IPv4 to IPv6 Network 2-9
Rolling Conversion from DNS to GNS Cluster Name Resolution 2-10
Node Failure Isolation 2-11
Server Hardware Configuration for IPMI 2-12
Post-installation Configuration of IPMI-based Failure Isolation Using CRSCTL 2-13
IPMI Post-installation Configuration with Oracle Clusterware 2-13
Modifying IPMI Configuration Using CRSCTL 2-14
Removing IPMI Configuration Using CRSCTL 2-15
Understanding Network Addresses on Manually Configured Networks 2-16
Understanding Network Address Configuration Requirements 2-16
About IPv6 Address Formats 2-17
Name Resolution and the Network Resource Address Type 2-17
Understanding SCAN Addresses and Client Service Connections 2-18
SCAN Listeners and Service Registration Restriction With Valid Node Checking 2-19
Configuring Shared Single Client Access Names 2-20
About Configuring Shared Single Client Access Names 2-20
Configuring the Use of Shared SCAN 2-20
Changing Network Addresses on Manually Configured Systems 2-21
Changing the Virtual IP Addresses Using SRVCTL 2-22
Changing Oracle Clusterware Private Network Configuration 2-24
About Private Networks and Network Interfaces 2-24
Redundant Interconnect Usage 2-24
Consequences of Changing Interface Names Using OIFCFG 2-25
Changing a Network Interface 2-26
Creating a Network Using SRVCTL 2-28
Network Address Configuration in a Cluster 2-29
Changing Static IPv4 Addresses To Static IPv6 Addresses Using SRVCTL 2-29

Changing Dynamic IPv4 Addresses To Dynamic IPv6 Addresses Using SRVCTL 2-31
Changing an IPv4 Network to an IPv4 and IPv6 Network 2-33
Transitioning from IPv4 to IPv6 Networks for VIP Addresses Using SRVCTL 2-33

3 Oracle Extended Clusters

About Oracle Extended Clusters 3-1
Converting to Oracle Extended Cluster 3-1

4 Oracle Clusterware to Manage Generic Applications

About Oracle Grid Infrastructure to Manage Generic Applications 4-1
Enabling Oracle RAC Databases on a Generic Application Cluster 4-1

5 Managing Oracle Cluster Registry and Voting Files

Managing Oracle Cluster Registry and Oracle Local Registry 5-1
Migrating Oracle Cluster Registry to Oracle Automatic Storage Management 5-2
Migrating Oracle Cluster Registry from Oracle ASM to Other Types of Storage 5-3
Adding, Replacing, Repairing, and Removing Oracle Cluster Registry Locations 5-4
Adding an Oracle Cluster Registry Location 5-5
Removing an Oracle Cluster Registry Location 5-6
Replacing an Oracle Cluster Registry Location 5-6
Repairing an Oracle Cluster Registry Configuration on a Local Node 5-7
Overriding the Oracle Cluster Registry Data Loss Protection Mechanism 5-8
Backing Up Oracle Cluster Registry 5-9
Listing Backup Files 5-10
Changing Backup Location 5-10
Restoring Oracle Cluster Registry 5-11
How to Check Oracle Cluster Registry Issues 5-11
Restoring the Oracle Cluster Registry on Linux or Unix Systems 5-11
Restoring the Oracle Cluster Registry on Windows Systems 5-14
Restoring the Oracle Cluster Registry in an Oracle Restart Environment 5-15
Diagnosing Oracle Cluster Registry Problems 5-16
Administering Oracle Cluster Registry with Export and Import Commands 5-16
Importing Oracle Cluster Registry Content on Linux or Unix Systems 5-18
Importing Oracle Cluster Registry Content on Windows Systems 5-20
Oracle Local Registry 5-21
Managing Voting Files 5-23
Storing Voting Files on Oracle ASM 5-24
Backing Up Voting Files 5-25
Restoring Voting Files 5-26

Adding, Deleting, or Migrating Voting Files 5-27
Modifying Voting Files that are Stored in Oracle ASM 5-27
Modifying Voting Files that are Not Stored on Oracle ASM 5-27
Migrating Voting Files to Oracle ASM 5-28
Verifying the Voting File Location 5-28

6 Adding and Deleting Cluster Nodes

Prerequisite Steps for Adding Cluster Nodes 6-1
Adding and Deleting Cluster Nodes on Linux and UNIX Systems 6-2
Adding a Cluster Node on Linux and UNIX Systems 6-3
Deleting a Cluster Node on Linux and UNIX Systems 6-7
Adding and Deleting Cluster Nodes on Windows Systems 6-9
Adding a Node to a Cluster on Windows Systems 6-10
Deleting a Cluster Node on Windows Systems 6-12

7 Cloning Oracle Clusterware

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

8 Making Applications Highly Available Using Oracle Clusterware

Oracle Clusterware Resources and Agents 8-1
Oracle Clusterware Resources 8-1
Virtual Machine Resources 8-2
Resource Groups 8-5
Oracle Clusterware Resource Types 8-12

Agents in Oracle Clusterware 8-15
Oracle Clusterware Built-in Agents 8-18
Action Scripts 8-19
Building an Agent 8-19
Building and Deploying C and C++ Agents 8-20
Registering a Resource in Oracle Clusterware 8-21
Overview of Using Oracle Clusterware to Enable High Availability 8-22
Resource Attributes 8-23
Resource States 8-24
Resource Dependencies 8-24
Start Dependencies 8-25
Stop Dependencies 8-29
Resource Placement 8-30
Registering an Application as a Resource 8-31
Creating an Application VIP Managed by Oracle Clusterware 8-32
Adding an Application VIP with Oracle Enterprise Manager 8-33
Adding User-Defined Resources 8-34
Creating Resources that Use the generic_application Resource Type 8-34
Adding Resources Using Oracle Enterprise Manager 8-36
Changing Resource Permissions 8-37
Application Placement Policies 8-37
Unregistering Applications and Application Resources 8-38
Managing Resources 8-38
Registering Application Resources 8-39
Starting Application Resources 8-39
Relocating Applications and Application Resources 8-40
Stopping Applications and Application Resources 8-40
Displaying Clusterware Application and Application Resource Status Information 8-41
Managing Automatic Restart of Oracle Clusterware Resources 8-41
Preventing Automatic Restarts of Oracle Clusterware Resources 8-42
Automatically Manage Restart Attempts Counter for Oracle Clusterware Resources 8-42

A Configuration Verification Utility Reference

About Configuration Verification Utility A-1
Overview of CVU A-2
Downloading Configuration Verification Utility A-3
CVU Operational Notes A-4
CVU Installation Requirements A-4
CVU Usage Information A-4
CVU Configuration File A-6
Privileges and Security A-7

Using CVU Help A-7
Deprecated and Desupported CLUVFY Commands A-8
Special CVU Topics A-8
Generating Fixup Scripts A-9
Periodic Cluster Health Check A-9
Enabling or Disabling Automatic Self Correction A-9
Using CVU to Determine if Installation Prerequisites are Complete A-10
Entry and Exit Criteria A-11
Verbose Mode and UNKNOWN Output A-11
CVU Node List Shortcuts A-11
Configuration Verification Utility Command Reference A-12
cluvfy comp acfs A-12
cluvfy comp admprv A-13
cluvfy comp asm A-15
cluvfy comp baseline A-17
cluvfy comp clocksync A-19
cluvfy comp clu A-20
cluvfy comp clumgr A-21
cluvfy comp clusterware A-22
cluvfy comp containerhost A-23
cluvfy comp crs A-24
cluvfy comp dhcp A-25
cluvfy comp dns A-27
cluvfy comp freespace A-29
cluvfy comp gns A-29
cluvfy comp gpnp A-31
cluvfy comp ha A-31
cluvfy comp healthcheck A-32
cluvfy comp nodeapp A-33
cluvfy comp nodecon A-34
cluvfy comp nodereach A-36
cluvfy comp ocr A-37
cluvfy comp ohasd A-39
cluvfy comp olr A-40
cluvfy comp peer A-42
cluvfy comp scan A-43
cluvfy comp software A-45
cluvfy comp space A-46
cluvfy comp ssa A-47
cluvfy comp sys A-50
cluvfy comp vdisk A-52
cluvfy comp vital A-53

cluvfy stage acfscfg A-53
cluvfy stage appcluster A-55
cluvfy stage cfs A-55
cluvfy stage crsinst A-56
cluvfy stage dbcfg A-60
cluvfy stage dbinst A-61
cluvfy stage hacfg A-63
cluvfy stage hwos A-65
cluvfy stage nodeadd A-65
cluvfy stage nodedel A-67
cluvfy stage osupg A-68
cluvfy stage patch A-69
Troubleshooting and Diagnostic Output for CVU A-70
Enabling Tracing A-70
Known Issues for the Configuration Verification Utility A-70
Database Versions Supported by Configuration Verification Utility A-70
Linux Shared Storage Accessibility (ssa) Check Reports Limitations A-70
Shared Disk Discovery on Red Hat Linux A-71

B Oracle Clusterware Resource Reference

Resource Attributes B-1
Configurable Resource Attributes B-2
ACL B-2
ACTION_SCRIPT B-3
ACTION_TIMEOUT B-3
ACTIONS B-3
ACTIVE_PLACEMENT B-4
AGENT_FILENAME B-4
ALERT_TEMPLATE B-4
AUTO_START B-4
CARDINALITY B-5
CARDINALITY_ID B-5
CHECK_INTERVAL B-5
CHECK_TIMEOUT B-5
CLEAN_TIMEOUT B-5
CRITICAL_RESOURCES B-6
DELETE_TIMEOUT B-6
DESCRIPTION B-6
ENABLED B-6
FAILURE_INTERVAL B-6
FAILURE_THRESHOLD B-7

HOSTING_MEMBERS B-7
INSTANCE_FAILOVER B-7
INTERMEDIATE_TIMEOUT B-8
LOAD B-8
MODIFY_TIMEOUT B-8
NAME B-8
OFFLINE_CHECK_INTERVAL B-8
ONLINE_RELOCATION_TIMEOUT B-9
PLACEMENT B-9
RANK B-9
RELOCATE_KIND B-9
RELOCATE_BY_DEPENDENCY B-10
RESTART_ATTEMPTS B-10
SCRIPT_TIMEOUT B-10
SERVER_CATEGORY B-10
START_CONCURRENCY B-11
START_DEPENDENCIES B-11
START_TIMEOUT B-13
STOP_CONCURRENCY B-14
STOP_DEPENDENCIES B-14
STOP_TIMEOUT B-15
UPTIME_THRESHOLD B-15
USER_WORKLOAD B-16
USE_STICKINESS B-16
WORKLOAD_CPU B-16
Read-Only Resource Attributes B-17
ACTION_FAILURE_EVENT_TEMPLATE B-17
INSTANCE_COUNT B-17
INTERNAL_STATE B-17
LAST_SERVER B-17
LAST_STATE_CHANGE B-17
PROFILE_CHANGE_EVENT_TEMPLATE B-17
RESOURCE_LIST B-17
RESTART_COUNT B-18
STATE B-18
STATE_CHANGE_EVENT_TEMPLATE B-18
STATE_DETAILS B-18
TARGET B-18
TARGET_SERVER B-18
TYPE B-19
Examples of Action Scripts for Third-party Applications B-19

C OLSNODES Command Reference

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

D Oracle Interface Configuration Tool (OIFCFG) Command Reference

Starting the OIFCFG Command-Line Interface D-1
Summary of the OIFCFG Usage D-2
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-6
crsctl add resource E-7
crsctl delete resource E-10
crsctl eval add resource E-11
crsctl eval fail resource E-14
crsctl eval relocate resource E-15
crsctl eval modify resource E-16
crsctl eval start resource E-17
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-30
crsctl add resourcegroup E-32
crsctl check resourcegroup E-33
crsctl delete resourcegroup E-33
crsctl eval add resourcegroup E-34
crsctl eval fail resourcegroup E-35
crsctl eval relocate resourcegroup E-35
crsctl eval start resourcegroup E-36
crsctl eval stop resourcegroup E-37
crsctl export resourcegroup E-37
crsctl modify resourcegroup E-37
crsctl relocate resourcegroup E-39
crsctl restart resourcegroup E-40
crsctl start resourcegroup E-41
crsctl status resourcegroup E-42
crsctl stop resourcegroup E-43
crsctl add resourcegrouptype E-44
crsctl delete resourcegrouptype E-45
crsctl modify resourcegrouptype E-46
crsctl get tracefileopts E-46
crsctl set tracefileopts E-47
crsctl add type E-47
crsctl delete type E-49
crsctl getperm type E-50
crsctl modify type E-51
crsctl setperm type E-52
crsctl status type E-53
Oracle RAC Environment CRSCTL Commands E-54
crsctl request action E-54
crsctl add category E-55
crsctl delete category E-56
crsctl modify category E-56
crsctl status category E-57
crsctl check cluster E-59
crsctl start cluster E-60
crsctl stop cluster E-60
crsctl get cluster class E-61
crsctl get cluster configuration E-61
crsctl set cluster disabledtlsciphersuite E-61
crsctl get cluster extended E-62

crsctl get cluster mode E-62
crsctl set cluster mode E-62
crsctl get cluster name E-63
crsctl add cluster site E-63
crsctl delete cluster site E-63
crsctl modify cluster site E-64
crsctl query cluster site E-64
crsctl get cluster tlsciphersuite E-65
crsctl get cluster type E-65
crsctl set cluster type E-66
crsctl get cpu equivalency E-66
crsctl set cpu equivalency E-66
crsctl get restricted placement E-67
crsctl set restricted placement E-67
crsctl check crs E-68
crsctl config crs E-68
crsctl disable crs E-68
crsctl enable crs E-69
crsctl start crs E-69
crsctl stop crs E-70
crsctl query crs activeversion E-71
crsctl query crs autostart E-72
crsctl set crs autostart E-72
crsctl query crs releasepatch E-73
crsctl query crs releaseversion E-73
crsctl query crs site E-74
crsctl query crs softwarepatch E-74
crsctl query crs softwareversion E-75
crsctl get css E-75
crsctl pin css E-76
crsctl set css E-76
crsctl unpin css E-76
crsctl unset css E-77
crsctl get ipmi binaryloc E-78
crsctl set ipmi binaryloc E-78
crsctl get css ipmiaddr E-78
crsctl set css ipmiaddr E-79
crsctl set css ipmiadmin E-79
crsctl query css ipmiconfig E-80
crsctl unset css ipmiconfig E-80
crsctl query css ipmidevice E-81
crsctl get css noautorestart E-81

crsctl set css noautorestart E-82
crsctl add css votedisk E-82
crsctl delete css votedisk E-83
crsctl query css votedisk E-83
crsctl discover dhcp E-84
crsctl get clientid dhcp E-84
crsctl release dhcp E-85
crsctl request dhcp E-86
crsctl replace discoverystring E-86
crsctl query dns E-87
crsctl start ip E-87
crsctl status ip E-88
crsctl stop ip E-88
crsctl lsmodules E-89
crsctl delete node E-90
crsctl check resource E-90
crsctl relocate resource E-91
crsctl get resource use E-93
crsctl set resource use E-93
crsctl start rollingpatch E-94
crsctl stop rollingpatch E-94
crsctl start rollingupgrade E-95
crsctl eval add server E-95
crsctl eval delete server E-97
crsctl modify server E-98
crsctl status server E-99
crsctl get server css_critical E-100
crsctl set server css_critical E-100
crsctl get server label E-100
crsctl set server label E-101
crsctl query socket udp E-101
crsctl start testdns E-102
crsctl status testdns E-103
crsctl stop testdns E-103
crsctl replace votedisk E-104
crsctl add wallet E-105
crsctl delete wallet E-105
crsctl modify wallet E-106
crsctl query wallet E-107
Oracle Restart Environment CRSCTL Commands E-108
crsctl check has E-108
crsctl config has E-108

crsctl disable has E-109
crsctl enable has E-109
crsctl query has releaseversion E-109
crsctl query has softwareversion E-110
crsctl start has E-110
crsctl stop has E-111
Troubleshooting and Diagnostic Output E-111
Dynamic Debugging Using crsctl set log E-111
Component Level Debugging E-112
Enabling Debugging for Oracle Clusterware Modules E-112
Enabling Debugging for Oracle Clusterware Resources E-115

F Server Control (SRVCTL) Command Reference

SRVCTL Usage Information F-1
Specifying Command Parameters as Keywords Instead of Single Letters F-2
Character Set and Case Sensitivity of SRVCTL Object Values F-2
Using SRVCTL Help F-3
SRVCTL Privileges and Security F-4
Additional SRVCTL Topics F-4
Deprecated SRVCTL Subprograms or Commands F-5
Single Character Parameters Deprecated for all SRVCTL Commands F-5
Miscellaneous SRVCTL Commands and Parameters F-11
SRVCTL Command Reference F-12
srvctl config all F-14
srvctl add asm F-15
srvctl config asm F-16
srvctl disable asm F-17
srvctl enable asm F-18
srvctl getenv asm F-18
srvctl modify asm F-19
srvctl predict asm F-20
srvctl relocate asm F-20
srvctl remove asm F-21
srvctl setenv asm F-22
srvctl start asm F-22
srvctl status asm F-23
srvctl stop asm F-24
srvctl unsetenv asm F-25
srvctl add asmnetwork F-26
srvctl config asmnetwork F-26
srvctl modify asmnetwork F-26

srvctl remove asmnetwork F-27
srvctl add cdp F-27
srvctl config cdp F-28
srvctl disable cdp F-28
srvctl enable cdp F-28
srvctl modify cdp F-29
srvctl relocate cdp F-29
srvctl remove cdp F-30
srvctl start cdp F-30
srvctl status cdp F-30
srvctl stop cdp F-31
srvctl add cvu F-31
srvctl config cvu F-32
srvctl disable cvu F-32
srvctl enable cvu F-32
srvctl modify cvu F-33
srvctl relocate cvu F-33
srvctl remove cvu F-34
srvctl start cvu F-34
srvctl status cvu F-35
srvctl stop cvu F-35
srvctl add exportfs F-36
srvctl config exportfs F-37
srvctl disable exportfs F-37
srvctl enable exportfs F-38
srvctl modify exportfs F-38
srvctl remove exportfs F-39
srvctl start exportfs F-40
srvctl status exportfs F-40
srvctl stop exportfs F-41
srvctl add filesystem F-42
srvctl config filesystem F-44
srvctl disable filesystem F-45
srvctl enable filesystem F-45
srvctl modify filesystem F-46
srvctl predict filesystem F-47
srvctl remove filesystem F-48
srvctl start filesystem F-49
srvctl status filesystem F-50
srvctl stop filesystem F-51
srvctl add gns F-52
srvctl config gns F-53

srvctl disable gns F-54
srvctl enable gns F-55
srvctl modify gns F-55
srvctl relocate gns F-56
srvctl remove gns F-57
srvctl start gns F-57
srvctl status gns F-58
srvctl stop gns F-58
srvctl update gns F-59
srvctl add havip F-61
srvctl config havip F-61
srvctl disable havip F-62
srvctl enable havip F-63
srvctl modify havip F-63
srvctl relocate havip F-64
srvctl remove havip F-64
srvctl start havip F-65
srvctl status havip F-66
srvctl stop havip F-66
srvctl add ioserver F-67
srvctl config ioserver F-67
srvctl disable ioserver F-68
srvctl enable ioserver F-68
srvctl getenv ioserver F-68
srvctl modify ioserver F-69
srvctl relocate ioserver F-69
srvctl remove ioserver F-70
srvctl setenv ioserver F-70
srvctl start ioserver F-70
srvctl status ioserver F-71
srvctl stop ioserver F-72
srvctl unsetenv ioserver F-72
srvctl add mountfs F-72
srvctl config mountfs F-73
srvctl disable mountfs F-73
srvctl enable mountfs F-74
srvctl modify mountfs F-74
srvctl remove mountfs F-75
srvctl start mountfs F-75
srvctl status mountfs F-76
srvctl stop mountfs F-76
srvctl add netstorageservice F-77

srvctl config netstorageservice F-77
srvctl disable netstorageservice F-77
srvctl enable netstorageservice F-78
srvctl remove netstorageservice F-78
srvctl start netstorageservice F-79
srvctl status netstorageservice F-79
srvctl stop netstorageservice F-80
srvctl add rhpclient F-80
srvctl config rhpclient F-82
srvctl disable rhpclient F-82
srvctl enable rhpclient F-83
srvctl modify rhpclient F-83
srvctl relocate rhpclient F-84
srvctl remove rhpclient F-85
srvctl start rhpclient F-85
srvctl status rhpclient F-86
srvctl stop rhpclient F-86
srvctl add rhpserver F-87
srvctl config rhpserver F-89
srvctl disable rhpserver F-90
srvctl enable rhpserver F-90
srvctl modify rhpserver F-91
srvctl relocate rhpserver F-92
srvctl remove rhpserver F-92
srvctl start rhpserver F-93
srvctl status rhpserver F-94
srvctl stop rhpserver F-94
srvctl add vm F-95
srvctl check vm F-96
srvctl config vm F-96
srvctl disable vm F-97
srvctl enable vm F-98
srvctl modify vm F-98
srvctl relocate vm F-100
srvctl remove vm F-100
srvctl start vm F-101
srvctl status vm F-101
srvctl stop vm F-102

G Oracle Clusterware Agent Framework C Application Program Interfaces

Agent Framework Data Types G-1

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

H Oracle Clusterware C Application Program Interfaces

About the Programming Interface (C API) to Oracle Clusterware H-1
Overview H-1
Operational Notes H-2
Deprecated CLSCRS APIs H-8
Changes to Existing CLSCRS APIs H-9

Interactive CLSCRS APIs H-10
Non-Interactive CLSCRS APIs H-11
Command Evaluation APIs H-14
clscrs_whatif_fail_resource H-15
clscrs_whatif_register_resource H-16
clscrs_whatif_start_resource H-17
clscrs_whatif_stop_resource H-17
clscrs_whatif_add_server H-18
clscrs_whatif_delete_server H-19
Server Categorization APIs H-20
clscrs_servercategory_create H-20
clscrs_servercategory_destroy H-20
clscrs_register_servercategory H-21
clscrs_unregister_servercategory H-22
clscrs_get_server_by_category H-22
clscrs_register_server H-23
STAT3 API H-24
clscrs_stat3 H-24
Miscellaneous APIs H-25
clscrs_get_error_details H-25
clscrs_request_action H-26
clscrs_restart_resource H-27

I Oracle Cluster Registry Utility Reference

About OCRCONFIG I-1
OCRCONFIG Command Reference I-2
ocrconfig -add I-2
ocrconfig -backuploc I-2
ocrconfig -copy I-3
ocrconfig -delete I-4
ocrconfig -delbackup I-4
ocrconfig -export I-5
ocrconfig -import I-5
ocrconfig -manualbackup I-6
ocrconfig -overwrite I-6
ocrconfig -repair I-7
ocrconfig -replace I-8
ocrconfig -restore I-8
ocrconfig -showbackup I-9
ocrconfig -showbackuploc I-10
Troubleshooting Oracle Cluster Registry and Diagnostic Output I-10

Troubleshooting Oracle Cluster Registry I-10
Using the OCRCHECK Utility I-11
Syntax I-11
Examples I-11
Using the OCRDUMP Utility to View Oracle Cluster Registry Content I-14
OCRDUMP Utility Syntax and Options I-14
OCRDUMP Utility Examples I-15
Sample OCRDUMP Utility Output I-15

J Troubleshooting Oracle Clusterware

Troubleshooting an Incompatible Fleet Patching and Provisioning Client Resource J-1
Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures J-2
crsctl query calog J-3
crsctl get calog maxsize J-11
crsctl get calog retentiontime J-11
crsctl set calog maxsize J-12
crsctl set calog retentiontime J-12
Oracle Clusterware Diagnostic and Alert Log Data J-13
Diagnostics Collection Script J-15
Storage Split in Oracle Extended Clusters J-17
Rolling Upgrade and Driver Installation Issues J-17
Testing Zone Delegation J-18
Oracle Clusterware Alerts J-19
Alert Messages Using Diagnostic Record Unique IDs J-19

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
- Diversity and Inclusion
- Set Up Java Access Bridge to Implement Java Accessibility
- Command Syntax
- Conventions

Audience

The *Oracle Clusterware Administration and Deployment Guide* provides a description of Oracle Clusterware architecture as well as administrative and deployment topics for database administrators, network administrators, and system administrators.

This guide is also intended for administrators who administer and manage Oracle Real Application Clusters (Oracle RAC) databases.

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also

mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Set Up Java Access Bridge to Implement Java Accessibility

Install Java Access Bridge so that assistive technologies on Microsoft Windows systems can use the Java Accessibility API.

Java Access Bridge is a technology that enables Java applications and applets that implement the Java Accessibility API to be visible to assistive technologies on Microsoft Windows systems.

Refer to *Java Platform, Standard Edition Accessibility Guide* for information about the minimum supported versions of assistive technologies required to use Java Access Bridge. Also refer to this guide to obtain installation and testing instructions, and instructions for how to use Java Access Bridge.

Related Topics

- *Java Platform, Standard Edition Java Accessibility Guide*

Command Syntax

Refer to these command syntax conventions to understand command examples in this guide.

Convention	Description
`$`	Bourne or BASH shell prompt in a command example. Do not enter the prompt as part of the command.
`%`	C Shell prompt in a command example. Do not enter the prompt as part of the command.
`#`	Superuser (root) prompt in a command example. Do not enter the prompt as part of the command.
`monospace`	UNIX command syntax
backslash \	A backslash is the UNIX and Linux command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash: `dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \  count=10000`
braces { }	Braces indicate required items: `.DEFINE {macro1}`
brackets [ ]	Brackets indicate optional items: `cvtcrt termname [outfile]`

Convention	Description
ellipses ...	Ellipses indicate an arbitrary number of similar items: `CHKVAL fieldname value1 value2 ... valueN`
italic	Italic type indicates a variable. Substitute a value for the variable: *`library_name`*
vertical line |	A vertical line indicates a choice within braces or brackets: `FILE filesize [K|M]`

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.
italic	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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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 12*c* 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).

Clusterware Architectures

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.

Oracle Clusterware provides one deployment architecture choice for new clusters during the installation process.

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

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]

> **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

[1] Oracle Clusterware supports up to 100 nodes in a cluster on configurations running Oracle Database 12c release 2 (12.2) and later releases.

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 Oracle*MetaLink*) 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 for Linux*

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 Oracle Database Configuration Assistant (Oracle 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 or later releases (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.

> **Note:**
>
> The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai.
> The highly-available GNS provides the ability to run multiple GNS instances in a multi-cluster environment with different roles. This feature is being deprecated. There is no replacement.

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 cluster environment.

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 for Linux*

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, Pluggable Database (PDB), 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 (`oprocd`), 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.

- **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 11*g* 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. 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 (GPNPD)**: 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, GPNPD, 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 only one process, but an instance. Given Oracle Flex ASM, Oracle ASM does not necessarily run on every cluster node but only some of them.

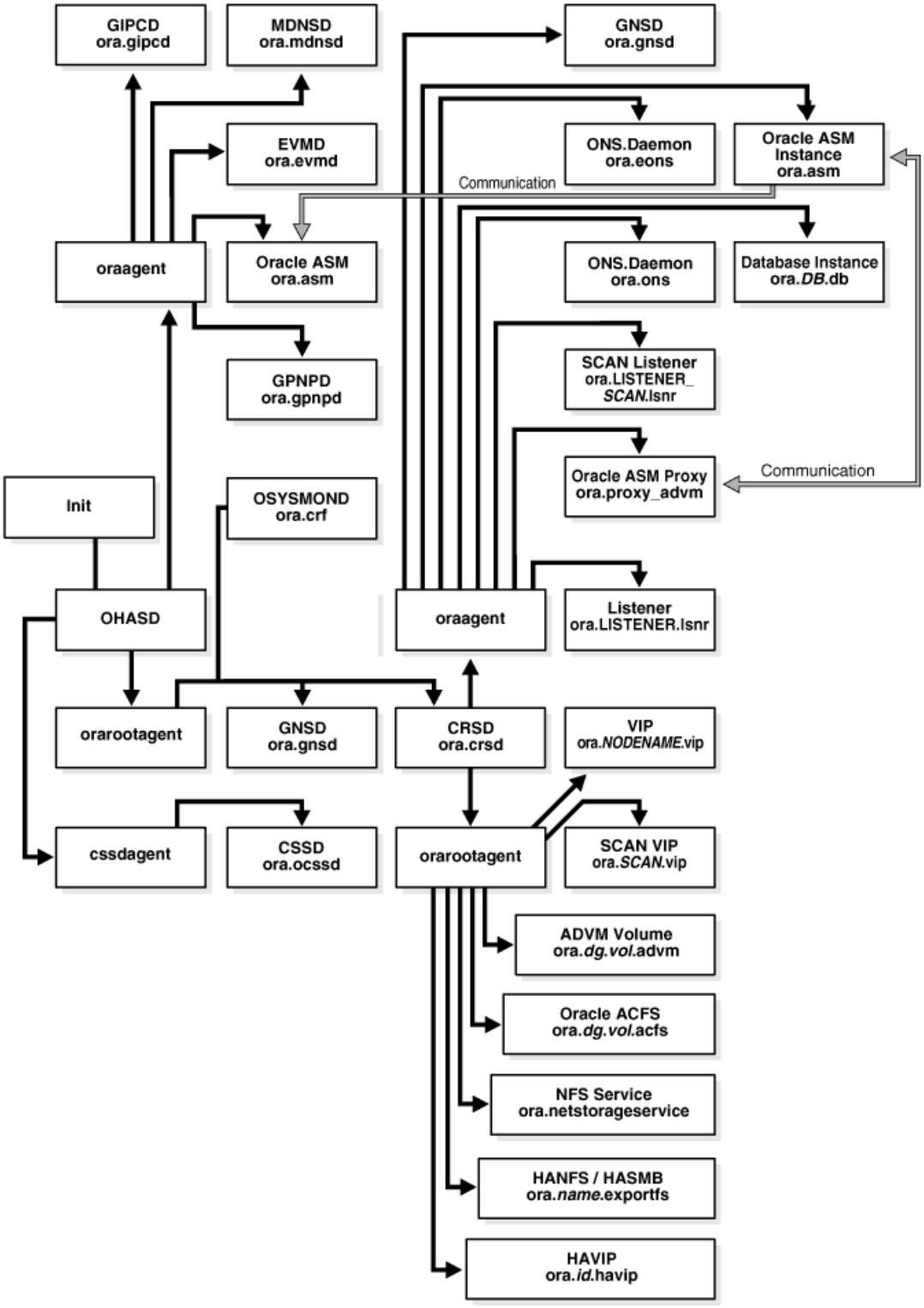
Table 1-1 List of Processes and Services Associated with Oracle Clusterware Components

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`
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 12*c* release 1 (12.1) and later releases	`oraagent.exe`
Oracle High Availability Services	`ohasd.bin` (r)	`ohasd.exe`
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 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.

Starting with Oracle Clusterware 19c, the 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, you could not disable a given cipher suite included in the set 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

High Availability Options for Oracle Database

Review the high availability options available to you for Oracle Database using Standard Edition High Availability, Oracle Restart, Oracle Real Application Clusters (Oracle RAC), and Oracle RAC One Node.

The following is an overview of the high availability options available to you for Oracle Database.

Standard Edition High Availability

- Cluster-based active/passive Oracle Database failover solution
- Designed for single instance Standard Edition Oracle Databases
- Available with Oracle Database 19c release update (RU) 19.7 and later
- Requires Oracle Grid Infrastructure 19c RU 19.7 and later, installed as a Standalone Cluster

Oracle Restart

- Oracle Database instance restart only feature and basis for Oracle Automatic Storage Management (Oracle ASM) for standalone server deployments
- For single instance Oracle Databases
- Requires Oracle Grid Infrastructure for a standalone server (no cluster)

Oracle Real Application Clusters (Oracle RAC) One Node

- Provides a cluster-based active/passive Oracle Database failover and online database relocation solution
- Available for Oracle RAC-enabled Oracle Databases
- Only one instance of an Oracle RAC-enabled Oracle Database is running under normal operations
- Enables relocation of the active instance to another server in the cluster in an online fashion. To relocate the active instance, you can temporarily start a second instance on the destination server, and relocate the workload
- Supports Rolling Upgrades - patch set, database, and operating system
- Supports Application Continuity
- Requires Oracle Grid Infrastructure to be installed as a Standalone Cluster

Oracle Real Application Clusters (Oracle RAC)

- Provides active / active Oracle Database high availability and scalability solution
- Enables multiple servers to perform concurrent transactions on the same Oracle Database
- Provides high availability: a failure of a database instance or server does not interrupt the database service as a whole, because other instances and their servers remain operational
- Supports Rolling Upgrades - patch set, database, and operating system
- Supports Application Continuity
- Requires Oracle Grid Infrastructure to be installed as a Standalone Cluster

Oracle Clusterware Version Compatibility

You can install different releases of Oracle Clusterware 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 23ai supports Oracle Database 23ai, Oracle Database 21c, and Oracle Database 19c.
  - Oracle Database 23ai requires Oracle Clusterware 23ai. For example, If you have Oracle Clusterware 23ai installed as your clusterware, then you can have an Oracle Database 19c single-instance database running on one node, and separate Oracle Real Application Clusters 21c, and Oracle Real Application Clusters 23ai databases also running on the cluster.
- 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.
- Oracle recommends that you do not run different cluster software on the same servers unless they are certified to work together.

> **Note:**
>
> Starting with Oracle Clusterware 21c, the integration of vendor or third party clusterware with Oracle Clusterware is desupported.

Related Topics

- *Oracle Automatic Storage Management Administrator's 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 and later releases, Oracle supports in-place or out-of-place patching. Oracle supports only out-of-place upgrades.

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 or later releases, Oracle Clusterware and Oracle ASM binaries are installed as a single binary called the Oracle Grid Infrastructure.

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

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*

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.

  Starting with Oracle Grid Infrastructure 23ai, Cluster Health Monitor introduces a new diagnostic ability to listen for critical component events that indicates pending or actual failure and reports these with recommended corrective actions. In some cases, these actions may be processed autonomously. Improving the robustness and reliability of the Oracle Database hosting infrastructure is a critical business requirement for enterprises. This improved ability to detect and correct at first failure and self-heal autonomously delivers value by improving business continuity.
- **Configuration 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 and management of individual resources.

  Oracle Clusterware 12c introduced 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, such as `crsctl status server` 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. Starting in Oracle Database 21c, PDBs are a resource managed by Oracle Clusterware. You can use SRVCTL commands to manage PDBs.

Related Topics

- *Oracle Autonomous Health Framework User's Guide*
- Configuration Verification Utility Reference
  Learn about Configuration Verification Utility (CVU), and how to use it to prepare for installations, patching, or other system changes.
- 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
  This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.
- Oracle Interface Configuration Tool (OIFCFG) Command Reference
  The OIFCFG) command-line interface helps you to define and administer network interfaces.
- 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 and policies within your Oracle Clusterware environment without making any actual changes.

Reasoned command evaluation expands on this by showing you why Oracle Clusterware performs a particular action through verbose output of various CRSCTL commands.

> **Note:**
>
> The policy-managed database deployment option is desupported in Oracle Database 23ai.

The following example shows that the resource `cs1` is online on all three servers in the cluster:

```
$ crsctl stat res -t
--------------------------------------------------------------------------------
--
Name Target State Server State details
--------------------------------------------------------------------------------
--
Cluster Resources
--------------------------------------------------------------------------------
--
cs1
      1 ONLINE ONLINE server_n1_0 STABLE
      2 ONLINE ONLINE server_n1_1 STABLE
      3 ONLINE ONLINE server_n1_2 STABLE
--------------------------------------------------------------------------------
--
```

The following example shows what happens to the resource `cs1` when the server `server_n1_1` goes down:

```
$ crsctl eval delete server server_n1_1
Stage Group 1:
--------------------------------------------------------------------------------
--
Stage Required Action
--------------------------------------------------------------------------------
--

    1 Y Resource 'cs1' (2/1) will be in state 'OFFLINE'.
          Y Server 'server_n1_1' will be removed from pools 'Free'.

    2 Y Resource 'cs1' (2/1) will be in state 'ONLINE|INTERMEDIATE' on
                server 'server_n1_3'.

--------------------------------------------------------------------
```

The information contained in the preceding examples 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
  This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.

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 Grid Infrastructure 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
  This chapter describes how to clone an Oracle Grid Infrastructure home and use the cloned home to create a cluster.
- 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
  This appendix describes the Oracle Clusterware C application program interfaces (APIs).

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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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.

> **Note:**
>
> The policy-managed database deployment option is desupported in Oracle Database 23ai.

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, 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 in Oracle Clusterware no longer depends on a cluster administrator (although Oracle maintains backward compatibility). By default, the user who 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 using the Oracle Database Configuration Assistant (Oracle DBCA), but do not enable role separation.

Configuring Role Separation

Role separation is the determination of the roles that are needed, the resources that they will administer, and what their access privileges should be.

After you determine the roles, 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 and create the cluster.

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 user-defined cluster resources and types.

You create the resources under the `root` or `grid` accounts. For the designated operating system users to administer these 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, resources, or both. The CRSCTL 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 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
  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.

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.

> **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 remaining 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 exist after the tie-breaking process, perhaps because of certain hardware characteristics, or that certain resources remain, 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 remain active.

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.

Overview of Grid Naming Service

Oracle Clusterware uses Grid Naming Service (GNS) for address resolution in a cluster environment.

> **Note:**
>
> The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai.
> The highly-available GNS provides the ability to run multiple GNS instances in a multi-cluster environment with different roles. This feature is being deprecated. There is no replacement.

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 remaining cluster member node. If the cluster is an Oracle Flex Cluster configuration, then Oracle Clusterware fails over the GNS daemon and the GNS VIP.

> **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 for Linux*
- Oracle Extended Clusters
  You can extend an Oracle RAC cluster across two, or more, geographically separate sites, each equipped with its own storage.

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.

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.

Administering Grid Naming Service

Use SRVCTL to administer Grid Naming Service (GNS) in a cluster environment.

> **Note:**
>
> The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai.
> The highly-available GNS provides the ability to run multiple GNS instances in a multi-cluster environment with different roles. This feature is being deprecated. There is no replacement.

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
```

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 Configuration 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 -allnodes
   ```

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 for Linux*

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.

> **Note:**
>
> You must configure the IPMI driver either on all or none of the cluster nodes.

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 sectioin.

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, `ipmitool` or `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 `ipmitool` or `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.

> **Note:**
>
> You must configure the IPMI driver either on all or none of the cluster nodes.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*

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.

IPMI Post-installation Configuration with Oracle Clusterware

After you install the `ipmitool` or `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 `ipmitool` or `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 `ipmitool` or `ipmiutil` binary location.

After you complete the server configuration, and the configuration of the `ipmitool` or `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 `ipmitool` or `ipmiutil` binary location on each node.

   For example:

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

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

   > **Note:**
   >
   > The binary name must end in either `ipmitool` or `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 `ipmitool` or `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 OLR. Because the configuration information is kept in a secure store, it must be written by the Oracle Grid Infrastructure 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 `ipmitool` or `ipmiutil` binary file if the binary file location has been changed. For example:

```
$ crsctl set ipmi binaryloc /usr/bin/ipmitool
$ 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.

> **Note:**
>
> The binary name must end in either `ipmitool` or `ipmiutil`.

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 for Linux*
- 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 who 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 who installed Oracle Clusterware.

To completely remove IPMI, perform the following steps. To reconfigure IPMI as the user who 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 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.
- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*

Understanding Network Addresses on Manually Configured Networks

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

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 12*c* 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.

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 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 Grid Infrastructure 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.

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.

Database instance registration with a listener succeeds only when the request originates from a valid node. The network administrator can specify a list of valid nodes, excluded nodes, or disable valid node checking altogether. The list of valid nodes explicitly lists the nodes and subnets that can register with the database. The list of excluded nodes explicitly lists the nodes that cannot register with the database. The control of dynamic registration results in increased manageability and security of Oracle RAC deployments.

By default, the SCAN listener agent sets `REMOTE_ADDRESS_REGISTRATION_`*`listener_name`* to a private IP endpoint. The SCAN listener accepts registration requests only from the private network. Remote nodes that are not accessible to the private network of the SCAN listener must be included in the list of valid nodes by using the `registration_invited_nodes_`*`alias`* parameter in the `listener.ora` file, or by modifying the SCAN listener using the command-line interface, SRVCTL.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 12c, for a SCAN listener, if the `VALID_NODE_CHECKING_REGISTRATION_`*`listener_name`* and `REGISTRATION_INVITED_NODES_`*`listener_name`* parameters are set in the `listener.ora` file, then the listener agent overwrites these parameters.

If you use the SRVCTL utility to set the `invitednodes` and `invitedsubnets` values, then the listener agent automatically sets `VALID_NODE_CHECKING_REGISTRATION_`*`listener_name`* to SUBNET and sets `REGISTRATION_INVITED_NODES_`*`listener_name`* to the specified list in the `listener.ora` file.

For other listeners managed by CRS, the listener agent sets `VALID_NODE_CHECKING_REGISTRATION_`*`listener_name`* to be SUBNET in the `listener.ora` file

only if it is not already set in the `listener.ora` file. The SRVCTL utility does not support setting the `invitednodes` and `invitedsubnets` values for a non-SCAN listener. The listener agent does not update `REGISTRATION_INVITED_NODES_listener_name` in the `listener.ora` file for a non SCAN listener.

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:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.

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.

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.

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 11*g* 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 -allnodes -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.

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: `eth1`	You can use wildcards when specifying network interface names. For example: `eth*`
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 `oifcfg setif` command to designate an interface as a private interface.

Related Topics

- Oracle Interface Configuration Tool (OIFCFG) Command Reference
  The OIFCFG) command-line interface helps you to define and administer network interfaces.

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, 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 for Linux*

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. 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
  The OIFCFG) command-line interface helps you to define and administer network interfaces.

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.

10. After the previous step completes, you can remove the static IPv4 address, as follows, by providing the name and subnet address:

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

    For example:

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

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.

6. After the previous step completes, you can remove the static IPv4 address, as follows, by providing the name and subnet address:

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

For example:

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

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.

3

Oracle Extended Clusters

You can extend an Oracle RAC cluster across two, or more, geographically separate sites, each equipped with its own storage.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

About Oracle Extended Clusters

An Oracle Extended Cluster consists of nodes that are located in multiple locations called sites. 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.

Converting to Oracle Extended Cluster

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.

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.

1. As the `root` user, perform a complete backup of the OCR and voting files.

   ```
   # ocrconfig -manualbackup
   ```

2. Log in to the first node, and run the following command.

   ```
   # rootcrs.sh -converttoextended -first -sites list_of_sites -site node_site
   ```

   list_of_sites is the comma-separated list of sites in the extended cluster and *node_site* is the name of the site with which the local node is associated.

   > **Note:**
   >
   > The node on which you are running the `converttoextended` command becomes unavailable, which may disrupt the database access.

3. Run the following command on all other cluster nodes:

   ```
   # rootcrs.sh -converttoextended -site node_site
   ```

   node_site is the name of the site with which the local node is associated.

   > **Note:**
   >
   > The node on which you are running the `converttoextended` command becomes unavailable, which may disrupt the database access.

4. Delete the default site after the associated nodes and storage are migrated.

   ```
   # crsctl delete cluster site site_name
   ```

5. Associate every Oracle ASM disk with a site by using the `ALTER DISKGROUP` SQL statement as the SYSASM user on the Oracle ASM instance after mounting the disk groups in restricted mode.

   ```
   SQL> ALTER DISKGROUP diskgroup_name RENAME DISK disk_name SITE site_name;
   ```

   > **Note:**
   >
   > If a disk group contains Oracle Clusterware data, like voting files and the Oracle ASM SPFILE, then use the standard procedure for migrating the voting files and the Oracle ASM SPFILE to a different disk group, before the site assignment for the disks. For a disk group to store voting files, a normal redundancy disk group requires a minimum of three disk devices. You can migrate voting files and the Oracle ASM SPFILE back to the original disk group after the site assignment for the disks in the disk group.

   After the disk groups are modified successfully, you can remount the disk groups in normal mode.

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 'la' identified by GUID '7b7b3bef4c1f5ff9ff8765bceb45433a' in state 'ENABLED' contains
nodes 'node1,node2' and disks 'disk1, disk2, disk3'.
Site 'ny' identified by GUID '888b3bef4c1f5ff9ff8765bceb45433a' in state 'ENABLED' contains
nodes 'node3,node4' and disks 'disk4, disk5, disk6'.
Site 'nj' identified by GUID '999b3bef4c1f5ff9ff8765bceb45433a' in state 'ENABLED' contains
nodes 'node5,node6' and disks 'disk7, disk8, disk9'.
```

4

Oracle Clusterware to Manage Generic Applications

You can configure Oracle Grid Infrastructure with minimum network and storage requirements to manage generic applications.

About Oracle Grid Infrastructure to Manage Generic Applications

An Oracle Grid Infrastructure deployment to manage generic applications can host any applications for which you want to enable dual instance management, failover, and high availability.

Oracle Grid Infrastructure deployment to manage generic applications enables you to make your applications highly available without requiring Virtual IP addresses and shared storage. This deployment model can also extend high availability and single server rolling patching capabilities to single-instance Oracle databases.

You can configure Oracle Grid Infrastructure to manage generic applications without SCAN, GNS, VIPs, and without using separate network interfaces for Oracle ASM, private, and public communication. Such clusters can manage single-instance Oracle databases, any generic applications, or third-party databases using only one network interface.

> **Note:**
>
> You can easily convert this deployment model to an Oracle Grid Infrastructure cluster to manage Oracle RAC databases by adding shared storage and VIP.

Enabling Oracle RAC Databases on a Generic Application Cluster

To enable creating Oracle Real Application Clusters (Oracle RAC) databases on an Oracle Grid Infrastructure deployment that was configured to manage generic applications, you must add VIP, SCAN, and SCAN listener to your cluster.

If you deploy Oracle Grid Infrastructure to manage generic applications, then you can not create Oracle RAC databases on such clusters without creating VIP, SCAN, and SCAN listener.

> **Note:**
>
> This procedure converts the general purpose cluster to a full Oracle RAC cluster.

1. As the `root` user, add a VIP to each Oracle Clusterware node:

   ```
   # srvctl modify vip -node node_name -address {VIP_name|ip}/netmask[/if1[|
   if2|...]]
   [-netnum network_number] [-skip] [-verbose]
   ```

   Where:

   - *node_name* is the name of the cluster node.
   - *vip_name*/*netmask* is the name of a local VIP that resolves to both IPv4 and IPv6 addresses. You can specify one *VIP_name* or address, along with an IPv4 netmask or IPv6 prefix length.
   - *network_number* is the network number from which VIPs are obtained. The default network number is 1.

2. As the `grid` user, add the SCAN to the cluster.

   ```
   $ srvctl add scan -scanname myscan.example.com
   ```

3. Create a SCAN listener that is exclusive to this cluster.

   ```
   $ srvctl add scan_listener [-netnum  network_number] [-listener
   listener_name] [-skip]
   [-endpoints [TCP:]port_list[/IPC:<key>][/NMP:pipe_name][/TCPS:s_port][/
   SDP:port][/EXADIRECT:port]]
   [-invitednodes "node_list"] [-invitedsubnets "subnet_list"] [-
   clientcluster cluster_name] [-clientdata file_name]
   ```

4. Start SCAN and SCAN listener.

   ```
   $ srvctl start scan
   $ srvctl start scan_listener
   ```

5. Check the cluster node network configuration to ensure that all the resources are online and stable.

   ```
   $ crsctl stat res -t
   --------------------------------------------------------------------------
   -----
   Name           Target  State        Server                   State
   details
   --------------------------------------------------------------------------
   -----
   Local Resources
   --------------------------------------------------------------------------
   -----
   ora.LISTENER.lsnr
                  ONLINE  ONLINE       node1                    STABLE
                  ONLINE  ONLINE       node2                    STABLE
   ora.net1.network
                  ONLINE  ONLINE       node1                    STABLE
                  ONLINE  ONLINE       node2                    STABLE
   ora.ons
                  ONLINE  ONLINE       node1                    STABLE
   ```

```
               ONLINE  ONLINE       node2                         STABLE
--------------------------------------------------------------------------------
-----
Cluster Resources
--------------------------------------------------------------------------------
-----
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE                                     STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE                                     STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE                                     STABLE
ora.node1.vip
      1        ONLINE  ONLINE                                     STABLE
ora.node2.vip
      1        ONLINE  ONLINE                                     STABLE
ora.scan1.vip
      1        ONLINE  ONLINE                                     STABLE
ora.scan2.vip
      1        ONLINE  ONLINE                                     STABLE
ora.scan3.vip
      1        ONLINE  ONLINE                                     STABLE
--------------------------------------------------------------------------------
-----
```

6. Use Oracle DBCA to create an Oracle RAC database on your cluster.

Related Topics

- srvctl modify vip
- srvctl add scan
- srvctl add scan_listener
- Using Oracle DBCA to Create an Oracle RAC Database

5

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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.

- 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:**
>
> You can store OCR and voting files on Oracle ASM or a shared file system on Oracle 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 Oracle Standalone Clusters.

Oracle recommends that you configure multiple voting files during Oracle Grid Infrastructure installation to improve availability. If you use Oracle ASM for Oracle Clusterware storage, 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 Grid Infrastructure installation process without stopping the cluster.

Managing Oracle Cluster Registry and Oracle Local Registry

To manage OCR and the Oracle Local Registry (OLR), use `OCRCONFIG`, `OCRDUMP`, and `OCRCHECK`.

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_base/crsdata/host_name/olr/`, where `host_name` is the host name of the node.

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

You can migrate OCR to reside on Oracle ASM, and take advantage of the improvements in managing Oracle Clusterware storage.

To migrate OCR to Oracle ASM using OCRCONFIG:

1. Ensure the upgrade to Oracle Clusterware 23ai 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, then 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, first ensure that the Oracle Clusterware stack is running, and then 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*

Migrating Oracle Cluster Registry from Oracle ASM to Other Types of Storage

To migrate the Oracle Cluster Registry (OCR) from Oracle ASM to another storage type, use this procedure.

1. Ensure the upgrade to Oracle Clusterware 23ai is complete. To verify the current running version, run the following command:

   ```
   $ 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 the OCR. The OCR supports up to five locations.

3. Ensure that there is at least 500 MB of space on the mount partition.
4. Ensure that the file that 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 then 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 network-attached storage (NAS) using `OCRCONFIG`.

```
# ocrconfig -add /nas/ocr
```

For OCRs not stored on Oracle ASM, Oracle recommends that you mirror the OCRs 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 23ai but did not choose to mirror OCR during the upgrade
- Created only one OCR location during the Oracle Grid Infrastructure installation

Oracle recommends that you configure at least three OCR locations, if OCR is configured on NFS or on any cluster file system that is certified by Oracle. Mirroring can help prevent OCR from becoming a single point of failure.

> **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.

> **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 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
  Lists and describes OCRCONFIG commands.
- *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.

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.

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.

Starting with Oracle Clusterware 18c, you can use Oracle ASM or a shared file system for generating backups, 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
  Lists and describes OCRCONFIG commands.

Restoring Oracle Cluster Registry

Learn how to check for Oracle Cluster Registry (OCR) issues, and how to resolve those issues.

How to Check Oracle Cluster Registry Issues

If you encounter an Oracle Cluster Registry (OCR) issue, then review these checks and guidelines.

If a resource fails, then before attempting to restore OCR, restart the resource. As a definitive verification that OCR failed, run `ocrcheck`. 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 by 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

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.

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 -allnodes -verbose
```

Restoring the Oracle Cluster Registry on Windows Systems

Use this procedure to restore the Oracle Cluster Registry (OCR) on Microsoft 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 CRSD 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.

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 Configuration Verification Utility (CVU) command to verify OCR integrity of all of the nodes in your cluster database:

```
C:\>cluvfy comp ocr -allnodes -verbose
```

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.
> - After 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.

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
   ```

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.

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, then 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.

5. To verify the integrity of OCR, enter the following command:

```
# ocrcheck
```

6. Stop Oracle Clusterware on the node where it is running in exclusive mode, using the following command:

```
# 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 -allnodes -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 Automatic Storage Management Administrator's Guide*
- Backing Up Oracle Cluster Registry
  This section describes how to back up OCR content and use it for recovery.
- Configuration Verification Utility Reference
  Learn about Configuration Verification Utility (CVU), and how to use it to prepare for installations, patching, or other system changes.

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 Configuration Verification Utility (CVU) command to verify OCR integrity of all of the nodes in your cluster database:

```
C:\>cluvfy comp ocr -allnodes -verbose
```

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- Configuration Verification Utility Reference
  Learn about Configuration Verification Utility (CVU), and how to use it to prepare for installations, patching, or other system changes.

Oracle Local Registry

In Oracle Clusterware, 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 Grid Infrastructure 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_base*`/crsdata/`*host_name*`/olr/`*hostname_release*`.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 Local Registry is as follows :
        Version                  : 4
        Total space (kbytes)     : 491684
        Used space (kbytes)      : 83120
        Available space (kbytes) : 408564
        ID                       : 1260380057
        Device/File Name         : Grid_base/crsdata/dglnx6/olr/
dglnx6_21.olr
                                   Device/File integrity check succeeded
        Local registry integrity check succeeded
        Logical corruption 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_base*/crsdata/*host_name*/olr/.

- 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
  ```

Managing 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 and later releases. Restoring voting files that were copied using `dd` or `cp` commands can prevent the Oracle Clusterware 12c and later releases 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 23ai, 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 `19.0` or later release.

Related Topics

- Restoring Oracle Cluster Registry
  Learn how to check for Oracle Cluster Registry (OCR) issues, and how to resolve those issues.
- *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 remain functional after 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 remain functional after 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
  If all of the voting files are corrupted, then you can use this procedure to restore them.

Restoring Voting Files

If all of the voting files are corrupted, then you can use this procedure to restore them.

1. Restore the OCR as described in "Restoring Oracle Cluster Registry", if necessary.

   This step is necessary only if the OCR is also corrupted, or otherwise unavailable, such as if the OCR is on Oracle ASM, and the ASM 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 command `crsctl query css votedisk` to retrieve the list of voting files currently defined.

   For example:

   ```
   $ 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 the list 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

- *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

6

Adding and Deleting Cluster Nodes

Describes how to add nodes to an existing cluster, and how to delete nodes from clusters.

> **Note:**
>
> - Unless otherwise instructed, perform all add and delete node steps as the user who installed Oracle Clusterware.
> - Oracle recommends that you use the cloning procedure described in "Cloning Oracle Clusterware" to create clusters.

Related Topics

- Cloning Oracle Clusterware
  This chapter describes how to clone an Oracle Grid Infrastructure home and use the cloned home to create a cluster.

Prerequisite Steps for Adding Cluster Nodes

This section lists prerequisite steps that you must follow before adding a node to a cluster.

> **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 Grid Infrastructure. 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. Use the `cluvfy stage -pre nodeadd` command to verify that 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.

   ```
   $ cluvfy stage -pre nodeadd -n node_list -method root
   ```

After completing this procedure, 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.

Related Topics

- Prerequisite Steps for Adding Cluster Nodes
  This section lists prerequisite steps that you must follow before adding a node to a cluster.

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 Oracle Grid Infrastructure Installer to Add a Node

> **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.

This procedure assumes that:

- There is an existing cluster with two nodes named node1 and node2.
- You have successfully installed Oracle Clusterware on node1 and node2.
- You are adding a node named node3.

To add a node to the cluster using the Oracle Grid Infrastructure installer

1. In the *Grid_home* on an existing node of the cluster, run ./gridSetup.sh as the *grid* user to start the installer. The *Grid_home* is the Oracle Grid Infrastructure home.
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 on the node being added to populate the /etc/oraInst.loc file with the location of the central inventory. For example:

   ```
   # /opt/oracle/oraInventory/orainstRoot.sh
   ```

   > **Note:**
   >
   > If there is no database with a preconfigured database instance for the new node, run the *Grid_home*/root.sh script as prompted. You may be prompted to run *Grid_home*/root.sh in the following steps, but you do not need to run it again after running the script successfully.

6. Run the *Grid_home*/root.sh script on the node3 as root and run the subsequent script, as instructed. Review the following note before running the script.

> **Note:**
>
> - If you ran the `root.sh` script in the previous step, then you do not need to run it again.
> - If you have any 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 database instances using the `srvctl remove instance` command.

7. Perform the following procedures that apply to your system configuration.

> **Note:**
>
> Note that running *`Oracle_home`*`/addnode` in interactive mode displays several prerequisite check failures because the new node has not yet been configured for the Oracle Grid Infrastructure. These warnings can be ignored.

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 who installed Oracle RAC using the following syntax:

```
$ ./addnode.sh "CLUSTER_NEW_NODES='{node3,node4}'"
```

b. Run the *`Oracle_home`*`/root.sh` script on `node3` as `root`, where *`Oracle_home`* is the Oracle RAC home.

   Click **OK** to close the **Execute Configuration Scripts** dialog box.

c. Click **Close** to complete the installation process and exit the installer.

d. Navigate to the `ORACLE_HOME/bin/` directory on `node1` and run Oracle DBCA as the user who installed Oracle RAC.

```
$ ./dbca
```

e. Select **Oracle RAC database instance management** on the **Database Operation** window and click **Next**.

f. Select **Add an instance** on the **Instance Operation** window and click **Next**.

g. Select the database for which you want to add an instance on the new node on the **Select Database** window and click **Next**.

   You can also enter database credentials on this window to use if operating system authentication is not possible.

h. Specify the instance name on the **Instance Details** window and click **Next**.

i. Review the configuration details on the **Summary** window and click **Finish**.

After the instance addition is complete, click **Close** on the Finish window to exit Oracle DBCA.

If you have a shared Oracle home that is shared using Oracle Advanced 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 who 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 who installed Oracle RAC using the following syntax:

```
$ ./addnode.sh -noCopy "CLUSTER_NEW_NODES='{node3,node4}'"
```

> **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 who 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 who installed Oracle RAC using the following syntax:

```
$ ./addnode.sh -noCopy "CLUSTER_NEW_NODES='{node3,node4}'"
```

> **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.

8. Start the Oracle ACFS resource on the new node (`node3`) by running the following command as `root` from the *`Grid_home`*`/bin` directory:

```
# srvctl start filesystem -device volume_device_name -node node3
```

> **Note:**
>
> - This step is required only if there were Oracle ACFS file systems registered in the cluster before performing the adding cluster node procedure. If the step is required, then repeat for each registered Oracle ACFS file system.
> - 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 who 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]
```

Using Oracle Fleet Patching and Provisioning to Add a Node

If you have a Oracle Fleet Patching and Provisioning (Oracle FPP) Server, then you can use Oracle FPP 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).

Related Topics

- cluvfy stage 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.

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- Oracle Fleet Patching and Provisioning Administrator's Guide

Deleting a Cluster Node on Linux and UNIX Systems

Delete a node from a cluster on Linux and UNIX systems.

> **Note:**
>
> - 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 you create node-specific configuration for a node, such as disabling a service on a specific node, then 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.

To delete a node from a cluster, you can do one of the following procedures. The first procedure using `gridSetup.sh` is the recommended procedure.

> **Note:**
>
> Some steps in the below procedures include commands that you need to run from the node that you are deleting. If the node that you are deleting is not accessible, then you can skip these steps.

Using gridSetup to Delete a Node

1. As the `grid` user, run the `gridSetup.sh` script in the Oracle Grid Infrastructure home from a node that you are *not deleting*.

   ```
   $ Grid_home/gridSetup.sh
   ```

2. Select the **Remove nodes from the cluster** option, and click **Next**.
3. Select the nodes that you want to delete, and click **Next**. Make sure that the node is accessible for you to be able to delete the selected node.
4. Select the root execution option, such as the `sudo` privilege.
5. When prompted by the installer, open a new terminal window and run the following scripts:

a. `rootdeinstall.sh` on the node that you are deleting.

b. `rootdelete.sh` on the node from the current local node.

6. Click **OK** on the dialog box to finish deleting the node.

Using the crsctl delete node Command to Delete a Node

1. (Optional) 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. (Optional) Run the following command as either `root` or the user who 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, stop all the running Oracle Database instances:

   ```
   $ srvctl stop instance -n node_to_be_deleted
   ```

4. (Optional) 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 who 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.

     > **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.

5. 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 [-purge]
```

Use the -purge option to delete the node permanently and to reuse its node number. However, if you want to add the deleted node back with the same node name and the same node number, then do not use the -purge option.

6. 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]
```

7. 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 -vip vip_name
$ srvctl remove vip -vip vip_name
```

Using Oracle 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 rhpclient -node clientnode2 -root
```

The preceding example deletes a node named clientnode2 from the Fleet Patching and Provisioning Client named rhpclient, using root credentials (login for the node you are deleting).

Related Topics

- cluvfy stage nodedel
- *Oracle Real Application Clusters Administration and Deployment Guide*
- Oracle Fleet Patching and Provisioning Administrator's Guide

Adding and Deleting Cluster Nodes on Windows Systems

Explains how to add a new cluster node or delete an existing cluster node on Windows systems.

> **See Also:**
>
> *Oracle Grid Infrastructure Installation and Upgrade Guide for Microsoft Windows* for more information about deleting an entire cluster

Adding a Node to a Cluster on Windows Systems

This procedure describes how to add a node to your cluster.

Ensure that you complete the prerequisites listed in "Prerequisite Steps for Adding Cluster Nodes" before adding nodes.

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

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}"
   ```

   `addnode.bat` starts Oracle Grid Infrastructure installer in the GUI mode.

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 -allnodes [-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 12*c*, the OraMTS service was created as part of a software-only installation. Starting with Oracle Database 12*c*, 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
  This section lists prerequisite steps that you must follow before adding a node to a cluster.
- *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. 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.

2. On a node that you are not deleting, run the following command:

   ```
   C:\>Grid_home\bin\crsctl delete node -n node_to_be_deleted
   ```

3. 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]
   ```

4. 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 stop vip -vip vip_name
   C:\> ORACLE_HOME\bin\srvctl remove vip -vip vip_name
   ```

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*
- cluvfy stage nodedel

7

Cloning Oracle Clusterware

This chapter describes how to clone an Oracle Grid Infrastructure home and use the cloned home to create a cluster.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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 23ai 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

Introduction to Cloning Oracle Clusterware

Cloning is the process of copying an existing Oracle Grid Infrastructure 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 Grid Infrastructure 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. 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/23.0.0/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

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. 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 only 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 for Linux*

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
  This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.

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.

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.

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

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 Linux* 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 gpnp/host_name
   [root@node1 grid]# find gpnp -type f -exec rm -f {} \;
   [root@node1 grid]# rm -rf cfgtoollogs/*
   [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/crsgenconfig_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/host_name/*
   Grid_home/bin/clsecho/*
   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/crsgenconfig_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.

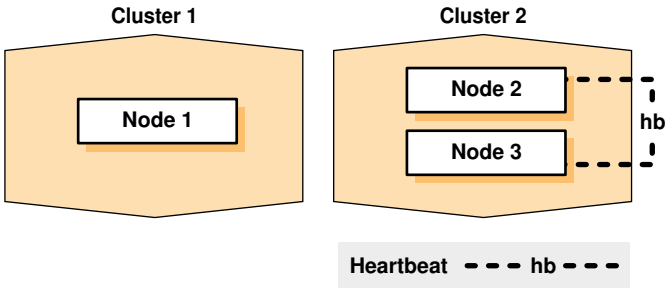
Creating a Cluster by Cloning Oracle Clusterware

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 7-1 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 7-1 Cloning to Create an Oracle Clusterware Environment



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 in the directory where you extract the archive file) on the node before you add the node to the cluster.

> **Note:**
>
> Starting with Oracle Clusterware 21c, the integration of vendor or third party clusterware with Oracle Clusterware is desupported.

8. To verify that your hardware and operating system environment is configured correctly, run the Configuration 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 Grid Infrastructure 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/23.0.0/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
  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.
- 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.

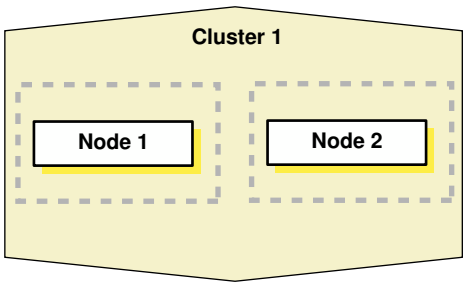
Once you launch the *`$ORACLE_home`*`/gridSetup.sh` utility, you can select to run it either in interactive mode, using the interactive interface, or in silent mode, where you provide a response file.

Using Cloning to Add Nodes to a Cluster

You can also use cloning to add nodes to a cluster.

Figure 7-2 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 7-2 Cloning to Add Nodes to a Cluster



Using Figure 7-2 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 Configuration Verification Utility (CVU) command on Node 1:

   ```
   $ cluvfy stage -post nodeadd -n destination_node_name [-verbose]
   ```

Related Topics

- cluvfy stage 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 7-1 lists the log files that are generated during cloning that are the key log files for diagnostic purposes.

> **Note:**
>
> *Central_inventory* in Table 7-1 refers to the Oracle Inventory directory.

Table 7-1 Cloning Log Files and their Descriptions

Log File Name and Location	Description
Central_inventory/logs/cloneActions*timestamp*.log	Contains a detailed log of the actions that occur during the OUI part of the cloning.
Central_inventory/logs/oraInstall*timestamp*.err	Contains information about errors that occur when OUI is running.
Central_inventory/logs/oraInstall*timestamp*.out	Contains other miscellaneous information.

Table 7-2 lists the location of the Oracle Inventory directory for various platforms.

Table 7-2 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	/var/opt/oracle/oraInst.loc
IBM AIX and Linux	/etc/oraInst.loc
Windows	C:\Program Files\Oracle\Inventory

8

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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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.

Oracle Clusterware Resources and Agents

This section discusses the framework that Oracle Clusterware uses to monitor and manage resources, to ensure high application availability.

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. 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
  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.
- 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.

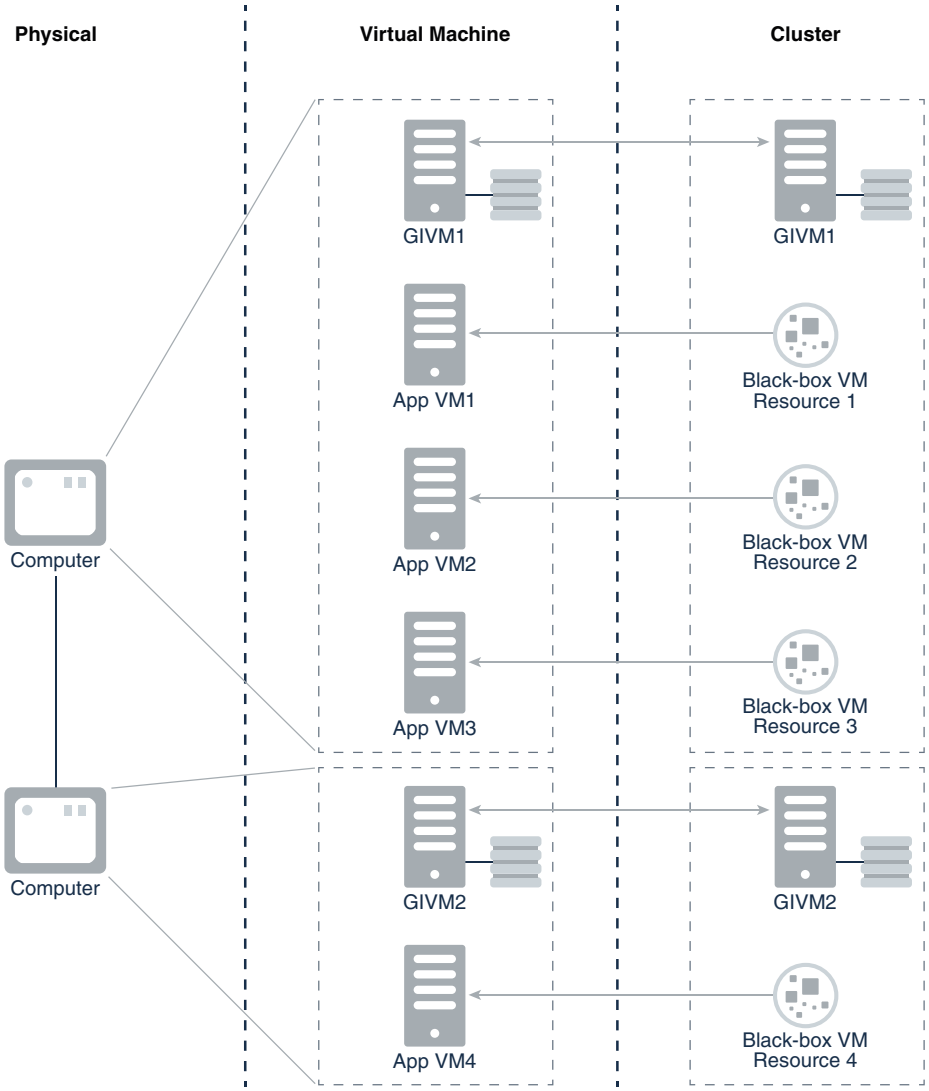
You can manage black-box Oracle virtual machines on physical hardware using Oracle Clusterware, which provides high availability and ease of management of virtual machines.

> **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 8-1 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 CRSCTL, 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 and also define privileges for modifying and running 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 8-1 Resource Group Dependency Types and Modifiers

Dependency Type	Description
hard start	Specifies the requirement that specific other resource groups must be online (anywhere in the cluster) before this resource group can be started. For example: `START_DEPENDENCIES=hard([global: | intermediate: | uniform: ] other_resource_group)` If the start of any dependent resource group fails, then Oracle Clusterware cancels the start of this resource group.
weak start	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. For example: `START_DEPENDENCIES=weak([global: | concurrent: | uniform: ] other_resource_group)`

Table 8-1 (Cont.) Resource Group Dependency Types and Modifiers

Dependency Type	Description
pullup	Use This dependency when this resource group must be automatically started when a dependent resource group starts. For example: `START_DEPENDENCIES=pullup([intermediate: | always: ]other_resource_group])` Oracle recommends that you use this dependency when a stop dependency exists between the resource groups.
hard stop	This dependency specifies the mandatory requirement of stopping this resource group when another specific resource group stops running. For example: `STOP_DEPENDENCIES=hard([intermediate: | global: | shutdown: ]other_resource_group)`
attraction	Specifies a co-location preference with specific other resource groups. For example: `START_DEPENDENCIES=attraction([intermediate:]other_resource_group)` Oracle Clusterware will attempt to start this resource group on the same server where a specific other resource group is already online.
dispersion	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. For example: `START_DEPENDENCIES=dispersion([intermediate: | active: | pool:]:other_resource_group)`
exclusion	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. For example: `START_DEPENDENCIES=exclusion[(preempt_pre: | preempt_post )] other_resource_group)`

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 CRSCTL 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
  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.

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 resourcegrouptype 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 stop responding, 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 stops responding, 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 8-2 at regular intervals, as specified by the `CHECK_INTERVAL` or `OFFLINE_CHECK_INTERVAL` resource attributes.

Table 8-2 Agent Framework Monitoring Characteristics

State	Condition	Frequency
ONLINE	Always	\ `CHECK_INTERVAL`
PARTIAL	Always	`CHECK_INTERVAL`
OFFLINE	Only if the value of the `OFFLINE_CHECK_INTERVAL` resource attribute is greater than 0.	`OFFLINE_CHECK_INTERVAL`
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 `CHECK_INTERVAL` 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 runs 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 8-2, then the agent framework periodically runs 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 11*g* 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 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
  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.

Registering a Resource in Oracle Clusterware

Register resources in Oracle Clusterware 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
  This section lists and describes attributes used when you register applications as resources in Oracle Clusterware.
- 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
  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.
- 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.
- 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
  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.

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 8-3 lists and describes the possible resource states.

Table 8-3 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 `crsctl stop resource` command to reset the state of the resource to OFFLINE.
INTERMEDIATE	A resource can be in the INTERMEDIATE state because of one of two events: 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. 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. 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. In either case, however, Oracle Clusterware transitions the resource out of the INTERMEDIATE state automatically as soon as it is appropriate.Use the `STATE_DETAILS` 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.

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 is deprecated in Oracle Clusterware 12c and later releases.

Resource dependencies are separated into start and stop categories. This separation improves and expands the start and stop dependencies between resources and resource types.

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.

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)`

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(intermedite: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 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 straightforward 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
  This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.
- 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_nummber -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*.

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 who 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 Extended Clusters
  You can extend an Oracle RAC cluster across two, or more, geographically separate sites, each equipped with its own storage.
- 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.

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.

> **Note:**
>
> Oracle recommends that you use shared storage, such as Oracle Advanced Cluster File System (Oracle ACFS), to store action scripts to decrease script maintenance.

Related Topics

- Examples of Action Scripts for Third-party Applications
  You can use third-party applications, such as Apache scripts, or `xclock`, by using script agents.

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 samba1 -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
  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.

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` attribute determines 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. With the `HOSTING_MEMBERS` attribute, 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 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.

- `favored`: If a value is assigned to either of the `HOSTING_MEMBERS` or `SERVER_CATEGORY` resource attributes, then that value expresses a preference. If `HOSTING_MEMBERS` is populated and `SERVER_CATEGORY` is set, then `HOSTING_MEMBERS` indicates placement preference and `SERVER_CATEGORY` indicates a restriction. For example, the `ora.cluster.vip` resource has a policy that sets the value of `PLACEMENT` to `favored`, 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 given category and then it prefers the server known as `server_name1`.
- `restricted`: Oracle Clusterware only considers 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.

  You can also use the `restricted:pack` variation in which the resource placement is further affected by the `WORKLOAD_CPU` attribute value of the resource and the current packing type of the cluster. Oracle Clusterware supports the following types of packing:

  - `fill`: Oracle Clusterware fills up a node with resources up to the configured CPUs before placing resources on other nodes. You can configure the amount of CPUs, which a resource uses on a node, using the `WORKLOAD_CPU` attribute.
  - `spread`: Oracle Clusterware spreads resources evenly among cluster nodes using the `WORKLOAD_CPU` as the measure.

Related Topics

- crsctl get restricted placement
- crsctl set restricted placement
- 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.

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 describes how to register, start, relocate, stop, and display resources.

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
  This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.

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
  This appendix contains reference information for the Oracle Clusterware Control (CRSCTL) utility.

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.

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 considers start dependencies contained in the profile of a resource when the start effort evaluation for that resource begins.
- 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.

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

Configuration Verification Utility Reference

Learn about Configuration Verification Utility (CVU), and how to use it to prepare for installations, patching, or other system changes.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Cluster Verification Utility (CVU) is renamed to Configuration Verification Utility (CVU).

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

When you modify your cluster, Oracle recommends that you use Configuration Verification Utility (CVU) to perform system checks. 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

About Configuration Verification Utility

Learn about Cluster Verification Utility (CVU) to ensure your cluster configuration is working as expected.

Overview of CVU

Configuration Verification Utility (CVU) is a centralized prerequisites repository that can verify the primary cluster components during an operational phase or stage.

Starting with Oracle Grid Infrastructure 23ai, CVU plays the role of a centralized prerequisites repository. The CLUVFY commands automatically include the prerequisite checks from tools such as Oracle ORAchk, Oracle EXAchk, Oracle Database AutoUpgrade, and Datapatch. CVU chooses a verification tool based on the current configuration stage of your system.

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	cluvfy comp sys
Oracle Cluster File System verification	cluvfy stage cfs
Storage verifications	• cluvfy comp freespace • cluvfy comp space • cluvfy comp ssa • cluvfy stage acfscfg
Network verification	cluvfy stage hwos
Connectivity verifications	• cluvfy comp nodecon • cluvfy comp nodereach
Time Synchronization Services verification	cluvfy comp clocksync
User and Permissions verification	cluvfy comp admprv
Node comparison and verification	cluvfy comp peer
Healthcheck verification	cluvfy comp healthcheck
Vital Healthcheck verification	cluvfy comp vital
Clusterware functionality verification	cluvfy comp clusterware
Patch verification	cluvfy stage patch
Deletion verification	cluvfy stage nodedel

Table A-1 (Cont.) Performing Various CVU Verifications

Verification to Perform	CVU Commands to Use
Oracle Clusterware and Oracle ASM Component verifications	• cluvfy comp acfs • cluvfy comp asm • cluvfy comp clumgr • cluvfy comp crs • cluvfy comp dhcp • cluvfy comp dns • cluvfy comp gns • cluvfy comp gpnp • cluvfy comp ha • cluvfy comp nodeapp • cluvfy comp ocr • cluvfy comp ohasd • cluvfy comp olr • cluvfy comp scan • cluvfy comp software • cluvfy comp vdisk

Downloading Configuration Verification Utility

Download the Configuration Verification Utility (CVU) for your Oracle Grid Infrastructure and Oracle Database installations from My Oracle Support.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Cluster Verification Utility (CVU) is renamed to Configuration Verification Utility (CVU).

1. Use a web browser to view the My Oracle Support website:

   https://support.oracle.com

2. Log in to My Oracle Support website.

   > **Note:**
   >
   > If you are not a My Oracle Support registered user, then click **Register for My Oracle Support** and register.

3. On the main My Oracle Support page, click **Patches & Updates**.
4. In the **Patch Search** region, type **30839369** in the **Patch Name or Number** search box.
5. Click the plus icon to add a new filter and select **Platform** from the drop-down list.
6. In the **Platform** drop-down list, select your server platform and click **Search**.

   The **Patch Search** page opens, displaying the results of your search.

7. Click **Patch Name** link of the CVU version you want to download.

The patch details page is displayed. It contains information about the CVU version, applicable releases and platforms, and a link to the README file. The README file provides steps to install CVU.

8. Click **Download** to download CVU and follow the instructions provided in the README file to install CVU.

CVU Operational Notes

Describes Cluster Verification Utility (CVU) installation, configuration, and operational requirements.

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 a value for the `CV_DESTLOC=`*`location`* parameter in the CVU configuration file `Grid_home/cv/admin/cvu_config`.

  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.

`cluvfy` (`cluvfy.bat` on Windows) are 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/23.0.0/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 commands may include the following report options:

- `-json`: Displays the report output in a JSON format. If the `-json` option is not specified, then the detailed report is displayed as text to the standard output.
- `-html`: Displays the CVU report output in HTML format. If the `-html` option is not specified, then the detailed report is displayed as text to the standard output.
- `-verbose`: Displays explanations about all CVU checks, both failed and passed, in the detailed summary section of the output.
- `-file` *file_location*: Depending on the format specified, save the output with the `.htm`, `.json`, or `.txt` file extension to the 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 Grid Infrastructure 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 stops responding.
- **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: May 5, 2023 6:40:41 AM
CVU version: 23.4.0.0.0 (050323x8664)
CVU home: /scratch/oracle/crshome
User: grid
Operating system: Linux5.4.17-2136.302.7.2.1.el8uek.x86_64

$ cluvfy comp baseline -collect all -allnodes
Collection report for this execution is saved in file "/scratch/oracle/
base/crsdata/@global/cvu/baseline/users/cvubaseline_04272022190027.zip".
CVU operation performed: baseline
Date: May 5, 2023 8:20:31 AM
CVU version: 23.4.0.0.0 (050323x8664)
Clusterware version: 23.0.0.0.0
CVU home: /scratch/oracle/crshome
Grid home: /u01/app/23.0.0/grid
User: grid
Operating system: Linux5.4.17-2136.302.7.2.1.el8uek.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_DESTLOC`: If set, it specifies a different location of the CVU work directory. 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.
- `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 (12.2, 18, 19, 20, 21, or 23) 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 23.
- 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 or Command Options	Deprecated	Desupported
`cluvfy comp cfs` (command)	Oracle Database 12c release 1 (12.1)	No
`-n all` (command option)	Oracle Database 21c	No

Note:

Use the `-allnodes` option, instad of `-n all` to run verification on all the cluster nodes.

Special CVU Topics

Describes how to generate Cluster Verification Utility (CVU) fixup scripts and perform other CVU operations.

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"

Periodic Cluster Health Check

Starting with Oracle Clusterware 19c, CVU is integrated as a cluster resource that performs Oracle Clusterware health check every 360 minutes.

CVU automatically runs as an Oracle Clusterware resource and periodically checks Oracle Clusterware health every 360 minutes or 6 hours by default. The period check time interval is contraolled by the `RUN_INTERVAL` attribute, which is set to `21600` seconds (21600 seconds = 360 minutes = 6 hours). You can modify this value using the `crsctl modify res` command.

The CVU healthcheck reports are stored in ZIP archives. You must unzip the CVU report archives stored in the `$ORACLE_BASE/crsdata/@global/cvu/baseline/cvures` directory. The archives contain only the XML files that you can open using an XML editor.

Example

The following example shows that the CVU periodic check feature is enabled and it set to run every 6 hours.

```
/u01/app/19c/grid/bin/srvctl config cvu
CVU is configured to run once every 360 minutes
CVU is enabled.
CVU is individually enabled on nodes:
CVU is individually disabled on nodes:
```

Enabling or Disabling Automatic Self Correction

You can enable or disable the Automatic Self Correction feature using the `configureAutoFixup` script.

The automated fixup framework of Configuration Verification Utility (CVU) identifies and corrects any configuration errors in your Oracle Clusterware environment without user

intervention. You can enable or disable Automatic Self Correction, using either GUI or command-line interface, during Oracle Grid Infrastructure installation or upgrade.

If you did not enable Automatic Self Correction during installation or upgrade, then use the following command to enable this feature:

- `$CRS_HOME/bin/configureAutoFixup enable`

Use the following command to check the configuration status of this feature:

- `$CRS_HOME/bin/configureAutoFixup status`

Use the following command to disable this feature:

- `$CRS_HOME/bin/configureAutoFixup disable`

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 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 who 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

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

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 `-allnodes`. 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:**
   >
   > The integration of vendor or third party clusterware with Oracle Clusterware is desupported in Oracle Clusterware 21c and later releases.

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]
```

Configuration Verification Utility Command Reference

This section lists and describes CVU commands.

cluvfy comp acfs

Checks the integrity of Oracle Advanced Cluster File System (Oracle ACFS) on the nodes in a cluster.

Syntax

```
cluvfy comp acfs [-n node_list | -allnodes] [-f file_system] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

> **Note:**
>
> Starting with Oracle Database 21c, the name of Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is changed to Oracle Advanced Cluster File System (Oracle ACFS). This change is only a change of the name. The basic function of Oracle's cluster file system continues to be the same. Oracle continues to develop and enhance Oracle ACFS.

Parameters

Table A-3 cluvfy comp acfs Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-f file_system`	The name of the file system to check.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.

Table A-3 (Cont.) cluvfy comp acfs Command Parameters

Parameter	Description
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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, or all nodes in a cluster.

Syntax

On Linux and UNIX platforms:

```
cluvfy comp admprv [-n node_list | -allnodes] {-o user_equiv [-sshonly] [-
fixup] | -o crs_inst
  [-asmgrp asmadmin_group] [-asmdbagrp asmdba_group] [-asmopergrp
asmoper_group] [-orainv orainventory_group]
  [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location
directory_path] | -method root]
  | -o db_inst [-osdba osdba_group] [-osoper osoper_group] [-fixup] [-
fixupnoexec]
  [-method sudo -user user_name [-location dir_path] | -method root] | -o
db_config -d oracle_home
  [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location
directory_path] | -method root]}
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

On Windows platforms:

```
cluvfy comp admprv [-n node_list | -allnodes] {-o user_equiv [-fixup] | -o
crs_inst [-orainv orainventory_group]
  [-fixup] [-fixupnoexec] | -o db_inst [-fixup] [-fixupnoexec] | -o db_config
-d oracle_home [-fixup] [-fixupnoexec]}
  [-verbose] [-json] [-html] [-savedir directory_path]]
```

Parameters

Table A-4 cluvfy comp admprv Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-o user_equiv [-sshonly]`	Checks user equivalence between the nodes. On Linux and UNIX platforms, you can optionally verify user equivalence using `ssh` by adding the `-sshonly` parameter.
`-o crs_inst [option]`	Checks administrative privileges for installing Oracle Clusterware. Optionally, you can specify the following: `-asmgrp`: Specify the name of the OSASM group. The default is `asmadmin`. `-asmdbagrp`: Specify the name of the ASMDBA group. The default is `asmdba`. `-orainv`: Specify the name of the Oracle Inventory group. The default is `oinstall`.
`-o db_inst option]`	Checks administrative privileges for installing an Oracle RAC database. Optionally, you can specify the following: `-osdba`: The name of the OSDBA group. The default is `dba`. `-osoper`: The name of the OSOPER group.
`-o db_config -d oracle_home`	Checks administrative privileges for creating or configuring an Oracle RAC database. Specify the location of the Oracle home for the Oracle RAC database.
`-fixup`	Specifies that if the verification fails, then CVU generates fixup instructions, 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 user_name [-location dir_path] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.

Table A-4 (Cont.) cluvfy comp admprv Command Parameters

Parameter	Description
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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\23.0.0\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\23.0.0\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 | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-5 cluvfy comp asm Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 Grid Infrastructure installation, before and after upgrading Oracle Grid Infrastructure, 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
| -allnodes]
   [-d Oracle_home | -db db_unique_name] [-bestpractice | -mandatory] [-
binlibfilesonly]
   [-reportname report_name] [-savedir save_dir]
   [-method sudo -user user_name [-location directory_path] | -method root]
   [-orachk_loc location] [-skip check_utilities]

cluvfy comp baseline -compare baseline1,baseline2,... [-cross_compare] [-
deviations] [-savedir save_dir]
```

Parameters

Table A-6 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 *Grid_home*`/cv/report/xml` directory. You can collect a baseline related to Oracle Clusterware (`cluster`), the database (`database`), Oracle ASM, or all three (`all`).
`-n` *node_list* `| -allnodes`	Specify a comma-delimited list of non domain-qualified node names, or all nodes in a cluster, on which the test should be conducted.
`-d` *Oracle_home*	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.
`-db` *db_unique_name*	The name of the database for which you want to collect information. When collecting a database baseline, if you specify the `-db` parameter, then CVU only collects the baseline for the specified database. If you do not specify `-db`, then CVU discovers all of the cluster databases configured in the system and the collects baselines for each of those.
`-bestpractice | -mandatory`	Specify `-bestpractice` to collect a baseline for only best practice recommendations. Specify `-mandatory` to collect a baseline for only mandatory requirements.

Table A-6 (Cont.) cluvfy comp baseline Command Parameters

Parameter	Description
`-binlibfilesonly`	Specify `-binlibfilesonly` to collect only files in the `bin/`, `lib/`, and `jlib/` subdirectories of the software home.
`-report` *`report_name`*	Use this optional parameter to specify a name for the report.
`-savedir` *`save_dir`*	Use this optional parameter to specify a location in which CVU saves the reports. If you do not specify the `-savedir` option, then CVU saves the reports in the *`Grid_home`*`/cv/report` directory.
`-method sudo -user user_name [-location dir_path] | -method 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.
`-compare` *`baseline1`*`,`*`baseline2`*`,...`	Specify `-compare` 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.
`-cross_compare`	Specify `-cross_compare` to compare baselines across clusters or across cluster nodes and databases.
`-deviations`	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 `-bestpractice` and `-mandatory` parameters).
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.

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 -allnodes -db orcl -bestpractice -report
bl1
   -savedir /tmp

$ cluvfy comp baseline -compare bl1,bl2
```

cluvfy comp clocksync

Checks clock synchronization across all the nodes in the node list.

> **Note:**
>
> Cluster Time Synchronization Service (CTSS) is desupported in Oracle Database 23ai.
> To synchronize time between cluster member nodes, use either an operating system configured network time protocol such as `ntp` or `chrony`, or Microsoft Windows Time service. To verify that you have network time synchronization configured, you can use the `cluvfy comp clocksync -n allnodes` command.

CVU verifies a time synchronization service is running (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 [-n node_list | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-7 cluvfy comp clocksync Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.

[1] Applicable only on UNIX platforms

Table A-7 (Cont.) cluvfy comp clocksync Command Parameters

Parameter	Description
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy comp clu

Checks the integrity of the cluster on all nodes.

Syntax

```
cluvfy comp clu [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-8 cluvfy comp clu Command Parameters

Parameter	Description
`–verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.

Table A-8 (Cont.) cluvfy comp clu Command Parameters

Parameter	Description
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

> **Note:**
>
> The command has been deprecated and may be desupported in a future release.

You can also choose to print detailed output.

> **See Also:**
>
> CVU Usage Information for CVU operational notes including information about report options

cluvfy comp clumgr

Checks the integrity of the cluster manager subcomponent, or Oracle Cluster Synchronization Services (CSS), on the nodes in the node list.

Syntax

```
cluvfy comp clumgr [-n node_list | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-9 cluvfy comp clumgr Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

You can specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.

You can also choose to print detailed output.

cluvfy comp clusterware

Performs diagnostics and reports potential issues that cause Oracle Cluster Ready Services (CRS) startup failures.

Syntax

```
cluvfy comp clusterware [-time_out seconds] [-fixup] [-fixupnoexec] [-method
sudo -user user_name
[-location directory_path] | -method root] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-10 cluvfy comp clusterware Command Parameters

Parameter	Description
`-time_out`	You can specify a time out of 30 seconds or more for the command execution using the `-time_out` option. If the `-time_out` option is not specified, then a default value is 60 seconds for the command execution.
`-fixup`	Specifies that if the verification fails, then CVU generates fixup instructions, 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` *`user_name`* `[-location` *`dir_path`*`] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy comp containerhost

Performs system requirement checks for container host and reports issues that may prevent successful creation of containers for hosting Oracle Clusterware and Database.

Syntax

```
cluvfy comp containerhost [-type {docker|k8|podman}] [-fixup] [-fixupnoexec]
[-method sudo -user user_name
[-location directory_path] | -method root] [-verbose] [-json] [-html] [-save
[-savedir directory_path]]
```

Parameters

Table A-11 cluvfy comp containerhost Command Parameters

Parameter	Description
`-type {docker|k8|podman}`	Specify type of the container manager configured on the host.
`-fixup`	Specifies that if the verification fails, then CVU generates fixup instructions, 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` *`user_name`* `[-location` *`dir_path`*`] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy comp crs

Checks the integrity of the Cluster Ready Services (CRS) daemon on the specified nodes.

Syntax

```
cluvfy comp crs [-n node_list | -allnodes] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-12 cluvfy comp crs Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this parameter, then CVU checks only the local node.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

You can also choose to print detailed output.

> **See Also:**
>
> CVU Usage Information for CVU operational notes including information about report options

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 |-allnodes] [-method sudo -user user_name [-location
directory_path] | -method root]
[-networks network_list] [-verbose] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-13 cluvfy comp dhcp Command Parameters

Parameter	Description
`-clustername cluster_name`	You must specify the name of the cluster of which you want to check the integrity of DHCP.
`-vipresname application_vip_resource_name`	Optionally, you can specify a comma-delimited list of the names of the application VIP resource.
`-port dhcp_port`	Optionally, you can specify the port to which DHCP packages are sent. The default port is 67.
`-n node_list | -allnodes`	Optionally, you can specify a comma-delimited list of non domain-qualified node names, or all nodes in a cluster, on which to conduct the verification. If you do not specify this parameter, then CVU checks only the local node.
`-method sudo -user user_name [-location directory_path] | -method 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 network_list`	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 `"if_name"[:subnet_id[:if_type[,if_type...]]]`. In the preceding format, you must enclose *`if_name`* in double quotation marks (`""`), and you can use regular expressions, such as `".*"`, as in `"eth*"`, to match interfaces like `eth1` and `eth02`. The *`subnet_id`* is the subnet number of the network interface. The *`if_type`* is a comma-separated list of interface types: `{CLUSTER_INTERCONNECT | PUBLIC | ASM}`.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.

Table A-13 (Cont.) cluvfy comp dhcp Command Parameters

Parameter	Description
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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] | -method root] [-
```

```
verbose]
  [-json] [-html] [-save [-savedir directory_path]]

cluvfy comp dns -client -domain gns_sub_domain -vip gns_vip [-port dns_port]
  [-last] [-method sudo -user user_name [-location directory_path] | -method
root] [-n node_list|-allnodes] [-verbose]
  [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-14 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 *gns_sub_domain*	Specify the name of the GNS subdomain.
-vipaddress *gns_vip_address*	Specify the GNS virtual IP address in the form {*ip_name* | *ip_address*}/*net_mask*/*interface_name*. You can specify either *ip_name*, which is a name that resolves to an IP address, or *IP_address*, which is an IP address. Either name or address is followed by *net_mask*, which is the subnet mask for the IP address, and *interface_name*, which is the interface on which to start the IP address.
-vip *gns_vip*	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 *dns_port*	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.
-method sudo -user *user_name* [-location *directory_path*] | -method 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.
-n *node_list* | -allnodes	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
-verbose	CVU prints detailed output.
-json	Use the -json flag to generate a detailed report in JSON format. If you do not specify the -json flag, then CVU generates the detailed report in a text file.
-html	Use the -html flag to generate a detailed report in HTML format. If you specify the -html flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the -html flag, then CVU generates the detailed report in a text file.

Table A-14 (Cont.) cluvfy comp dns Command Parameters

Parameter	Description
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

- You must run this command as `root`.
- Run `cluvfy comp dns -server` on one node of the cluster.
- On the last node, specify the `-last` option to terminate the `cluvfy comp dns -server` instance.
- Run `cluvfy comp dns -client` on the specified nodes, or all nodes of the cluster, to verify DNS server setup for the cluster.
- 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 | -allnodes]
```

If you choose to include the `-n` option, then enter a comma-delimited list of node names, or all nodes in a cluster, 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]}
  [-networks network_list] [-n node_list | -allnodes] [-verbose] [-json] [-
html] [-save [-savedir directory_path]]
```

```
cluvfy comp gns -postcrsinst [-verbose] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-15 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 *gns_vip*	Specify the GNS virtual IP address.
-domain *gns_domain*	Optionally, you can specify the GNS subdomain name.
-networks *network_list*	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 "*if_name*"[:*subnet_id*[:*if_type*[,*if_type*...]]]. In the preceding format, you must enclose *if_name* in double quotation marks (""), and you can use regular expressions, such as ".*", as in "eth*", to match interfaces like eth1 and eth02. The *subnet_id* is the subnet number of the network interface. The *if_type* is a comma-separated list of interface types: {CLUSTER_INTERCONNECT | PUBLIC | ASM}.
-n *node_list* | -allnodes	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
-json	Use the -json flag to generate a detailed report in JSON format. If you do not specify the -json flag, then CVU generates the detailed report in a text file.
-html	Use the -html flag to generate a detailed report in HTML format. If you specify the -html flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the -html flag, then CVU generates the detailed report in a text file.
-save [-savedir *directory_path*]	Use the -save or -save -savedir flags to save validation reports (cvuchecdkreport_timestamp.txt and cvucheckreport_timestamp.htm), where timestamp is the time and date of the validation report. If you use the -save flag by itself, then CVU saves the reports in the CVU_home/cv/report directory, where CVU_home is the location of the CVU binaries. If you use the -save -savedir flags, then specify a directory where you want CVU to save the reports.
-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 -verbose.
-verbose	CVU prints detailed output.

cluvfy comp gpnp

Checks the integrity of Grid Plug and Play on a list of nodes in a cluster.

Syntax

```
cluvfy comp gpnp [-n node_list | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-16 cluvfy comp gpnp Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy comp ha

Checks the integrity of Oracle Restart on the local node.

Syntax

```
cluvfy comp ha [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-17 cluvfy comp ha Command Parameters

Parameter	Description
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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] [-json] [-html] [-save [-savedir
directory_path]]
   [-orachk_loc location] [-skip check_utilities]
```

Parameters

Table A-18 cluvfy comp healthcheck Command Parameters

Parameter	Description
`-collect {cluster|database|asm}`	Use `-collect` to specify that you want to perform checks for Oracle Clusterware (`cluster`), Oracle Database (`database`), or Oracle ASM. If you do not use the `-collect` flag with the `healthcheck` option, then CVU performs checks for all three.

Table A-18 (Cont.) cluvfy comp healthcheck Command Parameters

Parameter	Description
`-db db_unique_name`	Use `-db` to specify checks on the specific database that you enter after the `-db` flag. If you use the `-db` flag but do not provide a database unique name, then CVU reports error. CVU uses JDBC to connect to the database as the user `dbsnmp` to verify various database parameters.
`[-bestpractice|-mandatory] [-deviations]`	Use the `-bestpractice` flag to specify best practice checks, and the `-mandatory` flag to specify mandatory checks. Add the `-deviations` 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 `-bestpractice` or `-mandatory` flag, but not both flags. If you specify neither `-bestpractice` nor `-mandatory`, then CVU displays both best practices and mandatory requirements.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.
`-orachk_loc location`	Specify location of the ORAchk binaries.

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 | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-19 cluvfy comp nodeapp Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 | -allnodes] [-networks network_list] [-
multicast]
[-repeat -iterations number_of_iterations [-interval interval_in_seconds]] [-
verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-20 cluvfy comp nodecon Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-networks network_list`	Specify a forward slash (/)-delimited list of networks on which to conduct the verification.
`-multicast`	CVU performs only multicast connectivity check.
`-repeat -iterations number_of_iterations`	CVU repeats the connectivity check for the number of iterations specified in the `-iterations` flag.
`-interval interval_in_seconds`	Specify the time interval between iterations.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 -allnodes -verbose
```

cluvfy comp nodereach

Determines whether a source node can communicate with other, specific nodes, or all nodes in a cluster.

Syntax

```
cluvfy comp nodereach {-n node_list | -allnodes} [-srcnode node] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-21 cluvfy comp nodereach Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster.
`-srcnode node`	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.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.

Table A-21 (Cont.) cluvfy comp nodereach Command Parameters

Parameter	Description
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 | -allnodes] [-method sudo -user user_name [-
location directory_path] | -method root]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-22 cluvfy comp ocr Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-method sudo -user user_name [-location directory_path] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 configurationl...
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 | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-23 cluvfy comp ohasd Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.

Table A-23 (Cont.) cluvfy comp ohasd Command Parameters

Parameter	Description
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 -allnodes -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 | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-24 cluvfy comp olr Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

- 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 | -allnodes} [-refnode node]
[-r {12.2|18|19|20|21|23}] [-osdba osdba_group] [-osoper osoper_group] [-
orainv orainventory_group]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-25 cluvfy comp peer Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names, or all nodes in a cluster, on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
`-refnode node`	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 {12.2|18|19|20|21|23}`	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 23ai or Oracle Database 23ai.
`-osdba osdba_group`	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.
`-osoper osoper_group`	Optionally, you can specify the name of the OSOPER group. **Note:** This parameter is not available on Windows systems.

Table A-25 (Cont.) cluvfy comp peer Command Parameters

Parameter	Description
`-orainv` *orainventory_group*	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.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *directory_path*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 23ai:

```
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] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-26 cluvfy comp scan Command Parameters

Parameter	Description
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 | -allnodes] [-d oracle_home]
[-allfiles] [-checksum] [-json] [-html] [-save [-savedir directory_path]] [-
verbose]
```

Parameters

Table A-27 cluvfy comp software Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Optionally, you can specify a comma-delimited list of non domain-qualified node names, or all nodes in a cluster, on which to conduct the verification. If you do not specify this option, then CVU checks only the local node.
`-d oracle_home`	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.
`-allfiles`	If you specify this parameter, then CVU checks the attributes of all files of the specified Oracle home. If you do not specify this parmater, then CVU checks the attributes of the `lib`, `jlib`, and `bin` files under the specified Oracle home.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

Add additional information about the command here.

Example

To verify that the installed files for Oracle Clusterware are configured correctly, use a command similar to the following:

```
$ cluvfy comp software -allnodes -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

```
cluvfy comp space [-n node_list | -allnodes] -l storage_location -z
disk_space {B | K | M | G} [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-28 cluvfy comp space Command Parameters

Parameter	Description
`-n node_list`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this parameter, then CVU checks only the local node.
`-l storage_location`	Specify the directory path to the storage location you want to check.
`-z disk_space {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.

Table A-28 (Cont.) cluvfy comp space Command Parameters

Parameter	Description
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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:

```
$ cluvfy comp space -allnodes -l /home/dbadmin/products -z 5G -verbose
```

cluvfy comp ssa

Use the `cluvfy 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, or all nodes in a cluster.

Syntax

```
cluvfy comp ssa [-n node_list | -allnodes] [-s storage_path_list]
  [-t {gi_software | rac_software | data | ocr_vdisk}]  [-asm [-asmdev
asm_device_list]]
  [-r {12.2|18|19|20|21|23}] [-verbose] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-29 cluvfy comp ssa Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify any of these options, then CVU checks only the local node.
`-s storage_path_list`	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 {gi_software | rac_software | data | ocr_vdisk}`	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 asm_device_list`	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 {12.2|18|19|20|21|23}`	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 `23`.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.

Table A-29 (Cont.) cluvfy comp ssa Command Parameters

Parameter	Description
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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.

> **See Also:**
>
> CVU Usage Information for CVU operational notes including information about report options

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 acessibility

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 | -allnodes]
  -p {crs | ha | database}  [-r {12.2|18|19|20|21|23}] [-osdba osdba_group] [-
osoper osoper_group]
  [-orainv orainventory_group] [-fixup] [-fixupnoexec] [-method sudo -user
user_name
  [-location directory_path] | -method root] [-verbose] [-json] [-html] [-
save [-savedir directory_path]]
```

Parameters

Table A-30 cluvfy comp sys Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-p {crs | ha | database}`	Specifies whether CVU checks the system requirements for Oracle Clusterware, Oracle Restart (HA), or Oracle RAC.
`-r {12.2|18|19|20|21|23}`	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 23ai.

Table A-30 (Cont.) cluvfy comp sys Command Parameters

Parameter	Description
`-osdba` *osdba_group*	The name of the OSDBA group. If you do not specify this option, then CVU uses `dba` as the OSDBA group.
`-osoper` *osoper_group*	The name of the OSOPER group.
`-orainv` *orainventory_group*	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` *user_name* `[-location` *directory_path*`] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *directory_path*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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
```

cluvfy 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

```
cluvfy comp vdisk [-n node_list | -allnodes] [-verbose]
[-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-31 cluvfy comp vdisk Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy comp vital

Performs vital healthchecks and reports issues that may affect the stability of Oracle Grid Infrastructure.

Syntax

```
cluvfy comp vital [-verbose|-progress] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-32 cluvfy comp vital Command Parameters

Parameter	Description
`[-verbose|-progress]`	Use the `-verbose` flag to print detailed output, or `-progress` flag to print current progress of the command.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage acfscfg

Use the `cluvfy stage -pre acfscfg` command to verify that your cluster nodes are set up correctly before configuring Oracle Advanced Cluster File System (Oracle ACFS). Use the `cluvfy stage -post acfscfg` to check an existing cluster after you configure Oracle ACFS.

Syntax

```
cluvfy stage -pre acfscfg {-n node_list | -allnodes} [-asmdev
asm_device_list] [-afdconfig] [-verbose]
    [-json] [-html] [-save [-savedir directory_path]]
```

```
cluvfy stage -post acfscfg {-n node_list | -allnodes} [-verbose] [-json] [-
html] [-save [-savedir directory_path]]
```

> **Note:**
>
> Starting with Oracle Database 21c, the name of Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is changed to Oracle Advanced Cluster File System (Oracle ACFS). This change is only a change of the name. The basic function of Oracle's cluster file system continues to be the same. Oracle continues to develop and enhance Oracle ACFS.

Parameters

Table A-33 cluvfy stage acfscfg Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names, or all nodes in a cluster, on which to conduct the verification, for both before and after configuring Oracle ACFS.
`-asmdev asm_device_list`	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, `/dev/raw/*` on Linux systems.
`-afdconfig`	Verifies Oracle ASM Filter Driver configuration prerequisites.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage appcluster

Performs the appropriate pre-stage and post-stage checks for Oracle Clusterware application cluster installations on the specified nodes.

Syntax

```
cluvfy stage -pre appcluster {-n node_list | -allnodes} [-verbose]

cluvfy stage -post appcluster {-n node_list | -allnodes} [-method sudo -user
user_name
   [-location dir_path] | -method root] [-verbose]
```

Parameters

Table A-34 cluvfy stage appcluster Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which you want to run the verification. Specify `-allnodes` to run the verification on all nodes in the cluster.
`-method sudo -user user_name[-location dir_path] | -method root`	Chose the privilege delegation method, either `sudo` or `root`, to be used for root user access. If you choose the `sudo` method, then you must provide a user name to access all the nodes with `root` privileges, and, optionally, the full file system path for the `sudo` executable.

cluvfy stage 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 Linux* for your platform for a list of supported shared storage types.

Syntax

```
cluvfy stage -pre cfs {-n node_list | -allnodes} -s storageID_list [-verbose]
    [-json] [-html] [-save [-savedir directory_path]]

cluvfy stage -post cfs {-n node_list | -allnodes} -f file_system [-verbose]
    [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-35 cluvfy stage cfs Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification, for both before and after configuring OCFS2. Specify `-allnodes` to run the verification on all nodes in the cluster.
`-s storageID_list`	Specify a comma-delimited list of storage locations to check before configuring OCFS2.
`-f file_system`	Specify a file system to check after configuring OCFS2.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

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 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 -responseFile file_name [-fixup] [-fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root] [-
r {12.2|18|19|20|21|23}]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]] [-orachk_loc
location] [-skip check_utilities]

cluvfy stage -pre crsinst -file config_file [-fixup] [-fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root] [-
verbose]
  [-json] [-html] [-save [-savedir directory_path]]

cluvfy stage -pre crsinst {-n node_list | -allnodes} [-r {12.2|18|19|20|21|
23}] [-c ocr_location_list] [-q voting_disk_list]
  [-osdba osdba_group] [-osoper osoper_group] [-orainv orainventory_group] [-
asm [-presence {local | flex}
  | -asmcredentials client_data_file] [-asmgrp asmadmin_group] [-asmdev
asm_device_list]] [-afdconfig]
  [-crshome Grid_home] [-fixup] [-fixupnoexec] [-method sudo -user user_name
[-location directory_path] | -method root]
  [-networks network_list] [-dhcp -clustername cluster_name [-dhcpport
dhcp_port]] [-dns -vipaddress gns_vip_address
  [-domain gns_sub_domain]] [-software_only] [-verbose] [-json] [-html] [-
save [-savedir directory_path]] [-orachk_loc location]
  [-skip check_utilities]

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] | -method root]
  [-verbose] [-orachk_loc location] [-skip check_utilities] [-json] [-html] [-
save [-savedir directory_path]]

cluvfy stage -post crsinst {-n node_list | -allnodes}
  [-method sudo -user user_name [-location directory_path] | -method root] [-
verbose]
  [-orachk_loc location] [-skip check_utilities] [-json] [-html] [-save [-
savedir directory_path]]
```

Parameters

Table A-36 cluvfy stage [-pre | -post] crsinst Command Parameters

Parameter	Description
`-responseFile`	Specify the response file, created by the Oracle Universal Installer, containing Oracle installation variables.
`-file config_file`	Specify the root script configuration file containing Oracle installation variables.
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. Specify `-allnodes` to run the verification on all nodes in the cluster.

Table A-36 (Cont.) cluvfy stage [-pre | -post] crsinst Command Parameters

Parameter	Description
`-r {12.2|18|19|20|21|23}`	Specifies the Oracle Grid Infrastructure release that CVU checks as required for installation of Oracle Grid Infrastructure. If you do not specify this option, then CVU assumes Oracle Grid Infrastructure 23ai.
`-c ocr_location_list`	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 voting_disk_list`	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 osdba_group`	Specify the name of the OSDBA group. If you do not specify this option, then CVU uses `dba` as the OSDBA group.
`-osoper osoper_group`	Specify the name of the OSOPER group.
`-orainv orainventory_group`	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 client_data_file`	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 asmadmin_group`	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 asm_device_list`	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.
`-afdconfig`	Verifies Oracle ASM Filter Driver configuration prerequisites.
`-crshome Grid_home`	Specify the location of the Oracle Grid Infrastructure home directory. If you specify this parameter, then the supplied file system location is checked for sufficient free space for an Oracle Grid Infrastructure installation.
`-networks network_list`	Checks the network parameters of a slash ("/")-delimited list of networks in the form of "*if_name*" `[:`*subnet_id* `[:public | :cluster_interconnect]]`. • You can use the asterisk (`*`) wildcard character when you specify the network interface name (*if_name*), such as `eth*`, to match interfaces. • Specify a subnet number for the network interface for the *subnet_id* variable and choose the type of network interface.
`-dhcp -clustername cluster_name [-dhcpport dhcp_port]`	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.
`-dns -vipaddress gns_vip_address`	Specify the GNS virtual IP address in the form `{`*ip_name* `|` *ip_address*`}/`*net_mask*`/`*interface_name*. You can specify either *ip_name*, which is a name that resolves to an IP address, or *IP_address*, which is an IP address. Either name or address is followed by *net_mask*, which is the subnet mask for the IP address, and *interface_name*, which is the interface on which to start the IP address.
`-domain gns_sub_domain`	Specify the name of the GNS subdomain.

Table A-36 (Cont.) cluvfy stage [-pre | -post] crsinst Command Parameters

Parameter	Description
`-upgrade`	Specify this parameter to verify upgrade prerequisites.
`-rolling`	Specify this parameter to perform a rolling upgrade.
`-src_crshome` *src_crshome*	Specify the location of the source Grid home.
`-dest_crshome` *dest_crshome*	Specify the location of the destination Grid home.
`-dest_version` *dest_version*	Specify the version to which you are upgrading, such as `23.0.0.0.0`.
`-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` *user_name* `[-location` *dir_path*`] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *directory_path*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

- To perform checks for a new installation, specify `-responseFile`, `-file`, or `-n` parameters. 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 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 | -allnodes} -d Oracle_home [-fixup] [-
fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root] [-
orachk_loc location]
  [-skip check_utilities] [-verbose] [-json] [-html] [-save [-savedir
directory_path]]
```

On Windows platforms:

```
cluvfy stage -pre dbcfg {-n node_list | -allnodes} -d oracle_home [-fixup] [-
fixupnoexec]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-37 cluvfy stage -pre dbcfg Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of nondomain qualified node names on which to conduct the verification. Specify `-allnodes` to run the verification on all nodes in the cluster.
`-d Oracle_home`	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 user_name [-location directory_path] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.

Table A-37 (Cont.) cluvfy stage -pre dbcfg Command Parameters

Parameter	Description
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage 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 | -allnodes} [-r {12.2|18|19|20|21|
23}] [-osdba osdba_group]
  [-osoper osoper_group] [-osbackup osbackup_group] [-osdg osdg_group]
  [-oskm oskm_group] [-d oracle_home] [-fixup] [-fixupnoexec] [-method sudo -
user user_name
  [-location directory_path] | -method root] [-verbose] [-orachk_loc
location] [-skip check_utilities]
  [-json] [-html] [-save [-savedir directory_path]]

cluvfy stage pre dbinst -upgrade -src_dbhome src_dbhome [-dbname dbname-list]
-dest_dbhome dest_dbhome
  -dest_version dest_version [-pdbs pdbname-list] [-fixup] [-fixupnoexec] [-
method sudo -user user_name
  [-location directory_path] | -method root] [-verbose] [-orachk_loc
location] [-skip check_utilities]
  [-json] [-html] [-save [-savedir directory_path]]
```

On Windows platforms:

```
cluvfy stage -pre dbinst {-n node_list | -allnodes} [-r {12.2|18|19|20|21|23}]
  [-d Oracle_home] [-fixup] [-fixupnoexec] [-serviceuser user_name [-
servicepasswd]]]
```

```
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]

cluvfy stage -pre dbinst -upgrade -src_dbhome src_dbhome [-dbname dbname-list
  -dest_dbhome dest_dbhome -dest_version dest_version [-fixup] [-fixupnoexec]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-38 cluvfy stage -pre dbinst Command Parameters

Parameter	Description
`-n node_list | -allnodes`	Specify a comma-delimited list of non domain-qualified node names on which to conduct the verification. Specify `-allnodes` to run the verification on all nodes in the cluster.
`-r {12.2|18|19|20|21|23}`	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 23ai.
`-osdba osdba_group`	The name of the OSDBA group. If you do not specify this option, then CVU uses `dba` as the OSDBA group.
`-osoper osoper_group`	Specify the name of the OSOPER group.
`-osbackup osbackup_group`	Specify the name of the OSBACKUP group.
`-osdg osdg_group`	Specify the name of the OSDG group.
`-oskm oskm_group`	Specify the name of the OSKM group.
`-d oracle_home`	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 src_dbhome`	Specify the location of the source database home from which you are upgrading.
`-dbname dbname-list`	Specify a comma-delimited list of unique names of the databases you want to upgrade.
`-dest_dbhome dest_dbhome`	Specify the location of the destination database home to which you are upgrading.
`-dest_version dest_version`	Specify the version to which you are upgrading, such as `23.0.0.0.0`.
`-pdbs`	Specify a comma-delimited list of unique names of the pluggable databases (PDBs) you want to upgrade.
`-method sudo -user user_name [-location directory_path] | -method 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.

Table A-38 (Cont.) cluvfy stage -pre dbinst Command Parameters

Parameter	Description
`-serviceuser` *user_name* `[-servicepasswd]`	If you specify this option, then CVU checks the following: • 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. **Note:** You must specify the user name of the Oracle home user. • 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 `-servicepwd` 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 `-servicepwd` option, then CVU prompts you for the password of the specified user, even if the password exists in the wallet. **Note:** The `-serviceuser` and `-servicepwd` parameters only apply to Windows.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *directory_path*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage hacfg

Checks the local node before and after configuring Oracle Restart.

Syntax

```
cluvfy stage -pre hacfg [-osdba osdba_group] [-osoper osoper_group] [-orainv
orainventory_group]
```

```
  [-fixup] [-fixupnoexec] [-method sudo -user user_name [-location
directory_path] | -method root]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]

cluvfy stage -post hacfg [-verbose] [-json] [-html] [-save [-savedir
directory_path]]
```

Parameters

Table A-39 cluvfy stage hacfg Command Parameters

Parameter	Description
`-osdba osdba_group`	Specify the name of the OSDBA group. If you do not specify this option, then CVU uses `dba` as the OSDBA group.
`-osoper osoper_group`	Specify the name of the OSOPER group.
`-orainv orainventory_group`	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 user_name [-location dir_path] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage hwos

Checks network and storage on the specified nodes in the cluster. This command also checks for supported storage types and checks each one for sharing.

Syntax

```
cluvfy stage -post hwos {-n node_list | -allnodes} [-s storageID_list] [-
verbose]
    [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-40 cluvfy stage hwos Command Parameters

Parameter	Description
`-n node_list`	The comma-delimited list of non domain-qualified node names on which to conduct the verification. Specify `-allnodes` to run the verification on all nodes in the cluster.
`-s storageID_list`	Checks the comma-delimited list of storage locations for sharing of supported storage types. If you do not specify the `-s` parameter, then CVU discovers supported storage types and checks sharing for each of them.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage 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] [-fixup] [-
fixupnoexec]
  [-method sudo -user user_name [-location directory_path] | -method root]
  [-verbose] [-json] [-html] [-save [-savedir directory_path]]

cluvfy stage -post nodeadd -n node_list [-verbose] [-json] [-html] [-save [-
savedir directory_path]]
```

Parameters

Table A-41 cluvfy stage nodeadd Command Parameters

Parameter	Description
`-n node_list`	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.
`-vip vip_list`	A comma-delimited list of virtual IP addresses to be used by the new nodes.
`-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 user_name [-location directory_path] | -method 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.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.

Table A-41 (Cont.) cluvfy stage nodeadd Command Parameters

Parameter	Description
`-save [-savedir` *`directory_path`*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage 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] [-json] [-html] [-save [-
savedir directory_path]]
```

Parameters

Table A-42 cluvfy stage nodedel Command Parameters

Parameter	Description
`-n` *`node_list`*	Specify a comma-delimited list of non domain-qualified node names that you have deleted from the cluster.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.

Table A-42 (Cont.) cluvfy stage nodedel Command Parameters

Parameter	Description
`-save [-savedir directory_path]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

Usage Notes

- If the `cluvfy stage -post nodedel` check fails, then repeat the node deletion procedure.

> **See Also:**
>
> Adding and Deleting Cluster Nodes

cluvfy stage osupg

Performs the Oracle Grid Infrastructure prerequisites checks on all the nodes in the nodelist to ensure readiness for Linux operating system upgrade.

> **Note:**
>
> This command is supported only on Linux platforms.

Syntax

```
cluvfy stage -pre osupg {-n node_list | -allnodes} [-checkGIdriverSupport
kernel_information]
[-verbose] [-json] [-html] [-save [-savedir directory_path]]
```

Parameters

Table A-43 cluvfy stage osupg Command Parameters

Parameter	Description
`-n node_list | -allnodes`	The comma-delimited list of non domain-qualified node names on which to conduct the verification, or all nodes in a cluster. If you do not specify this option, then CVU checks only the local node.

Table A-43 (Cont.) cluvfy stage osupg Command Parameters

Parameter	Description
`-checkGIdriverSupport` *kernel_information*	Optionally, specify the operating system kernel version or an absolute path to the kernel RPM file to check for the Oracle Grid Infrastructure driver support.
`-verbose`	CVU prints detailed output.
`-json`	Use the `-json` flag to generate a detailed report in JSON format. If you do not specify the `-json` flag, then CVU generates the detailed report in a text file.
`-html`	Use the `-html` flag to generate a detailed report in HTML format. If you specify the `-html` flag, and a browser that CVU recognizes is available on the system, then CVU starts the browser and displays the report on the browser when the checks are complete. If you do not specify the `-html` flag, then CVU generates the detailed report in a text file.
`-save [-savedir` *directory_path*`]`	Use the `-save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_timestamp.txt` and `cvucheckreport_timestamp.htm`), where `timestamp` is the time and date of the validation report. If you use the `-save` flag by itself, then CVU saves the reports in the `CVU_home/cv/report` directory, where `CVU_home` is the location of the CVU binaries. If you use the `-save -savedir` flags, then specify a directory where you want CVU to save the reports.

cluvfy stage patch

Performs the appropriate pre-stage checks for the patch application on all nodes. Performs post-stage check for the patch application.

Syntax

```
cluvfy stage -pre patch [-db -dbhome oracle_home [-dbname dbname-list]] [-
non_rolling]
    [-verbose] [-orachk_loc location] [-skip check_utilities]

cluvfy stage -post patch [-db -dbhome oracle_home] [-verbose]
```

Parameters

Table A-44 cluvfy stage patch Command Parameters

Parameter	Description
`-db -dbhome` *oracle_home*	Specify the location of the database home which you are patching.

Table A-44 (Cont.) cluvfy stage patch Command Parameters

Parameter	Description
`-dbname` *`dbname-list`*	Specify a comma-delimited list of unique names of the databases you want to patch.
`-non_rolling`	Use the `-non_rolling` flag to apply patch in non-rolling mode.
`-verbose`	CVU prints detailed output.

Usage Notes

By default, the pre-stage checks are performed for patching in rolling mode.

Troubleshooting and Diagnostic Output for CVU

Describes how to enable Cluster Verification Utility (CVU) tracing and lists known issues for CVU.

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/`*`host_name`*`/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/`*`host_name`*`/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 Configuration Verification Utility

Describes known issues and limitations for Configuration Verification Utility (CVU).

Database Versions Supported by Configuration Verification Utility

This section describes CVU compatibility with Oracle Database versions.

The current CVU release supports only Oracle Database 12c Release 2 (12.2.0.1) 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 12c Release 2 (12.2.0.1).

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 Grid Infrastructure installation. 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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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.

Configurable Resource Attributes

Describes the 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

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 who 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 and later releases: `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 values are:

```
CARDINALITY=%CRS_SERVER_POOL_SIZE%
CARDINALITY=%CRS_HUB_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 `favored` or `restricted`. When registering applications as Oracle Clusterware resources, use the `SERVER_POOLS` attribute, instead.

> **See Also:**
>
> - "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

The amount of time (minutes) after the relocated instance starts and services are migrated, before the former instance of the service stops. The default value is 30 minutes.

> **See Also:**
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information about online database relocation
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information about `srvctl relocate database`

PLACEMENT

Specifies how Oracle Clusterware selects a cluster server on which to start a resource. Valid values are `balanced`, `favored`, or `restricted`.

If you set the `PLACEMENT` attribute to `favored` 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

Usage Example

```
PLACEMENT=favored
```

RANK

Use this resource attribute to set rank of a resource. The higher rank resources start first, fail over first, and they shut down last. You must log in as the `grid` user or the `root` user to modify RANK of any resource.

Usage Example

```
RANK=5
```

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. If you set PLACEMENT to `restricted`, then Oracle Clusterware expects one of the following attributes to also be set:

- HOSTING_MEMBERS
- SERVER_CATEGORY

If PLACEMENT is set to `favored` and if only one of HOSTING_MEMBERS, or SERVER_CATEGORY is set, then that value expresses a preference. If HOSTING_MEMBERS is populated and

SERVER_CATEGORY is set, then the HOSTING_MEMBERS indicates placement preference and SERVER_CATEGORY indicates a restriction.

Usage Example

```
SERVER_CATEGORY=my_category
```

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.

- `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
```

WORKLOAD_CPU

Assigns the CPU cores to a resource. The value must be in hundredths of a CPU core (1/100 CPU core). Specify a positive integer value that is equal to or greater than 10 and increase this value by 5, as required. You can set the value of this resource attribute to greater than 100 to assign more than one CPU cores to a resource.

Usage Example

The following example assigns half CPU core to the specified resource.

```
WORKLOAD_CPU=50
```

The following example assign two CPU cores to the specified resource.

```
WORKLOAD_CPU=200
```

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

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 running 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 running 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

Examples of Action Scripts for Third-party Applications

You can use third-party applications, such as Apache scripts, or `xclock`, by using script agents.

The following example 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
```

The following example shows shows the **xclock** script, which is an 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.example.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
```

The following example is of an action script example, using 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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

Using OLSNODES

This section contains topics which relate to using the OLSNODES command.

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 check that your cluster is operational, and all nodes are registered as members of the cluster. This command also provides a method for obtaining the node numbers.

Operational Notes

This command is used by the Configuration Verification Utility (CLUVFY) to obtain a list of node names when the `-allnodes` 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 who installed Oracle Clusterware, or the user who installed Oracle Database.

Summary of the OLSNODES Command

The `olsnodes` command does not use keywords, but accepts one or more options.

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: `active` or `inactive`.
`-t`	Displays node type: pinned or unpinned.
`node_name`	Displays information for a particular node.
`-l [-p]`	Lists the local node and includes the private interconnect for the local node. The `-p` option is only valid when you specify along with the `-l` 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
```

D

Oracle Interface Configuration Tool (OIFCFG) Command Reference

The 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

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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

Describes how to use Oracle Interface Configuration Tool (OIFCFG).

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

`oifcfg iflist [-p [-n]]`	Shows the available interfaces that you can configure with `setif`. The `iflist` command queries the operating system to find which network interfaces are present on this node. You can specify two options with this command:
	• `-p`: Displays a heuristic assumption of the interface type (PRIVATE, PUBLIC, or UNKNOWN)
	• `-n`: Displays the netmask
	If the interface has an IPv6 network, then OIFCFG prints the IPv6 subnet on a separate line, similar to the following:
	`eth1 fec0::80 PUBLIC 64`
	`eth1 10.229.184.0 PUBLIC 255.255.248.0`

`oifcfg setif`	Sets an interface type (public, cluster interconnect, or Oracle ASM) for an interface.
`oifcfg getif`	Displays the interfaces for which an interface type has been defined with the `setif` command, along with the type for that interface. **Note:** The `-node` command parameter that you can use with this command returns no result if you not set the interface details using the `oifcfg setif -node` command.
`oifcfg delif`	Deletes the stored network configuration for global or node-specific interfaces. You can use the `-force` option without specifying the `-node` or `-global` 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
  This appendix describes the syntax and command options for the `olsnodes` command.

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 11*g* 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-11*g* 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 Fusion[1] 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, only 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.
- 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.

[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.

OIFCFG Examples

These 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 running 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:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

> **Note:**
>
> Do not use CRSCTL commands on Oracle entities (such as resources and resource types) 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. Starting with Oracle Database 21c, PDBs are a resource managed by Oracle Clusterware. You can use SRVCTL commands to manage PDBs.

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

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.
- You must use the CRSCTL commands from the same release to manage Oracle Clusterware.

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
`crsctl check crsd`	`crsctl check crs`
`crsctl check cssd`	`crsctl check css`
`crsctl check evmd`	`crsctl check evm`
`crsctl debug res log` *`resource_name:level`*	`crsctl set log`

`crsctl set css votedisk`	`crsctl add css votedisk`
	`crsctl delete css votedisk`
	`crsctl query css votedisk`
	`crsctl replace votedisk`

`crsctl start resources`	`crsctl start resource -all`
`crsctl stop resources`	`crsctl stop resource -all`
`crsctl set css logfilesize`	`crsctl get tracefileopts css` `crsctl set tracefileopts css`
`crsctl get css logfilesize`	`crsctl set tracefileopts css`
`crsctl set trace` *`module_name`* `"`*`component:debugging_level`* `[,`*`component:debugging_level`* `][,...]"`	
`crsctl get trace`	

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
resource_name	Specify a short, descriptive name for the resource.
-type *resource_type*	Specify the type of resource that you are adding preceded by the -type flag.
-group *group_name*	Specify a resource group to which you want to add the resource.
-file *file_path*	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[,...]"`	You can specify attributes for a resource you are adding in two different ways: • 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 "HOSTING_MEMBERS='hostname1 hostname2',START_TIMEOUT=25"` • 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: *attribute_name*`{@SERVERNAME(`*server_name*`)` `[@DEGREEID(`*did*`)]	@CARDINALITYID(`*cid*`)` `[@DEGREEID(`*did*`)]}=`*attribute_value* 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`
`-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.
`-f`	Use the force parameter: • 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 have execute permissions on one or more of the dependencies. If you do not specify the force parameter in this case, an error displays.

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
  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.

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
resource_name	Specify one or more resource names that you want to delete. Multiple resources 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: • "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 *and* the name of the resource contains the number 2 **Note:** All operators must be surrounded by spaces. **See Also:** Filters for more information about operators
-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.
-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
resource_name	Specify a short, descriptive name for the resource.
-type *resource_type*	Specify the type of resource that you are adding following the -type flag.
-file *file_path*	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*"	You can specify attributes for a resource you are adding in two different ways: • 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 "HOSTING_MEMBERS='hostname1 hostname2',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: *attribute_name*{@SERVERNAME(*server_name*) [@DEGREEID(*did*)] | @CARDINALITYID(*cid*) [@DEGREEID(*did*)]}=*attribute_value* 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.
-f	Use the force parameter: • 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.

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
resource_name	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 `-w` parameter.
`-w "filter"`	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: • `"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 *and* the name of the resource contains the number `2` **Note:** All operators must be surrounded by spaces. **See Also:** Filters for more information about operators
`-n server`	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 (""). For example, you could use `-env "RESTART_ATTEMPTS=0"` to predict what would happen if the resource has run out of local restart attempts and cannot be restarted locally.

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
resource_name	Specify the name of a resource you want to relocate.
-all	Alternatively, you can use this parameter to specify all resources for relocation.
-s *source_server*	Specify a particular server *from* which you want to relocate the resources.
-w "*filter*"	Alternative to using -s, 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: • "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 *and* the name of the resource contains the number 2 **See Also:** Filters for more information
-n *destination_server*	Specify the name of the server *to* 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	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.

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
resource_name	Specify the name of the resource you want to evaluate.

Table E-7 (Cont.) crsctl eval modify resource Command Parameters

Parameter	Description

`-attr "attribute_name=attribute_value[,...]"`	You can specify attributes for a resource you want to evaluate in two different ways:
	• 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:
	`-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 "HOSTING_MEMBERS='hostname1 hostname2',START_TIMEOUT=25"`
	• 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:
	attribute_name`{@SERVERNAME(`*server_name*`) [@DEGREEID(`*did*`)] | @CARDINALITYID(`*cid*`) [@DEGREEID(`*did*`)]}=`*attribute_value*
	If you specify the `@SERVERNAME(`*server_name*`)` 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(`*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 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
resource_name [...]	Specify one or more space-delimited resource names to start.
-w "*filter*"	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 *server_name*	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
resource_name [...]	Specify one or more space-delimited resource names to stop.

Table E-9 (Cont.) crsctl eval stop resource Command Parameters

Parameter	Description
`-w "filter"`	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: • `"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
resource_name	Specify the name of the resource for which you want to obtain permissions.
`-u` *user_name*	If you specify `-u`, then Oracle Clusterware obtains permissions for a particular user.
`-g` *group_name*	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
resource_name	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"`	You can specify attributes for a resource you want to modify in two different ways:
	• 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:
	`-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 "HOSTING_MEMBERS='hostname1 hostname2',START_TIMEOUT=25"`
	• 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:
	attribute_name`{@SERVERNAME(`*server_name*`) [@DEGREEID(`*did*`)] | @CARDINALITYID(`*cid*`) [@DEGREEID(`*did*`)]}=`*attribute_value*
	If you specify the `@SERVERNAME(`*server_name*`)` 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(`*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`

`-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.

Table E-11 (Cont.) crsctl modify resource Command Parameters

Parameter	Description
`-f`	Use the `-f` parameter when: • 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` *`group_name`*	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 CRSCTL 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
  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.

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
resource_name	Specify the name of a resource you want to relocate.
-k *cid*	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 "*filter*"	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: • "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 *and* the name of the resource contains the number 2 **See Also:** "Filters" for more information
-s *source_server*	Specify a particular server *from* which you want to relocate the resources.
-n *destination_server*	Specify the name of the server *to* 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 "*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 ("").
-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 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
resource_name [...]	Specify one or more space-delimited names of resources you want to restart.
-w "*filter*"	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 *cid*	Specify the resource cardinality ID. If you specify this parameter, then Oracle Clusterware restarts the resource instances that have the cardinality you specify.

Table E-13 (Cont.) crsctl restart resource Command Parameters

Parameter	Description
`-d did`	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 "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 ("").
`-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
resource_name	Specify the name of the resource for which you want to set permissions.

Table E-14 (Cont.) crsctl setperm resource Command Parameters

Parameter	Description

`-u | -x | -o | -g`	You can set only one of the following permissions for a resource:
	• `-u` *`acl_string`*: You can update the access control list (ACL) for a resource
	• `-x` *`acl_string`*: You can delete the ACL for a resource
	• `-o` *`user_name`*: You can change the owner of a resource by entering a user name
	• `-g` *`group_name`*: You can change the primary group of a resource by entering a group name
	Specify a user, group, or other ACL string, as follows:
	
	`user:`*`user_name`*`[:`*`readPermwritePermexecPerm`*`] |`
	`group:`*`group_name`*`[:`*`readPermwritePermexecPerm`*`] |`
	`other[::`*`readPermwritePermexecPerm`*`]`
	
	• `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 [-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
resource_name [...]	Specify one or more space-delimited names of resources you want to start.
-w "*filter*"	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 *server_name*	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.
-k *cid*	Specify the resource cardinality ID. If you specify this parameter, then Oracle Clusterware starts the resource instances that have the cardinality you specify.
-d *did*	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 ("").
-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 -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 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 -l 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
resource_name [...] | -w "*filter*"	Specify one or more space-delimited names of resources of which you want to check the status. Optionally, you can specify a resource filter that CRSCTL 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: • "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 *and* 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. **See Also:** "Filters" for more information
[-p | -v] | [-f | -l | -g]	You can optionally specify the following parameters: • Specify either the -p parameter to display the static configuration of the resource or the -v parameter to display the run-time configuration of the resource. • Specify the -f parameter to display the full configuration of the resource; or specify the -l parameter to display all cardinal and degree values of the resource; or specify the -g parameter to check whether the specified resources are registered
[[-k *cid* | -n *server_name*] [-e [-p | -v]] [-d *did* | [-s -k *cid* [-d *did*]]]	You can specify one of the following two options: • Specify the -k *cid* parameter to specify a cardinality ID of the resources you want to query. Or you can specify the -n parameter to specify a particular server on which to check resources. Optionally, you can specify the -d parameter with the -n 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. Use the -e parameter to evaluate the special values of a resource instance. You must also specify -p or -v with the -e parameter. • Specify the -s parameter with the -k parameter to obtain a list of target servers for relocation. You can further limit the output by specifying a degree ID with the -d parameter.
-t	Specify the -t parameter to display the output in tabular form.
-p [-attr *attribute_name*[,...] [-noname]	Specify -attr (followed by a single attribute name or a comma-delimited list of attribute names) or -noname following -p to either print specific attributes or bypass the default printing of the resource name, respectively.

Table E-16 (Cont.) crsctl status resource Command Parameters

Parameter	Description
`-dependency [-stop | -pullup]`	Specify the `-dependency` parameter to display resource dependencies. If you do not specify either the `-stop` or `-pullup` option, then CRSCTL displays the start dependencies of the resource. Use either of the following options with the `-dependency` parameter: • Specify the `-stop` parameter to display resource stop dependencies. • Specify the `-pullup` 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.staii14.vip

NAME=ora.staii14.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on staii14
```

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] [-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
resource_name [...]	Alternative to using -w or -all, you can specify one or more space-delimited resource names that you want to stop.
-w "*filter*"	Alternative to specifying a resource name or using -all, 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: • "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
-all	Alternative to specifying a resource name or using -w, you can use this parameter to stop all resources in the cluster.
-n *server_name*	Optionally, you can specify the name of the server on which the resource instances you want to stop reside. If you do not specify a server, then CRSCTL stops all instances of the resource.
-k *cid*	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 *did*	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.

Table E-17 (Cont.) crsctl stop resource Command Parameters

Parameter	Description
`-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
group_name	Specify the name of the resource group.
`-type` *group_type*	Specify a group type for the resource group.

Table E-18 (Cont.) crsctl add resourcegroup Command Parameters

Parameter	Description
`-file` *file_path* | `-attr` "*attribute_name=attribute_value*[,...]"	Provide either a file or a list of attributes that define the resource group. 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 marks ("").
`-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
group_name	Specify a resource group or a number of resource groups in a space-delimited list that you want to check.
`-w "`*filter*`"`	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` *server_name*	Optionally, you can specify the name of the server on which the resource group you want to check resides.
`-k` *cid*	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
group_name	Specify one or more group names that you want to delete. Multiple groups 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: • "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 *and* the name of the resource contains the number 2 **Note:** All operators must be surrounded by spaces. **See Also:** Filters for more information about operators
-f	Use this parameter to delete 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.

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
group_name	Specify the name of the resource group.
-type *group_type*	Specify a group type for the resource group.

Table E-21 (Cont.) crsctl eval add resourcegroup Command Parameters

Parameter	Description
`-file file_path | -attr "attribute_name=attribute_value[,...]"`	Provide either a file or a list of attributes that define the resource group. 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 marks ("").
`-f`	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
group_name	Specify the name of a resource group you want to evaluate.
`-w filter`	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, `-w "TYPE = ora.database.type"` or `-w "NAME = cs1"`. **See Also:** Filters for more information about filters
`-n server`	Optionally, you can specify the name of the server to which resource groups reside that you want to evaluate.
`-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 ("").

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
group_name	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` *server*	Specify the name of the source server on which the resource group resides that you want to relocate. This parameter is required.
`-w` *filter*	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` *server*	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
group_name	Specify a resource group or a number of resource groups in a space-delimited list that you want to evaluate. This parameter is required.
`-w` *filter*	Alternatively, 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 specify a particular resource group. **See Also:** Filters for more information about filters
`-n` *server*	Specify the name of the server on which the resource group you want to evaluate resides.
`-f`	Use the `-f` 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
group_name	Specify a resource group or a number of resource groups in a space-delimited list that you want to stop.
`-w "filter"`	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, `-w "TYPE = ora.database.type"` or `-w "NAME = cs1"`. **See Also:** Filters for more information about using filters.
`-f`	Use the `-f` 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
group_name	Specify a resource group or a number of resource groups in a space-delimited list that you want to export.
`-file` *file_name*	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
group_name	Specify the name of the resource group you want to modify.

-attr "*attribute_name=attribute_value*"	You can specify attributes for a resource group you want to modify in two different ways:
	• 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:
	`-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 "HOSTING_MEMBERS='hostname1 hostname2',START_TIMEOUT=25"`
	• 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:
	attribute_name{@SERVERNAME(*server_name*) [@DEGREEID(*did*)] | @CARDINALITYID(*cid*) [@DEGREEID(*did*)]}=*attribute_value*
	If you specify the @SERVERNAME(*server_name*) 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(*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`

Table E-27 (Cont.) crsctl modify resourcegroup Command Parameters

Parameter	Description
`-i`	If you specify `-i`, then CRSCTL 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.
`-f`	Use the `-f` parameter when: • 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
group_name	Specify a resource group or a number of resource groups in a space-delimited list that you want to relocate.
`-k` *cid*	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 "`*filter*`"`	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

Table E-28 (Cont.) crsctl relocate resourcegroup Command Parameters

Parameter	Description
`-s source_server_name`	Specify the name of the source server on which the resource groups reside that you want to relocate. This parameter is required.
`-n target_server_name`	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
group_name	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
`-k cid`	Specify the resource group cardinality ID. If you specify this parameter, then Oracle Clusterware restarts the resource instances that have the cardinality you specify.

Table E-29 (Cont.) crsctl restart resourcegroup Command Parameters

Parameters	Description
`-env "env1=val1,env2=val2,..."`	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 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 `-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

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.

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] [-k cid] [-env "attr=val[,..."]]
  [-f] [-i] [-l]
```

Parameters

Table E-30 crsctl start resourcegroup Command Parameters

Parameter	Description
group_name	Specify a resource group or a number of resource groups in a space-delimited list that you want to start.
`-w "filter"`	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.

Table E-30 (Cont.) crsctl start resourcegroup Command Parameters

Parameter	Description
`-n server_name`	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.
`-k cid`	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:

```
$ crsctl start resourcegroup rg1
```

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
`group_name`	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.

Table E-31 (Cont.) crsctl status resourcegroup Command Parameters

Parameter	Description
`-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
`-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 cid`	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 server_name`	
`-e`	Optionally, you can specify this parameter to exclude *automatic* resource groups. The default is to display the status of all resource groups.
`-p | -v`	
`-s -k cid`	
`-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.

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] [-k cid] [-env "attr=val[,...]"]
  [-f] [-i] [-l]
```

Parameters

Table E-32 crsctl stop resourcegroup Command Parameters

Parameter	Description
group_name	Alternative to using `-w` or `-all`, you can specify a resource group or a number of resource groups in a space-delimited list that you want to stop.
`-w "`*filter*`"`	Alternative to specifying a resource group name or using `-all`, 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"`. **See Also:** Filters for more information about using filters.
`-all`	Alternative to specifying a resource name or using `-w`, you can specify this parameter to stop all resource groups on the cluster.
`-n` *server_name*	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.
`-k` *cid*	Specify the resource group cardinality ID. If you specify this parameter, then Oracle Clusterware stops 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 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.
`-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 resourcegrouptype

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 resourcegrouptype group_type_name -basetype base_group_type
  {-file file_path | -attr "attribute_name=attribute_value"} [-i]
```

Parameters

Table E-33 crsctl add resourcegrouptype Command Parameters

Parameter	Description
group_type_name	Specify the name of the resource group type.
-basetype *base_group_type*	Specify the base group type of the resource group type.
-file *file_path* | -attr "*attribute_name=attribute_value*"	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 ("").
-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 delete resourcegrouptype

Deletes one or more resource group types from Oracle Clusterware.

Syntax

```
crsctl delete resourcegrouptype {"group_type_name [...]" | -w "filter" [-i]
```

Parameters

Table E-34 crsctl delete resourcegrouptype Command Parameters

Parameter	Description
group_type_name	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 ("").

Table E-34 (Cont.) crsctl delete resourcegrouptype Command Parameters

Parameter	Description
`-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: • `"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 *and* the name of the resource contains the number `2` **Note:** All operators must be surrounded by spaces. **See Also:** Filters for more information about operators
`-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 modify resourcegrouptype

Modifies a resource group type to add attributes or update the values of existing attributes.

Syntax

```
crsctl modify resourcegrouptype 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

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
module_name	Specify one of the following modules to set 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 evm: Event Manager gipc: Grid Interprocess Communication
-filesize *file_size*	Specify the size of a single trace file in kilobytes (K|k), megabytes (M|m), or gigabytes (G|g).
-numsegments *number_of_segments*	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
type_name	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.

Table E-36 (Cont.) crsctl add type Command Parameters

Parameter	Description
-basetype *base_type_name*	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.
-attr "ATTRIBUTE=*attribute_name*	You can specify the resource type attributes using the -attr 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: 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 integer or string. 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 (|): 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. You *cannot* use multiple -attr arguments to define multiple arguments for the resource type. Instead, you can specify multiple types within the double quotation marks after the -attr flag. For example: `"ATTRIBUTE=FOO,TYPE=integer,DEFAULT_VALUE=0, FLAGS=REQUIRED|HOTMOD,ATTRIBUTE=BAR,TYPE=string"` The preceding example defines two type attributes, FOO and BAR. When you specify the ATTRIBUTE keyword, it ends the previous type attribute (if any) and begins a new type attribute.

Table E-36 (Cont.) crsctl add type Command Parameters

Parameter	Description

`-file file_path`	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:
	
	`ATTRIBUTE=FOO`
	`TYPE=integer`
	`DEFAULT_VALUE=0`
	`FLAGS=REQUIRED`
	`ATTRIBUTE=BAR`
	`TYPE=string`
	
	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.
	The preceding example defines two type attributes, `FOO` and `BAR`.
	Note: All operators must be surrounded by spaces.
	See Also: "Filters" for more information about operators

`-i`	If you specify `-i`, then the command fails if Oracle Clusterware cannot process the request immediately.

> **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
resource_type	Specify the resource type for which you want to obtain permissions.
`-u` *user_name*	If you specify `-u`, then Oracle Clusterware obtains permissions for a particular user.
`-g` *group_name*	If you specify `-g`, 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:rwx,pgrp:oinstall:rwx,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
type_name	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: • TYPE • DEFAULT_VALUE • FLAGS **Note:** Although you must specify the ATTRIBUTE keyword, you *cannot* modify it. **See Also:** crsctl add type for descriptions of these keywords
-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
resource_type_name	Specify the name of the resource type for which you want to set permissions.
-u | -x | -o | -g	You can specify only one of the following parameters for a resource type: • -u *acl_string*: You can update the access control list (ACL) for a resource type • -x *acl_string*: You can delete the ACL for a resource type • -o *user_name*: You can change the owner of a resource type by entering a user name • -g *group_name*: You can change the primary group of a resource type by entering a group name Specify a user, group, or other ACL string, as follows: user:*user_name*[:*readPermwritePermexecPerm*] | group:*group_name*[:*readPermwritePermexecPerm*] | other[::*readPermwritePermexecPerm*] • 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
resource_type_name [...] | -w "*filter*"	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: • `-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
```

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 and the member nodes
- Oracle ASM (if installed)
- Cluster 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-41 crsctl request action Command Parameters

Parameter	Description
action_name	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` *resource_name* `[...]`	Specify a particular resource. Multiple resource names must be separated by a space.
`-w "`*filter*`"`	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: • `TYPE == cluster_resource`: This filter limits Oracle Clusterware to perform actions on only resources of `cluster_resource` type • `CHECK_INTERVAL > 10`: This filter limits Oracle Clusterware to perform actions on only 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 perform actions on only 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
`-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-41 (Cont.) crsctl request action Command Parameters

Parameter	Description
`-i`	If you specify `-i`, then the command fails if Oracle Clusterware cannot process the request immediately.

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-42 crsctl add category Command Parameters

Parameter	Description
`category_name`	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 *`attr_name`* and *`attr_value`* 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:** • "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-43 crsctl modify category Command Parameters

Parameter	Description
category_name	Specify the name of the server category you want to modify.

Table E-43 (Cont.) crsctl modify category Command Parameters

Parameter	Description
`-attr "attr_name=attr_value[,...]"`	Specify the name-value pair of a category attribute you want to modify preceded by the `-attr` flag. **Note:** The *attr_name* and *attr_value* parameters must be enclosed in double quotation marks (""), and separate multiple name-value pairs with commas. For example: `"ACL='owner:st-cdc\cdctest:rwx,pgrp::---', EXPRESSION='(SITE_NAME = west)'"` **See Also:** • "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_DEPENDENCIESs=s'hard(res1,res2,res3)'"
```

Example

To modify a server category:

```
$ crsctl modify category blue_server -attr  "EXPRESSION='(SITE_NAME = west)'"
```

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-44 crsctl status category Command Parameters

Parameter	Description
category_name [...]	Specify the name of the server category or a space-delimited list of server categories for which you want to obtain the status.
`-w "filter"`	Alternatively, you can specify a category filter preceded by the `-w` flag. **See Also:** "Filters" for more information

Table E-44 (Cont.) crsctl status category Command Parameters

Parameter	Description
`-server server_name`	Alternatively, you can specify a particular server to list all of the categories that the server matches.

Examples

To print all resources containing `ASM`:

```
$ crsctl stat res -w "NAME co asm" -t
--------------------------------------------------------------------------------
--
Name    Target    State    Server             State details
--------------------------------------------------------------------------------
--
Cluster Resources
--------------------------------------------------------------------------------
--
ora.asm(ora.asmgroup)
1       ONLINE    ONLINE    stbm000042-vm1    Started,STABLE
2       ONLINE    ONLINE    stbm000042-vm2    Started,STABLE
3       ONLINE    ONLINE    stbm000042-vm3    Started,STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
1       ONLINE  ONLINE     stbm000042-vm1    STABLE
2       ONLINE  ONLINE     stbm000042-vm2    STABLE
3       ONLINE  ONLINE     stbm000042-vm3    STABLE
--------------------------------------------------------------------------------
--
```

To print all resources containing `db`:

```
$ crsctl stat res -w "NAME co db" -t
--------------------------------------------------------------------------------
--
Name    Target    State    Server              State details
--------------------------------------------------------------------------------
--
Cluster Resources
--------------------------------------------------------------------------------
--
ora.cdb19.db
1       ONLINE    ONLINE    stbm000042-vm1     Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
2       ONLINE    ONLINE    stbm000042-vm2     Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
3       ONLINE    ONLINE    stbm000042-vm3     Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
4       ONLINE    ONLINE    stbm000042-vm4     Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
--------------------------------------------------------------------------------
--
```

To print all resources containing `db` and their state is `ONLINE`:

```
crsctl stat res -w "(NAME en db) AND (STATE = ONLINE)" -t
--------------------------------------------------------------------------------
--
Name   Target   State   Server             State details
--------------------------------------------------------------------------------
--
Cluster Resources
--------------------------------------------------------------------------------
--
ora.cdb19.db
1      ONLINE   ONLINE   stbm000042-vm1    Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
2      ONLINE   ONLINE   stbm000042-vm2    Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
3      ONLINE   ONLINE   stbm000042-vm3    Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
4      ONLINE   ONLINE   stbm000042-vm4    Open,HOME=/u01/app/oracle/
product/19c/dbhome,STABLE
```

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 [...]] | -w "filter"]
```

Parameters

Table E-45 crsctl start cluster Command Parameters

Parameter	Description
`-all`	Use this parameter to start the Oracle Clusterware stack on all servers in the cluster.
`-n server [...]`	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`.
`-w "filter"`	Optionally, you can use this parameter to start only servers that match a filter. **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 [...]] | -w "filter"] [-f]
```

Parameters

Table E-46 crsctl stop cluster Command Parameters

Parameter	Description
`-all`	Use this parameter to start the Oracle Clusterware stack on all servers in the cluster.
`-n server [...]`	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`.
`-w "filter"`	Optionally, you can use this parameter to stop only servers that match a filter. **See Also:** Filters for more information about filters.

Usage Notes

- You must run this command as `root`.

- 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 mode

Ascertains 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-47 crsctl modify cluster site Command Parameters

Parameter	Description
site_name	Specify the name of the site you want to modify.
-n *node1, node2, ...*	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.
-s {rejuvenate | quarantine}	Specify a state for the site. • `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	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.

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
```

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.
```

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'
```

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 `WORKLOAD_CPU` resources to the value of `fill`.
`spread`	Sets the placement policy for `WORKLOAD_CPU` resources to the value of `spread`.

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
```

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-48 crsctl start crs Command Parameters

Parameter	Description
`-excl [-nocrs] [-cssonly]`	Starts Oracle Clusterware in exclusive mode with two options: • Specify the `-nocrs` parameter to start Oracle Clusterware in exclusive mode without starting CRSD. • Specify the `-cssonly` parameter to start CSSD, only. **Note:** While running in the exclusive mode, if Oracle Clusterware detects that another node is running, then the exclusive mode aborts to preserve voting file integrity.
`-wait | -waithas | -nowait`	Choose one of the following: • Specify `-wait` to wait until startup is complete and display all progress and status messages. • Specify `-waithas` to wait until startup is complete and display OHASD progress and status messages. • Specify `-nowait` to not wait for OHASD to start

Table E-48 (Cont.) crsctl start crs Command Parameters

Parameter	Description
`-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.

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 [23.0.0.0.0]. The cluster
upgrade state is [NORMAL]. The cluster active patch level is [456789126].
```

States

The following table lists and describes Oracle Clusterware upgrade states that are displayed by the `crsctl query crs activeversion -f` command.

Table E-49 Oracle Clusterware Upgrade State Descriptions

State	Description
`NORMAL`	Oracle Clusterware upgrade has completed successfully and the cluster is ready for next patching or upgrade cycle.
`FORCED`	Oracle Clusterware upgrade was completed forcibly on all reachable nodes, while one or more cluster nodes were unreachable. You can either upgrade or delete the unreachable cluster nodes to change the upgrade state from `FORCED` to `NORMAL`.
`ROLLING PATCH`	Oracle Clusterware is in the middle of a rolling patch.
`ROLLING UPGRADE`	Oracle Clusterware is in the middle of a rolling upgrade.
`UPGRADE FINAL`	Oracle Clusterware upgrade and the `root` script execution has completed successfully, and the installer is performing the final upgrade steps.

Table E-49 (Cont.) Oracle Clusterware Upgrade State Descriptions

State	Description
`FORCED UPGRADE FINAL`	Oracle Clusterware upgrade and the `root` script execution has completed successfully on all reachable nodes, while one or more cluster nodes were unreachable. The installer is performing the final upgrade steps.
`ROLLING PATCH DURING ROLLING UPGRADE`	Oracle Clusterware is being patched in the middle of a rolling upgrade.
`UPGRADE AV UPDATED`	Oracle Clusterware active version has updated to latest version and the `root` script is performing the final upgrade steps.
`FORCED UPGRADE AV UPDATED`	Oracle Clusterware active version has updated forcibly to latest version on all reachable nodes, while one or more nodes were unreachable. The `root` script is performing the final upgrade steps.

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-50 crsctl set crs autostart Command Parameters

Parameter	Description
`delay` *`delay_time`*	Specify the number of seconds to delay Oracle Clusterware autostart.
`servercount` *`count`*	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
```

Examples

This command returns output similar to the following for a node which has no patches applied:

```
Oracle Clusterware release patch level is [0] and no patches have been
applied on the local node. The release
patch string is [23.6.0.24.08].
```

Example

This command returns output similar to the following for a node which has patches applied:

```
Oracle Clusterware release patch level is [2879017762] and the complete list
of patches [36551085 36741532
36744688 36748658 36749264] have been applied on the local node. The release
patch string is [23.5.0.24.07].
```

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 Availablity Services release version on the local node is
[23.0.0.0.0]
```

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 Grid Infrastructure installed on that particular host. Otherwise, CRSCTL displays the patch level of Oracle Grid Infrastructure 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 [23.0.0.0.0]
```

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
- You must restart Oracle ASM after pinning a node to update the `/etc/oratab` entries.
- 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 or a later release, 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 or a later release 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 `ipmitool` or `ipmiutil` binary.

Syntax

```
get ipmi binaryloc
```

Usage Notes

Run the command to retrieve the `ipmitool` or `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 `ipmitool` or `ipmiutil` binary location on a node:

```
crsctl get ipmi binaryloc
```

crsctl set ipmi binaryloc

Sets the location of the `ipmitool` or `ipmiutil` binary.

Syntax

```
set ipmi binaryloc binary_location
```

Usage Notes

Run the command to set the `ipmitool` or `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 `ipmitool` or `ipmiutil` binary location on a node:

```
crsctl set ipmi binaryloc /usr/bin/ipmitool
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 run under the account used to install Oracle Clusterware, or an authorization failure may be reported.
- An authorization failure may not result when run 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 add css votedisk

Use the `crsctl add css votedisk` command to add one or more voting files to the cluster on storage devices other than an Oracle ASM disk group.

Syntax

```
crsctl add css votedisk vdisk [...] [-purge]
```

Parameters

Table E-51 crsctl add css votedisk Command Parameters

Parameter	Description
vdisk `[...]`	Specify a fully qualified path of the voting file you want to add. Specify multiple voting file paths in a space-delimited list.
`-purge`	Specify the `-purge` option to remove all existing voting files at once. You can replace the existing set of voting files in one operation.

Usage Notes

- You should have at least three voting files, unless you have a storage device, such as a disk array, that provides external redundancy. Oracle recommends that you do not use more than 5 voting files. The maximum number of voting files that is supported is 15.

Example

To add a voting file to the cluster:

```
$ crsctl add css votedisk /stor/grid/ -purge
```

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-52 crsctl delete css votedisk Command Parameters

Parameter	Description
voting_disk_GUID [...]	Specify the file universal identifier (GUID) of the voting file you want to remove. Specify multiple GUIDs in a space-delimited list.
vdisk [...]	Alternatively, specify the path of the voting file you want to remove. Specify multiple voting file paths in a space-delimited list.
+diskgroup	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 26f7271ca8b34fd0bfcdc2031805581e
```

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 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-53 crsctl discover dhcp Command Parameters

Parameter	Description
`-clientid clientid`	Specify the client ID for which you want to attempt discovery. Obtain the client ID by running the crsctl get clientid dhcp command.
`-port port`	Specify the port to which CRSCTL sends the discovery packets.
`-subnet subnet`	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-54 crsctl get clientid dhcp Command Parameters

Parameter	Description
`-cluname cluster_name`	Specify the name of the cluster where the cluster resources are configured.
`-viptype vip_type`	Specify the type of the VIP resource for which you want to display client IDs: `HOSTVIP`, `SCANVIP`, or `APPVIP`.
`-vip vip_res_name`	Specify the name of the VIP resource. This parameter is required if you specify the `APPVIP` VIP type.
`-n node_name`	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-55 crsctl release dhcp Command Parameters

Parameter	Description
`-clientid clientid`	Specify the client ID for which you want to attempt release. Obtain the client ID by running the `crsctl get clientid` command.
`-port port`	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-56 crsctl request dhcp Command Parameters

Parameter	Description
`-clientid clientid`	Specify the client ID for which you want to attempt request. Obtain the client ID by running the `crsctl get clientid` command.
`-port port`	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 Grid Infrastructure 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-57 crsctl query dns Command Parameters

Parameter	Description
`-name` *name*	Specify the fully-qualified domain name that you want to query.
`-dnsserver` *DNS_server_address*	Optionally, you can specify the address of the DNS server on which you want the domain name to be queried.
`-type` *query_type*	Optionally, you can select either IPv4 or IPv6 by specifying either `A` or `AAAA`, respectively, as a query type.
`-port` *port*	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.
`-attempts` *number_of_attempts*	Optionally, you can specify the number of retry attempts.
`-timeout` *timeout_in_seconds*	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.example.com`:

```
CRS-10024: DNS server returned 192.168.29.250 for name
stmjk07-vip.stmjk0462.example.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-58 crsctl start ip Command Parameters

Parameter	Description
IP_name | *IP_address*	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.
/netmask/ interface_name	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-59 crsctl stop ip Command Parameters

Parameter	Description
IP_name | *IP_address*	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.
interface_name	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.example.com/eth0
```

crsctl lsmodules

Lists the components of the modules that you can debug.

Syntax

```
crsctl lsmodules {mdns | gpnp | css | crf | crs | evm | gipc}
```

Usage Notes

You can obtain a list of components for any of the following modules:

- `mdns`: Multicast domain name server
- `gpnp`: Grid Plug and Play service
- `css`: Cluster Synchronization Services
- `crf`: Cluster Health Monitor
- `crs`: Cluster Ready Services
- `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
  Setting log levels for Oracle Clusterware.

crsctl delete node

Removes a node from the cluster.

Syntax

```
crsctl delete node -n node_name [-purge]
```

Usage Notes

- You must be `root` to run this command
- The *node_name* parameter is required
- Use the `-purge` option to delete the node permanently and to reuse its node number. However, if you want to add the deleted node back with the same node name and the same node number, then do not use the `-purge` option.

Example

To delete the node named `node06` from the cluster, run the following command as `root`:

```
# crsctl delete node -n node06
```

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-60 crsctl check resource Command Parameters

Parameter	Description
resource_name `[...]`	Specify a particular resource. You can check multiple resources by entering a space-delimited list of resource names.

Table E-60 (Cont.) crsctl check resource Command Parameters

Parameter	Description
`-w "filter"`	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: • `"TYPE == cluster_resource"`: This filter limits Oracle Clusterware to check only resources of `cluster_resource` 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 *and* the name of the resource contains the number `2` **See Also:** "Filters" for more information
`-n node_name`	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 cardinality_id`	Specify the resource cardinality ID.
`-d degree_id`	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-61 crsctl relocate resource Command Parameters

Parameter	Description
resource_name [-k *cid*]	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.
resource_name | -all -s *source_server*	Alternatively, you can specify one particular resource or all resources located on a particular server that you specify *from* which you want to relocate those resources.
-w "*filter*"	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: • "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 *and* the name of the resource contains the number 2 **See Also:** "Filters" for more information
-n *destination_server*	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 "*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 ("").
-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 resources on the server node.

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 resources on the server node.
- 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.

> **Note:**
>
> `crsctl start rollingpatch` and `crsctl stop rollingpatch` should not be run directly except under the direction of Oracle Support. These commands are invoked as part of `rootcrs.pl -prepatch`/`-postpatch` which orchestrates patching on each node.

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.

> **Note:**
>
> `crsctl start rollingpatch` and `crsctl stop rollingpatch` should not be run directly except under the direction of Oracle Support. These commands are invoked as part of `rootcrs.pl -prepatch`/`-postpatch` which orchestrates patching on each node.

After 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.

> **Note:**
>
> This command should not be invoked directly except under the direction of Oracle Support. It is invoked by the `rootupgrade` scripts as part of the upgrade.

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.
- 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-62 crsctl eval add server Command Parameters

Parameter	Description
server_name	The name of the server you want to add.
-file *file_path*	Fully-qualified path to a file containing server attributes.
-attr "*attr_name=attr_value*[,...]"	Alternatively, you can 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 *attr_name=attr_value* 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=25"
-admin [-l *level*] [-x] [-a]	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 output levels: • resources: Restricts the output to resources running on server nodes • 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.

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'.
           Scanning server pools with MIN_SIZE or more servers in
           descending order of IMPORTANCE.
           Considering server pools (IMPORTANCE): Free(0) for relocating
           servers.
```

```
        E  Relocating server 'scott_node5_0' to server pool 'Free'.
        Y  Server 'scott_node5_0' will be added to pools 'Free'.
--------------------------------------------------------------------------------
```

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-63 crsctl eval delete server Command Parameters

Parameter	Description
server_name	Specify the name of the server you want to evaluate before deleting.
`-admin [-l level] [-x] [-a]`	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 output levels: • `resources`: Restricts the output to resources running on server nodes • `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.

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_node1_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_node1_2' to server pool 'sp1'.
        Y   Server 'scott_node1_2' will be moved from pool 'sp2' to pool
            'sp1'.
        Y   Server 'scott_node1_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_node1_2'.
------------------------------------------------------------------------
```

crsctl modify server

Modifies the configuration of a particular server.

Syntax

```
crsctl modify server server_name -attr "attr_name=attr_value[,...]"
```

Parameters

Table E-64 crsctl modify server Command Parameters

Parameter	Description
server_name	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 *attr_name=attr_value* 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 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-65 crsctl status server Command Parameters

Parameter	Description
server_name [...]	Specify one or more space-delimited server names.
-w "*filter*"	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. **See Also:** "Filters" for more information
-g | -p | -v | -f	You can specify one of the following parameters when Oracle Clusterware checks the status of specific servers: • -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 *category_name*	Optionally, you can specify a particular category of servers for which to obtain status.
-w "*filter*"	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. **See Also:** "Filters" for more information

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
RESOURCE_USE_ENABLED=1
SERVER_LABEL=
PHYSICAL_HOSTNAME=
STATE=ONLINE
ACTIVE_POOLS=ora.pool1
STATE_DETAILS=
```

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 HRserver
```

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 `HRserver`.
- You must restart the Oracle Clusterware technology stack on the node before any changes you make take effect.

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-66 crsctl query socket udp Command Parameters

Parameter	Description
`-address` *address*	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.
`-port` *port*	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-67 crsctl start testdns Command Parameters

Parameter	Description
`-address` *address*	Specify a server address in the form *IP_address*/*netmask*[/*interface_name*].
`-port` *port*	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-68 crsctl status testdns Command Parameters

Parameter	Description
`-server DNS_server_address`	Optionally, you can specify the DNS server address for which you want to check the status.
`-port port`	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.example.com
```

crsctl stop testdns

Stops a test DNS server.

Syntax

```
crsctl stop testdns [-address address] [-port port] [-v]
```

Parameters

Table E-69 crsctl stop testdns Command Parameters

Parameter	Description
`-address address`	Optionally, you can specify the server address for which you started the test DNS server in the form `IP_address/netmask[/interface_name]`.
`-port port`	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-69 (Cont.) crsctl stop testdns Command Parameters

Parameter	Description
`-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-70 crsctl replace votedisk Command Parameters

Parameter	Description
+asm_disk_group	Specify the Oracle ASM disk group (preceded by a plus sign (+))where you want to locate the voting file.
path_to_voting_disk [...]	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 Grid Infrastructure 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-9 Example 1

To replace a voting file that is located within Oracle ASM:

```
# crsctl replace votedisk +diskgroup1
```

Example E-10 Example 2

To replace a voting file that is located on a shared file system:

```
# crsctl replace votedisk /mnt/nfs/disk1 /mnt/nfs/disk2
```

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-71 crsctl add wallet Command Parameters

Parameter	Description
`-type wallet_type`	Specify the type of wallet you want to create, such as OSUSER or CVUDB. • **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 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.
`-name name`	You must specify a name for the wallet to create APPQOSDB and CVUDB wallets.
`-user user_name -passwd`	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

The 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.

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-72 crsctl delete wallet Command Parameters

Parameter	Description
`-type wallet_type`	Type of wallet you want to remove, such as OSUSER or CVUDB. • **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 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.
`-name name`	You must specify the name of the wallet to remove an APPQOSDB wallet.
`-user user_name`	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 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-73 crsctl modify wallet Command Parameters

Parameter	Description
`-type wallet_type`	Specify the type of wallet you want to create, such as OSUSER or CVUDB. • **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 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.
`-name name`	You must specify the wallet name to modify an APPQOSDB wallet.
`-user user_name -passwd`	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-74 crsctl query wallet Command Parameters

Parameter	Description
`-type wallet_type`	Type of wallet you want to query, such as OSUSER or CVUDB. • **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 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.
`-name name`	You must specify the name of the wallet to query an APPQOSDB wallet.
`-user user_name`	You must specify a user name to query a user from an OSUSER wallet.

Table E-74 (Cont.) crsctl query wallet Command Parameters

Parameter	Description
`-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 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
[23.0.0.0.0]
```

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 [23.0.0.0.0]
```

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.

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
  `gpnp`: Grid Plug and Play service
  `css`: Cluster Synchronization Services
  `crf`: Cluster Health Monitor
  `crs`: Cluster Ready Services
  `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 11*g* 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
  `gpnp`: Grid Plug and Play service
  `css`: Cluster Synchronization Services
  `crf`: Cluster Health Monitor
  `crs`: Cluster Ready Services
  `evm`: Event Manager
  `gipc`: Grid Interprocess Communication

- *component*: Specify the name of a component for one of the modules. See Table E-75 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 | gpnp | css | crf | crs | evm | gipc}
```

> **Note:**
>
> You do not have to be the `root` user to run the `crsctl lsmodulues` command.

Table E-75 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-75 Components for the CRS, CSS, and EVM Modules

CRS Components[1]	CSS Components[2]	EVM Components[3]
CRSUI	CSSD	EVMD
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-76 describes the Cluster Synchronization Services modules.

Table E-76 Cluster Synchronization Services (CSS) Modules and Functions

Module	Description
CSS	CSS client component
CSSD	CSS daemon component

Table E-77 describes the function of each communication (COMM) module.

Table E-77 Communication (COMM) Modules and Functions

Module	Description
COMMCRS	Clusterware communication layer
COMMNS	NS communication layer

Table E-78 describes the functions performed by each CRS module.

Table E-78 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-79. The components listed in Table E-79 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-79 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-80 describes the OCR tool modules.

Table E-80 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

Server Control (SRVCTL) Command Reference

Use the Server Control (SRVCTL) utility to manage various components and applications in your cluster.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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*. Starting with Oracle Database 21c, PDBs are a resource managed by Oracle Clusterware. You can use SRVCTL commands to manage PDBs.

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.

- Use the version of SRVCTL that is provided with the current Oracle Database release from the Oracle home of the database that you are managing. The version of SRVCTL must be the same as the version of the object (listeners, Oracle ASM instances, Oracle Database, Oracle Database instances, and Oracle Database services) being managed.
- SRVCTL does not support concurrent executions of commands on the same object. 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.

When you specify a comma-delimited list in Windows environments, 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.
- SRVCTL returns `0` on success, `1` on failure, and `2` on warnings. Some commands, such as `start`, `stop`, `enable`, and `disable`, can return 2 for a warning when the request would not change anything. In other words, the object of the command is already started, already stopped, already disabled, and so on. In warning cases, SRVCTL also prints a message about what was already done.

Specifying Command Parameters as Keywords Instead of Single Letters

The use of single letter commands is deprecated. Oracle recommends that you use full command words with SRVCTL.

In releases earlier than Oracle Database 12c, the `SRVCTL` command-line interface used single letter parameters. However, single letter parameters impose a limit on the number of unique parameters available for use with SRVCTL commands. SRVCTL command parameters in current Oracle Database releases use 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:**
>
> The use of single letter parameters are deprecated. Oracle recommends that you use the keyword parameters, so that you 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 F-1 String Restrictions for SRVCTL Object Names

Object Type	Character Set Limitations	Case Sensitive?	Maximum Length
db_domain	Alpha-numeric characters, underscore (_), and number sign (#)	No	128 characters
db_unique_name	Alpha-numeric characters, underscore (_), number sign (#), and dollar sign ($); the first 8 characters must be unique because those characters are used to form instance names	No	30 characters but the first 12 characters must be unique relative to any other database in the same cluster
pdb_name	Alpha-numeric characters and underscore (_); the first character must be an alphabet character	No	30 characters
diskgroup_name	Naming disk groups have the same limitations as naming other database objects.	No (all names are converted to uppercase)	30 characters
instance_name	Alphanumeric characters	Depends on the platform	15 characters
listener_name	Alphanumeric characters	Depends on the platform	15 characters
node_name	Alphanumeric characters	No	15 characters
scan_name	The first character must be an alphabetic character	No	15 characters
service_name	Alphanumeric characters, underscore (_), number sign (#), period (.), and dollar sign ($); the name cannot begin with a period, nor contain single quotation marks (' ')	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

Learn about how to use context sensitive help with SRVCTL commands.

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, to run the 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. In this appendix, those commands are preceded by the root prompt (`#`) in the command examples.

Additional SRVCTL Topics

You can use SRVCTL to manage Oracle-supplied resources, but Oracle strongly advises you to follow the guidelines provided here.

- Use SRVCTL to manage Oracle-supplied resources such as listener, instances, disk groups, and networks, and CRSCTL for managing Oracle Clusterware and its resources.

> **Caution:**
>
> Oracle strongly discourages you from using CRSCTL to directly manipulate Oracle-supplied resources (resources whose names begin with `ora`). Making manual changes to Oracle resources using CRSCTL can adversely affect the cluster configuration.

- Although you may be able to cancel running SRVCTL commands by pressing the Control-C keys, Oracle strongly advises that you do not attempt to do this, because you can corrupt your configuration data by doing this.

  Do not to attempt to terminate SRVCTL in this manner.

Deprecated SRVCTL Subprograms or Commands

Oracle recommends that you use alternatives for several SRVCTL commands and parameters deprecated with Oracle Database 12c.

Single Character Parameters Deprecated for all SRVCTL Commands

Single-character parameters were deprecated in Oracle Database 12c. Use the full keyword for each parameter. Refer to the information here to understand how to update scripts using single-character parameters.

Oracle recommends that you use the full keyword for each SRVCTL parameter. To support older tools and scripts that still use single-character parameters, the current version of SRVCTL continues to support both single-character parameters and full keyword parameters. However, deprecated functionality can be desupported in a future release.

The command reference topics for SRVCTL show the keywords for each SRVCTL command. The following table lists the deprecated single-character parameters.

Table F-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`	none	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`	None	Print detailed configuration information	Common
`a`	`available`	*`available_list`*	A comma-delimited list of available instances	Service commands

Table F-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
a	abort	None	Stop failed online relocation	Relocate database
a	viponly	None	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 commands
c	currentnode	*current_node*	Node name from which to relocate the service	Service commands
c	cardinality	{UNIFORM| SINGLETON}	Whether the service should run on every active server (UNIFORM) or only one server (SINGLETON)	Service commands
c	dbtype	*type*	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	*db_unique_name*	Database unique name	Common
d	device	*volume_device*	Volume device path	Database, instance, Oracle ASM, disk group, and file system commands
d	domain	None	Display subdomain served by GNS	OC4J, home, CVU, and GNS commands
e	emport	*em_port_number*	Local listen port for Oracle Enterprise Manager	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
e	failovertype	{NONE|SESSION BASIC| TRANSACTION}	The failover type for a service	Service commands
e	server	*server_list*	Candidate server list for Oracle RAC One Node database	Database, instance, Oracle ASM, disk group, and file system commands
f	force	None	Force remove	Common
g	diskgroup	*diskgroup_name*	Disk group name	File system, Diskgroup commands
h	help	None	None	Common

Table F-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
i	instance	*instance_name* *instance_list*	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	*ip_address*	VIP address on which GNS is to listen	OC4J, home, CVU, and GNS commands
i	oldinst	*instance_name*	The old instance name	Service commands
i	scannumber	*scan_ordinal_number*	Ordinal number of the IP address for the SCAN	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
i	vip	*vip_name* or "*vip_name_list*"	VIP names	Node applications, GNS, VIP, network, listener, SCAN VIP, and SCAN listener commands
j	acfspath	*acfs_path_list*	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 commands
k	netnum	*network_number*	The network number	Service 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	*listener_name*	The name of a listener	ASM commands
l	loglevel	*log_level*	Specify the level (0-6) of logging that GNS should run with	OC4J, home, CVU, and GNS commands
l	onslocalport	*port_number*	Oracle Notification Service listening port for local client connections	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands

Table F-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
l	role	*service_role*	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 commands
m	domain	domain_name	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 commands
m	multicastpost		The port on which the GNS daemon is listening for multicast requests	OC4J, home, CVU, and GNS commands
m	path	*mountpoint_path*	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	*node_name*	The name of a specific node	Common
n	nodes	*node_list*	A comma-delimited list of node names	File system commands
n	dbname	*database_name*	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	scan_name	Fully-qualified SCAN name (includes the domain)	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
n	servers	*server_list*	A comma-delimited list of candidate server names	Service commands
n	targetnode	*node_name*	Node name to which to relocate the service	Service commands
o	oraclehome	*oracle_home*	*$ORACLE_HOME* path	Database commands
p	endpoints	[TCP:]*port_number*[/IPC:*key*][/NMP:*pipe_name*][/TCPS:*s_port*][/SDP:*port*]	SCAN listener endpoints	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands

Table F-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
p	port		The port which the GNS daemon uses to communicate with the DNS server	OC4J, home, CVU, and GNS commands
p	rmiport	*port_number*	OC4J RMI port number	OC4J, home, CVU, and GNS commands
P	tafpolicy	{NONE|BASIC}	TAF policy specification	Service commands
p	spfile	*spfile_location*	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	*preferred_list*	A comma-delimited list of preferred instances	Service commands
r	onsremoteport	*port_number*	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	None	Remove target node of failed online relocation request from the candidate server list of administrator-managed Oracle RAC One Node database	Relocate database
r	role	*role_type*	Role of the standby database: `PRIMARY`, `PHYSICAL_STANDBY`, `LOGICAL_STANDBY`, or `SNAPSHOT_STANDBY`	Database, instance, Oracle ASM, disk group, and file system commands
s	onsonly		Display Oracle Notification Service daemon configuration	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
s	skip	None	Skip checking the ports	Listener, SCAN, and SCAN listener.
s	statfile	*file_name*	The file path of the *state_file* created by a previously run `srvctl stop home` command	OC4J, home, CVU, and GNS commands

Table F-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
s	status		Display the status of GNS	OC4J, home, CVU, and GNS commands
S	subnet	*subnet*/*net_mask*/[*if1*[|*if2*...]]	Network address specification for a network	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
s	service	*service_name* *service_name_list*	The name of a service A comma-delimited list of service names	Service commands
s	startoption	*start_options*	Startup options for the database (mount, open, read only)	Database, instance, Oracle ASM, disk group, and file system commands
t	checkinterval	*time_interval*	Interval in minutes between checks	OC4J, home, CVU, and GNS commands
t	edition	*edition_name*	The initial session edition of a service	Service commands
t	envs	"*name_list*"	A list of environment variables	Common
t	namevals	"*name= value*,..."	Names and values of environment variables	Common
T	nameval	"*name=value*"	Name and value of a single environment variable	Common
t	update	*instance_name*	The new instance name	Service commands
t	remoteservers	*host_name*[:*port_number*][,*host_name*[:*port_number*]...]	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	*stop_options*	Stop options for the database (NORMAL, TRANSACTIONAL, IMMEDITATE, or ABORT)	Database, instance, Oracle ASM, disk group, and file system commands
t	toversion	*target_version*	Version to which you are downgrading	Database, instance, Oracle ASM, disk group, and file system commands
u	nettype	*network_type*	The network server type, which can be STATIC, DHCP, or MIXED	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
u	newinst	None	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

Table F-2 (Cont.) Deprecated Single-Character Parameters for SRVCTL Commands

Single Letter	Long Form	Values	Description	Related Commands
u	user	*oracle_user*	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	*volume_name*	Name of a volume	Database, instance, Oracle ASM, disk group, and file system commands
V	versions			Common
w	failoverdelay	*number*	Failover delay	Service commands
w	nettype	*network_type*	The network server type, which can be `STATIC`, `DHCP`, or `MIXED`	Node applications, VIP, network, listener, SCAN VIP, and SCAN listener commands
w	timeout	*timeout*	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 commands
x	node	*node_name*	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, and service commands
z	failoverretry	*number*	Number of failover retries	Service commands
z	rmdependisk		To remove a database's dependency upon disk groups	Database, instance, Oracle ASM, disk group, and file system commands

Miscellaneous SRVCTL Commands and Parameters

If you have scripts dating from Oracle Database 12.2 or earlier releases, then Oracle recommends that you review the deprecated parameters and update your usage to current forms.

The following command parameters were deprecated in Oracle Database 12c:

Table F-3 Deprecated Commands and Parameters for SRVCTL

Command	Deprecated Parameters
`srvctl modify asm`	`-node node_name`
`srvctl modify instance`	`-z` Instead, use the `-node` option with the value set to `""`
`srvctl modify gns`	`[-ip ip_address] [-advertise host_name -address address] [-delete host_name -address address] [-createalias name -alias alias] [-deletealias alias]` Use the `srvctl update gns` command instead.
`srvctl * oc4j`	The `oc4j` noun has been deprecated. SRVCTL still accepts the `oc4j` noun until it is desupported.
`srvctl add service`	The `PRECONNECT`option with the `-tafpolicy` parameter is deprecated.
`srvctl modify service`	The `-failovermethod {NONE | BASIC}` is deprecated. The `PRECONNECT`option with the `-tafpolicy` 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 Automatic Storage Management (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 (`""`).
> - Starting with Oracle Database Release 23ai, Oracle Quality of Service Management (QoSM, or QoS Management) is desupported.

Table F-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 F-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
Configuration Verification Utility	`cvu`	To add, modify, enable, disable, remove, relocate, start, stop, and obtain the status of the Configuration Verification Utility
Cross-cluster dependency	`cdp`	To add, modify, enable, disable, remove, relocate, start, stop, and obtain the status of cross-cluster dependency services.
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 Advanced 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, 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 IOServer	`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 IOServer
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
Fleet Patching and Provisioning Client	`rhpclient`	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 config all

Displays the configuration details of the cluster and database components.

Syntax

```
srvctl config all [-help]
```

Example

This command returns output similar to the following:

```
$ srvctl config all

Oracle Clusterware configuration details
=========================================
Oracle Clusterware basic information
-----------------------------------
Operating system       Linux
Name                   stbm000025-rac3
Class                  STANDALONE
Cluster nodes          stbm000025-vm11, stbm000025-vm10, stbm000025-vm9,
                       stbm000025-vm12
Version                23.0.0.0.0
Groups                 SYSOPER:oinstall SYSASM:oinstall SYSRAC:oinstall
                       SYSDBA:oinstall
OCR locations          +DATA2
Voting disk locations  DATA2
Voting disk file paths AFD:DATA21

Cluster network configuration details
-------------------------------------
Interface name  Type  Subnet          Classification
ens3            IPV4  100.70.0.0/20   PUBLIC
ens4            IPV4  192.168.1.0/24  PRIVATE, ASM

SCAN configuration details
--------------------------
SCAN "stbm000025-scan3" details
++++++++++++++++++++++++++++++
Name                stbm000025-scan3
IPv4 subnet         100.70.0.0/20
DHCP server type    static
End points          TCP:1521

SCAN listeners
--------------
Name              VIP address
LISTENER_SCAN1    100.70.5.26
LISTENER_SCAN2    100.70.5.20
LISTENER_SCAN3    100.70.5.22

ASM configuration details
-------------------------
```

```
Mode            remote
Password file   +DATA2
SPFILE          +DATA2

ASM disk group details
++++++++++++++++++++++
Name   Redundancy
DATA2  EXTERN

Database configuration details
==============================
Database "ora.orcl.db" details
------------------------------
Name                ora.orcl.db
Type                RAC
Version             23.0.0.0.0
Role                PRIMARY
Management policy   AUTOMATIC
SPFILE              +DATA2
Password            +DATA2
file Groups         OSDBA:oinstall OSOPER:oinstall OSBACKUP:oinstall
OSDG:oinstall
                    OSKM:oinstall OSRAC:oinstall
Oracle home         /scratch/app/23ai/220523/dbhome
```

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 F-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 F-5 (Cont.) srvctl add asm Command Parameters

Parameter	Description
`-flex [-count {number_of_instances | ALL}] | -proxy [-spfile server_parameter_file_path]`	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 and later 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 F-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

To manage Oracle ASM on Oracle Database 12c and later 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/23.0.0/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 F-7 srvctl disable asm Command Parameters

Parameter	Description
`-proxy`	Optionally, you can use this parameter to specify whether you are disabling a proxy Oracle ASM configuration.
`-node node_name`	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 and later 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 F-8 srvctl enable asm Command Parameters

Parameter	Description
`-proxy`	Optionally, you can use this parameter to specify whether you are enabling a proxy Oracle ASM configuration.
`-node node_name`	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 and later 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 and later 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] [-pwfile password_file_path]
  [-pwfilebackup backup_password_file_path] [-count {number_of_instances |
ALL}] [-diskstring asm_diskstring]
  [-spfile spfile_path_name [-proxy] [-force]]
```

Parameters

Parameter	Description
`-listener listener_name`	Specify the name of the listener with which Oracle ASM registers.
`-pwfile password_file_path`	Specify the full path to the location of the password file for Oracle ASM.
`-pwfilebackup backup_password_file_path`	Specify this option to update the location of the Oracle ASM password backup file. **Note:** This option does not backup the Oracle ASM password file.
`-count {number_of_instances | ALL}`	Specify the number of Oracle ASM instances to create, or that an Oracle ASM instance should be created on all nodes.
`-diskstring asm_diskstring`	Specify a new Oracle ASM disk group discovery string.

Parameter	Description
`-spfile` *spfile_path_name*	Specify the path to the new spfile to be used by Oracle ASM.
`-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 cluster node to another.

Syntax

```
srvctl relocate asm -currentnode current_node [-targetnode target_node] [-
force]
```

Parameters

Table F-9 srvctl relocate asm Parameters

Parameter	Description
`-currentnode` *current_node*	Name of the node on which the Oracle Flex ASM instance currently resides.

Table F-9 (Cont.) srvctl relocate asm Parameters

Parameter	Description
`-targetnode` *target_node*	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 `node2` is:

```
$ srvctl relocate asm -currentnode node1 -targetnode node2
```

srvctl remove asm

Removes the Oracle ASM resource from Oracle Clusterware management.

> **Note:**
>
> To manage Oracle ASM on Oracle Database 12c and later 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 F-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.
`-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 F-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 and later 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 F-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 node_name`	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 start_options`	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:** *SQL*Plus User's Guide and Reference* 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 and later 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 F-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` *`node_name`*	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
Detailed 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 F-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.

Table F-14 (Cont.) srvctl stop asm Command Parameters

Parameter	Description
`-node node_name`	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 stop_options`	Optionally, you can specify options to the shutdown command, such as `NORMAL`, `TRANSACTIONAL`, `IMMEDIATE`, or `ABORT`. **See Also:** *SQL*Plus User's Guide and Reference* 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 and later 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/if1[|if2...]]
```

Parameters

Table F-15 srvctl add asmnetwork Command Parameters

Parameter	Description
`-netnum asm_network_number`	Optionally, you can specify an Oracle ASM network number (default number is 1) that you want to add.
`-deptype {optional | mandatory}`	Specify whether the Oracle ASM dependency type is optional or mandatory. The default is `optional`.
`-subnet subnet[/netmask/if1[|if2...]]`	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, *netmask* 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 F-16 srvctl modify asmnetwork Command Parameter

Parameter	Description
`-netnum asm_network_number`	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 F-17 srvctl remove asmnetwork Command Parameters

Parameter	Description
`-netnum asm_network_number | -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]
[-passfile_events efile] [-allow ip/network,...]
```

Parameters

Table F-18 srvctl add cdp Command Parameters

Parameter	Description
`-port port_number`	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 cluster.

Table F-18 (Cont.) srvctl add cdp Command Parameters

Parameter	Description
`-passfile_admin afile`	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.
`-passfile_readonly rfile`	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.
`-passfile_events efile`	If you have a password file for events users, then you can specify it using this parameter. If no password file exists, then SRVCTL writes a randomly-generated password to a file.
`-allow ip/network`	The IP and network mask pair defined in CIDR format.

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]
[-passfile_events efile] [-allow ip/network,...]
```

Parameters

Table F-19 srvctl modify cdp Command Parameters

Parameter	Description
`-port port_number`	Optionally, you can modify the port on which all cross-cluster dependency proxy resources in the cluster listen.
`-passfile_admin afile`	Optionally, you can modify the password information for administrators.
`-passfile_readonly rfile`	Optionally, you can modify the password information for read-only users.
`-passfile_events efile`	Optionally, you can modify the password information for events users.
`-allow ip/network`	Optionally, you can modify the IP and network mask pair defined in CIDR format.

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 F-20 srvctl relocate cdp Command Parameters

Parameter	Description
`-cdpnumber cdp_ordinal_number`	Specify a cross-cluster dependency proxy resource that you want to relocate by specifying its identifying ordinal number.

Table F-20 (Cont.) srvctl relocate cdp Command Parameters

Parameter	Description
`-node node_name`	Optionally, you can specify the name of a node on which a cross-cluster dependency proxy resource resides that you want to relocate.
`-force`	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 F-21 srvctl start cdp Command Parameters

Parameter	Description
`-cdpnumber cdp_ordinal_number`	Optionally, you can use this parameter to start a specific cross-cluster dependency proxy resource by specifying its identifying ordinal number.
`-node node_name`	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 cvu

Adds the Configuration 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 Configuration 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.

> **Note:**
>
> When you disable CVU on all cluster nodes, then local CVU configuration is overridden by the global configuration.

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 Configuration 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.

> **Note:**
>
> When you enable CVU on all cluster nodes, then local CVU configuration is overridden by the global configuration.

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.

> **Note:**
>
> The local and global status can be different, if CVU is enabled or disabled on selected nodes. Do not use the `-node` parameter to change CVU status globally.

Example

An example of this command to check the status of CVU on the node in the cluster named `crmnode1` is:

```
$ srvctl status cvu -node crmnode1
```

srvctl stop cvu

Stops the Configuration 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 F-22 srvctl add exportfs Command Parameters

Parameter	Description
`-name unique_name`	Enter a unique name for the export file system that you are creating. This parameter is required.
`-id havip_name`	Enter the number or name that identifies an existing HAVIP, that you specified in the `srvctl add havip` command. This parameter is required.
`-path path_to_export`	Enter the path to the Oracle Advanced Cluster File System (Oracle ACFS) that you want to export. This parameter is required.
`-clients export_clients -options export_options`	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: `-clients @netgroup1, *.example.com, 16.140.0.0, hostname1` **Note:** You can use the `-clients` parameter only for Linux systems. Use the `-options` parameter for Solaris and AIX systems.
`-type NFS | SMB`	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 F-23 srvctl config exportfs Command Parameters

Parameter	Description
`-name unique_name`	Enter the unique name of the NFS export for which you want to display the configuration.
`-id havip_name`	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 F-24 srvctl modify exportfs Command Parameters

Parameter	Description
`-name` *unique_name*	Enter the unique name of the NFS export file system to modify. This parameter is required.
`-path` *path_to_export*	You can modify the path to the Oracle Advanced Cluster File System (Oracle ACFS) where the export is located.
`-clients` *node_list*	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: `-clients @netgroup1,*.example.com,16.140.0.0, hostname1` **Note:** You can use the `-clients` parameter only for Linux systems. Use the `-options` parameter for Oracle Solaris and IBM AIX systems.
`-options` *nfs_client_string*	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/racdb1
```

srvctl remove exportfs

Removes the specified export file system configuration.

Syntax

```
srvctl remove exportfs -name exportfs_name [-force]
```

Parameters

Table F-25 srvctl remove exportfs Command Parameters

Parameter	Description
`-name` *exportfs_name*	Enter the unique name of the export file system you want to remove.
`-force`	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 F-26 srvctl start exportfs Command Parameters

Parameter	Description
`-name unique_name`	Enter the unique name of the NFS export you want to start.
`-id havip_name`	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 F-27 srvctl status exportfs Command Parameters

Parameter	Description
`-name unique_name`	Optionally, you can specify the unique name of the NFS export for which you want to display the status.
`-id havip_name`	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 F-28 srvctl stop exportfs Command Parameters

Parameter	Description
`-name unique_name`	Enter the unique name of the NFS export you want to stop.
`-id havip_name`	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 Advanced 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 | -volume volume_name -diskgroup
disk_group_name}
    -path mountpoint_path [-node "node_list"]
    [-user "user_list"] [-mountowner user_name] [-mountgroup group_name] [-
mountperm octal_permission]
    [-fstype {ACFS | EXT3 | EXT4}] [-fsoptions options] [-description
description]
    [-appid application_id] [-autostart {ALWAYS | NEVER | RESTORE}] [-
acceleratorvols volume_name]
```

> **Note:**
>
> Oracle supports the `EXT3` and `EXT4` values for the `-fstype` parameter only for Linux.

Parameters

Table F-29 srvctl add filesystem Command Parameters

Parameter	Description
`-device volume_device`	Specify the path to the file system volume device.
`-volume volume_name`	Specify the name of the volume.
`-diskgroup disk_group_name`	Specify the name of the disk group on which the `-volume` is located.
`-path mountpoint_path`	Specify the mount point path that is used to set up dependencies on other resources for nested mounts. This must be an absolute path.
`-node node_list`	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, then SRVCTL creates a single-node file system resource. Without any nodes, SRVCTL creates a file system resource that runs on every cluster node.

Table F-29 (Cont.) srvctl add filesystem Command Parameters

Parameter	Description
`-user` *user_list*	Optionally, you can specify a comma-delimited list of users authorized to mount and unmount the file system. If the file system is for Oracle Database software, then specify the name of the Oracle home owner. If you specify no users, then only `root` can mount and unmount the file system, or run `srvctl start filesystem` on the file system.
`-mountowner` *user_name*	Optionally, you can specify the user who is the mount point owner.
`-mountgroup` *group_name*	Optionally, you can specify the group that includes the mount point owners.
`-mountperm` *octal_permission*	Optionally, you can specify the permissions (OCTAL format) for the mount point.
`-fstype {ACFS | EXT3 | EXT4}`	Optionally, you can specify the type of file system to be mounted. The default value is `ACFS`. 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 `-node`.
`-fsoptions` *options*	Optionally, you can specify options used to mount the file system. The options are file system and operating system dependent.
`-description` *description*	Optionally, you can specify a description of the file system. This is stored as an attribute for later viewing by an administrator.
`-appid` *application_id*	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 `ora.id.fs.type`.
`-autostart {ALWAYS | NEVER | RESTORE}`	Optionally, you can specify the file system resource `autostart` policy. `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.
`-acceleratorvols` *volume_name*	Optionally, you can specify the accelerator volume. An accelerator volume is a secondary Oracle ACFS volume that is used by Oracle ACFS to store critical metadata.

Usage Notes

- Use the `-device` parameter to specify the path to the file system resource for which you want to add, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- 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 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.
- Standard Edition High Availability (SEHA) databases are supported only on Oracle Database 19c (19.7) release or later.

- This command is only available with Oracle Clusterware.

Examples

To add the `d1volume1-295` 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/d1volume1-295
  -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
```

To add an Oracle ACFS file system for configuring Standard Edition High Availability (SEHA) on the `sehome` volume of the `DATA` disk group with a mount point of `/u01/app/oracle`:

```
# srvctl add filesystem -volume sehome -diskgroup DATA -path /u01/app/oracle -
fstype ACFS
  -autostart ALWAYS -user oracle
```

srvctl config filesystem

Displays the configuration for a specific file system resource.

Syntax

```
srvctl config filesystem [-device volume_device | -volume volume_name -
diskgroup disk_group_name]
```

Parameters

Table F-30 srvctl config filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.

Usage Notes

- Use the `-device` parameter to specify the path to the file system resource for which you want to display the configuration or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- This command is only available with Oracle Clusterware.

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/d1volume1-295
```

srvctl disable filesystem

Disables a specific Oracle Clusterware-managed file system volume.

Syntax

```
srvctl disable filesystem {-device volume_device | -volume volume_name -
diskgroup disk_group_name}
```

Parameters

Table F-31 srvctl disable filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.

Usage Notes

- Use the `-device` parameter to specify the path to the file system resource for which you want to disable, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- This command is only available with Oracle Clusterware.

Example

An example of this command is:

```
# srvctl disable filesystem -device /dev/asm/racvol1
```

srvctl enable filesystem

Enables an Oracle ACFS volume or generic file system in Oracle Clusterware.

Syntax

```
srvctl enable filesystem {-device volume_device | -volume volume_name -
diskgroup disk_group_name}
```

Parameters

Table F-32 srvctl enable filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.

Usage Notes

- Use the `-device` parameter to specify the path to the file system resource for which you want to enable, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- This command is only available with Oracle Clusterware.

Example

An example of this command is:

```
# srvctl enable filesystem -device /dev/asm/racvol1
```

srvctl modify filesystem

Modifies the configuration of the file system resource.

Syntax

```
srvctl modify filesystem
    {-device volume_device | -volume volume_name -diskgroup disk_group_name}
    [-user {/+|/-}user_name | "user_list"]
    [-mountowner user_name] [-mountgroup group_name] [-mountperm
octal_permission]
    [-path mountpoint_path] [-node "node_list"]
    [-fsoptions options] [-description description]
    [-autostart {ALWAYS | NEVER | RESTORE}] [-force]
```

Parameters

Table F-33 srvctl modify filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.
`-user` *user_name* | "*user_list*"	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.

Table F-33 (Cont.) srvctl modify filesystem Command Parameters

Parameter	Description
`-mountowner` *user_name*	Modify the owner of the mount point with the specified value.
`-mountgroup` *group_name*	Modify the group of the mount point with the specified value.
`-mountperm` *octal_permission*	Modify the permissions of the user, group, and others with the specified value.
`-path` *mountpoint_path*	Modify the mount point path that is used to set up dependencies on other resources for nested mounts. This must be an absolute path.
`-node "`*node_list*`"`	Modify a comma-delimited list of nodes on which to mount the file system device. If you specify a list of nodes, then SRVCTL creates a single-node file system resource. Without any nodes, SRVCTL creates a file system resource that runs on every cluster node.
`-fsoptions` *options*	Modify options used to mount the file system. The options (comma-delimited list) are file system and operating system dependent.
`-description` *description*	Modify the description of the file system.
`-autostart {ALWAYS | NEVER | RESTORE}`	Modify the file system resource autostart policy. `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.
`-force`	Force modification. This option ignores dependencies.

Usage Notes

- Use the `-device` parameter to specify the path to the file system resource for which you want to modify, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- When performing online changes to the file system resource, the changes take effect only after the resource is next (re)started.
- You must run this command as `root` user on Linux and UNIX platforms.
- This command is only available with Oracle Clusterware.

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 | -volume volume_name -
diskgroup disk_group_name} [-verbose]
```

Parameters

Table F-34 srvctl predict filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.
`-verbose`	Optionally, you can use the `-verbose` parameter to display detailed output.

Usage Notes

- Specify the `-device` parameter to specify the path to the file system resource for which you want to predict the consequences of file system failure, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- Optionally, you can choose verbose output.
- This command is only available with Oracle Clusterware.

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.

Syntax

```
srvctl remove filesystem {-device volume_device | -volume volume_name -
diskgroup disk_group_name} [-force]
```

Parameters

Table F-35 srvctl remove filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.
`-force`	Optionally, you can use the `-force` parameter to forcibly remove the resource from the cluster.

Usage Notes

- Specify the `-device` parameter to specify the path to the file system resource for which you want to remove, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- 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.
- This command is only available with Oracle Clusterware.

Examples

An example of this command is:

```
# srvctl remove filesystem -device /dev/asm/racvol1
```

srvctl start filesystem

Starts (mounts) the file system resource.

Syntax

```
srvctl start filesystem {-device volume_device | -volume volume_name -
diskgroup disk_group_name} [-node node_name]
```

Parameters

Table F-36 srvctl start filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path of the file system resource device that you want to start.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.
`-node` *node_name*	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

- Use the `-device` parameter to specify the path to the file system resource for which you want to start, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- 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.

  If `srvctl add filesystem` was run with the `-user` parameter to identify users authorized to mount and unmount the file system, then you can run `srvctl start filesystem` as one of the configured file system users.

> **See Also:**
>
> srvctl add filesystem for more information about configuring file system users

- This command is only available with Oracle Clusterware.

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.

Syntax

```
srvctl status filesystem [-device volume_device | -volume volume_name -
diskgroup disk_group_name] [-verbose]
```

Parameters

Table F-37 srvctl status filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.
`-verbose`	Optionally, you can use the `-verbose` parameter to display detailed output.

Usage Notes

- Optionally, you can use the `-device` parameter to specify the path to the file system resource for which you want to obtain status, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*. If you do not specify `-device` or `-volume -diskgroup`, then SRVCTL displays the status of all file systems.
- This command is only available with Oracle Clusterware.

Examples

You can specify a device, or volume and disk group.

For example, if you specify a device name:

```
$ srvctl status filesystem -d /dev/asm/fs1vol-358
ACFS file system /acfs/fs1 is not mounted

$ srvctl start filesystem -d /dev/asm/fs1vol-358
...
$ srvctl status filesystem -d /dev/asm/fs1vol-358
ACFS file system /acfs/fs1 is mounted on nodes
nd1-4n-01,nd1-4n-02,nd1-4n-03,nd1-4n-04
```

For example, if you specify a volume and disk group:

```
$ srvctl status filesystem -volume fs1vol -diskgroup DATA
ACFS file system /acfs/fs1 is mounted on nodes
nd1-4n-01,nd1-4n-02,nd1-4n-03,nd1-4n-04
```

If you do not specify a device, or volume and disk group:

```
$ srvctl status filesystem
ACFS file system /acfs/fs1 is mounted on nodes
nd1-4n-01,nd1-4n-02,nd1-4n-03,nd1-4n-04
```

> **Note:**
>
> In the preceding examples, the file system is Oracle ACFS. If you are using other file systems, then the file systems display as EXT3 or EXT4.

srvctl stop filesystem

Stops (unmounts) the Oracle ACFS file system or generic file system.

Syntax

```
srvctl stop filesystem {-device volume_device | -volume volume_name -
diskgroup disk_group_name} [-node node_name] [-force]
```

Parameters

Table F-38 srvctl stop filesystem Command Parameters

Parameter	Description
`-device` *volume_device*	Specify the path to the file system volume device that you want to stop.
`-volume` *volume_name*	Specify the name of the volume.
`-diskgroup` *disk_group_name*	Specify the name of the disk group on which the `-volume` is located.

Table F-38 (Cont.) srvctl stop filesystem Command Parameters

Parameter	Description
`-node` *node_name*	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.
`-force`	Optionally, you can use `-force` to stop the file system and also stop any databases or other resources that depend on this file system.

Usage Notes

- Use the `-device` parameter to specify the path to the file system resource for which you want to stop, or specify `-volume` *volume_name* `-diskgroup` *disk_group_name*.
- 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.

  If `srvctl add filesystem` was run with the `-user` parameter to identify users authorized to mount and unmount the file system, then you can run `srvctl stop filesystem` as one of the configured file system users.

  > **See Also:**
  >
  > srvctl add filesystem for more information about configuring file system users

- This command is only available with Oracle Clusterware.

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.

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]
```

Parameters

Table F-39 srvctl add gns Command Parameters

Parameter	Description
`-domain` *domain_name*	The network subdomain that is used for Forward DNS Lookup of cluster IP addresses. You can only use this parameter if you specify `-vip`, 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 `-vip` with the `-domain` parameters becomes a server cluster. If you do not specify `-domain`, then SRVCTL adds GNS without a domain.
`-vip {`*vip_name* `|` *ip_address*`}`	Specify either the virtual IP (VIP) name or IP address on which GNS listens for DNS requests. Use the `-domain` parameter with `-vip` 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.
`-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.

Examples

To add GNS to a cluster:

```
# srvctl add gns -vip 192.168.16.17 -domain cluster.mycompany.com
```

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] [-loglevel]
  [-network] [-instances]
```

Parameters

Table F-40 srvctl config gns Command Parameters

Parameter	Description
`-detail`	Print detailed configuration information about the GNS.

Table F-40 (Cont.) srvctl config gns Command Parameters

Parameter	Description
`-subdomain`	Display the subdomain served by GNS.
`-multicastport`	Display the port on which the GNS daemon is listening for multicast requests.
`-node` *`node_name`*	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` *`name`*	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.
`-loglevel`	Print the log level of the GNS.
`-network`	Display network on which GNS is listening.
`-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 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_address} [-skip]] [-verbose]
```

Parameters

Table F-41 srvctl modify gns Command Parameters

Parameter	Description
`-loglevel log_level`	Specify the GNS diagnostic logging level (acceptable values are 1-6).
`-resolve name`	Resolve the name you specify through GNS.

Table F-41 (Cont.) srvctl modify gns Command Parameters

Parameter	Description
`-verify` *name*	Check to see if the specified name is advertised through GNS.
`-parameter` *name*`:`*value*`[,`*name*`:`*value* `...]`	Set the value of one or more configuration parameters.
`-vip {`*vip_name* `|` *ip*`}`	Specify a VIP name or IP address on which GNS is to listen. **Note:** You can modify this attribute using Online Resource Attribute Modification.
`-skip`	Specify this parameter to skip the reachability check of the VIP address.
`-verbose`	Verbose output.

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 F-42 srvctl start gns Command Parameters

Parameter	Description
`-loglevel` *log_level*	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).
`-node` *node_name*	Optionally, you can specify the name of a node in the cluster where you want to start GNS.
`-verbose`	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 `crmnode1` is:

```
$ srvctl start gns -node crmnode1
```

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 F-43 srvctl update gns Command Parameters

Parameter	Description
`-advertise name`	Use this parameter to advertise a name through GNS.
`-address ip_address`	Specify the IP address for an advertised name.
`-timetolive time_to_live`	Optionally, you can specify, in seconds, an amount of time for the record to live.
`-delete name`	Use this parameter to remove the advertisement of a name from GNS.
`-alias alias`	Use this parameter to create an alias for an advertised name.
`-name name`	You must specify a name to associate with the alias.
`-deletealias alias`	Use this parameter to delete an alias.
`-createsrv service`	Use this parameter to create a service that is described by the record.
`-target target`	Use this parameter to associate a name with the service.
`-protocol protocol`	Use this parameter to specify which protocol is used to connect to the service.
`-weight weight`	Optionally, you can specify the weight of the record.
`-priority priority`	Optionally, you can specify a priority of the record, with a value ranging from 0 to 65535.
`-port port_number`	Optionally, you can specify a port number used to connect to the service, with a value ranging from 0 to 65535.
`-instance instance_name`	Optionally, you can specify an instance name for the service.
`-deletesrv service_name`	Use this parameter to specify the name of the service record you want to delete.
`-createtxt name`	Use this parameter to add a text (TXT) record using the specified name.
`-namettl name_ttl`	Optionally, you can specify the time, in seconds, to live for the name.
`-deletetxt name`	Use this parameter to delete a text (TXT) record for a specific name.
`-createptr name`	Use this parameter to add a pointer (PTR) record for a specific name.
`-deleteptr name`	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 F-44 srvctl add havip Command Parameters

Parameter	Description
`-id havip_name`	Specify a unique ID for the HAVIP resource.
`-address {host_name | ip_address}`	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 network_number`	Optionally, you can specify a network resource upon which the HAVIP depends. The default value for this parameter is 1.
`-description text`	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 node_name`	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 F-45 srvctl config havip Command Parameters

Parameter	Description
`-id havip_name`	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 F-46 srvctl disable havip Command Parameters

Parameter	Description
`-id havip_name`	Specify the unique name of the HAVIP resource you want to disable.
`-node node1,node2,...,noden`	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 F-47 srvctl enable havip Command Parameters

Parameter	Description
`-id havip_name`	Specify the unique name of the HAVIP resource you want to enable.
`-node node_name`	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 F-48 srvctl modify havip Command Parameters

Parameter	Description
`-id havip_name`	Specify the unique name for the HAVIP resource you want to modify.
`-address {host_name | ip_address`	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.

Table F-48 (Cont.) srvctl modify havip Command Parameters

Parameter	Description
`-netnum` *network_number*	Optionally, you can change the network resource upon which the HAVIP depends. The default value for this optional parameter is 1.
`-skip`	Specify this parameter to skip checking the reachability of the IP address.
`-description` *text*	Specify a text description for the HAVIP.
`-homenode` *node_name*	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 F-49 srvctl relocate havip Command Parameters

Parameter	Description
`-id` *havip_name*	Specify the unique name of the HAVIP resource you want to relocate.
`-node` *node_name*	Specify the name of the node to which you want to relocate the HAVIP resource.
`-force`	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 F-50 srvctl remove havip Command Parameters

Parameter	Description
`-id havip_name`	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 F-51 srvctl start havip Command Parameters

Parameter	Description
`-id havip_name`	Specify the unique name of the HAVIP resource you want to start.
`-node node_name`	Optionally, you can use the `-node` 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.ha1.havip is enabled
HAVIP ora.ha1.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 F-52 srvctl stop havip Command Parameters

Parameter	Description
`-id havip_name`	Specify the unique name of the HAVIP resource to stop.
`-node node_name`	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 IOServer configuration to the cluster.

When you configure or upgrade to an Oracle Flex Cluster, you can create an Oracle ASM IOServers on the nodes in the cluster.

Syntax

```
srvctl add ioserver [-spfile spfile] [-count number_of_ioserver_instances|
ALL] [-listener listener_name]
```

Parameters

Table F-53 srvctl add ioserver Command Parameters

Parameter	Description
`-spfile spfile`	Specify the path to the server parameter file.
`-count number_of_ioserver_instances| ALL`	Specify the number of Oracle ASM IOServer instances you want to add or `ALL` to add instances to all nodes.
`-listener listener_name`	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 IOServers.

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 IOServer.

Syntax

```
srvctl disable ioserver -node node_name
```

Usage Notes

Use the `-node` parameter to specify the name of a cluster node on which you want to disable an Oracle ASM IOServer.

srvctl enable ioserver

Enable an Oracle ASM IOServer on a specific node.

Syntax

```
srvctl enable ioserver [-node node_name]
```

Usage Notes

Use the `-node` parameter to specify the name of a cluster node on which you want to enable an Oracle ASM IOServer.

srvctl getenv ioserver

Displays the values for environment variables associated with an Oracle ASM IOServer.

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 IOServer configuration on a cluster.

Syntax

```
srvctl modify ioserver [-spfile spfile] [-count number_of_ioserver_instances
| ALL]
  [-listener listener_name] [-force]
```

Parameters

Table F-54 srvctl modify ioserver Command Parameters

Parameter	Description
`-spfile spfile`	Optionally, specify the path to the server parameter file path.
`-count number_of_ioserver_instances | ALL`	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.
`-listener listener_name`	Optionally, specify the name of the listener.
`-force`	Optionally, you can ignore any dependencies and force the modification of the Oracle ASM IOServer.

srvctl relocate ioserver

Relocates an Oracle ASM IOServer 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 F-55 srvctl relocate ioserver Command Parameters

Parameter	Description
`-currentnode node_name`	The name of the node on which the Oracle ASM IOServer instance currently resides
`-targetnode node_name`	The name of the node to which to relocate the IOServer instance
`-force`	Optionally, you can force the relocation of the Oracle ASM IOServer instance

Example

The following command relocates the Oracle ASM IOServer 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 IOServer instance from the cluster.

Syntax

```
srvctl remove ioserver [-force]
```

Usage Notes

Use the `-force` parameter to forcibly remove the Oracle ASM IOServer instance, ignoring any dependencies.

srvctl setenv ioserver

Administers the environment configuration for an Oracle ASM IOServer.

Syntax

```
srvctl setenv ioserver {-envs "name=val[,name=val][...]" | -env "name=val"}
```

Parameters

Table F-56 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 IOServers in the cluster.

You can also start an Oracle ASM IOServer on a particular cluster node by specifying the node name.

Syntax

```
srvctl start ioserver [-node node_name]
```

Usage Notes

If you choose to start an Oracle ASM IOServer on a particular cluster node, then use the `-node` parameter and specify the name of the node. If you do not specify a particular cluster node, then SRVCTL starts all Oracle ASM IOServers in the cluster.

Example

The following command starts an Oracle ASM IOServer on a cluster node named `mjk46`:

```
$ srvctl start ioserver -node mjk46
```

srvctl status ioserver

Displays the status of Oracle ASM IOServers running in the cluster.

You can display the status of all Oracle ASM IOServers running a cluster, the status of an Oracle ASM IOServer running on a particular cluster node, or the information about clients from a specific database.

Syntax

```
srvctl status ioserver [-node node_name] [-detail]
```

Parameters

Table F-57 srvctl status ioserver Parameters

Parameter	Description
—node *node_name*	Specify a particular cluster node to display the status of the Oracle ASM IOServer resource instance running there. If you do not specify a cluster node, then SRVCTL returns the status of all Oracle ASM IOServer resource instances running in the cluster.
-detail	Use this parameter to display more detailed information about the status of the Oracle ASM IOServer.

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 IOServer in the cluster.

Syntax

```
srvctl stop ioserver [-node node_name] [-force]
```

Usage Notes

- Use the `-node` parameter to stop an Oracle ASM IOServer on a particular cluster node. If you do not specify a particular node, then SRVCTL stops all of the Oracle ASM IOServers 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 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 F-58 srvctl add mountfs Command Parameters

Parameter	Description
`-name mountfs_name`	Specify a unique name for the NAS.
`-path mount_path`	Specify a mount path for the NAS.
`-exportserver server_name`	Specify a name for the export server.
`-exportpath path`	Specify an export file path.
`-mountoptions mount_options`	Optionally, you can specify mount options for the NAS.
`-user user`	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 F-59 srvctl disable mountfs Command Parameters

Parameter	Description
`-name mountfs_name`	Specify the unique name for the NAS you want to disable.
`-node node_list`	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 F-60 srvctl enable mountfs Command Parameters

Parameter	Description
`-name mountfs_name`	Specify the unique name for the NAS you want to enable.
`-node node_list`	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 F-61 srvctl modify mountfs Command Parameters

Parameter	Description
`-name mountfs_name`	Specify a unique name for the NAS you want to modify.
`-path mount_path`	Modify the mount path for the NAS.
`-exportserver server_name`	Modify the name of the export server.
`-exportpath path`	Modify the export file path.
`-mountoptions mount_options`	Modify mount options for the NAS.
`-user user_name`	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 F-62 srvctl start mountfs Command Parameters

Parameter	Description
`-name mountfs_name`	Specify the unique name for the NAS to mount.
`-node node_list`	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 F-63 srvctl stop mountfs Command Parameters

Parameter	Description
`-name mountfs_name`	Specify the unique name for the NAS to unmount.
`-node node_list`	Optionally, you can specify a comma-delimited list of nodes from which to unmount NAS.

Table F-63 (Cont.) srvctl stop mountfs Command Parameters

Parameter	Description
`-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 F-64 srvctl add netstorageservice Command Parameters

Parameter	Description
`-device` *`volume_device`*	Specify the device and volume name to which you want to add the `netstorageservice` 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 F-65 srvctl disable netstorageservice Command Parameters

Parameter	Description
`-node node_name`	Optionally, you can specify the name of the node on which the netstorageservice resource should be disabled. If you do not specify this parameter, then SRVCTL disables the netstorageservice resource on all the available nodes in the cluster.

Examples

To disable the `netstorageservice` resource on a specified node:

```
$ srvctl disable netstorageservice -node node1
```

srvctl enable netstorageservice

Enables the netstorageservice resource.

Syntax

```
srvctl enable netstorageservice [-node node_name]
```

Parameters

Table F-66 srvctl enable netstorageservice Command Parameters

Parameter	Description
`-node node_name`	Optionally, you can specify the name of the node on which the netstorageservice resource should be enabled. If you do not specify this parameter, then SRVCTL enables the netstorageservice resource on all the available nodes in the cluster.

Examples

To enable the `netstorageservice` resource on a specified node:

```
$ srvctl enable netstorageservice -node node1
```

srvctl remove netstorageservice

Removes the netstorageservice resource.

Syntax

```
srvctl remove netstorageservice [-force]
```

Parameters

Table F-67 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 F-68 srvctl start netstorageservice Command Parameters

Parameter	Description
`-node` *`node_name`*	Optionally, you can specify the name of the node on which the netstorageservice resource should be started. If you do not specify this parameter, then SRVCTL starts the netstorageservice 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 F-69 srvctl stop netstorageservice Command Parameters

Parameter	Description
`-node node_name`	Optionally, you can specify the name of the node on which the netstorageservice resource should be stopped. If you do not specify this parameter, then SRVCTL stops the netstorageservice resource on all the available nodes in the cluster.
`-force`	Force a stop.

Examples

To stop the `netstorageservice` resource on a specified node:

```
$ srvctl stop netstorageservice -node node1
```

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]
  [-enableTLS {YES|NO} [-p12certpath P12_file_absolute_path -certname
custom_cert_name]]
  [-email email_address -mailserver mail_server_address -mailserverport
mail_server_port] [-verbose]
```

Parameters

Table F-70 srvctl add rhpclient Command Parameters

Parameter	Description
`-clientdata` *path_to_file*	Specify the path to the file that contains the Fleet Patching and Provisioning Client data.
`-diskgroup` *disk_group_name*	Specify the Oracle ASM disk group from which to create the Oracle ACFS file system for image storage.
`-storage` *base_path*	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.
`-enableTLS {YES|NO}`	Specify this parameter to enable or disable TLS. **Note:** You must enable TLS to use custom security certificates.
`-p12certpath` *P12_file_absolute_path*	Specify absolute path to the P12 certificate file.
`-certname` *custom_cert_name*	Specify a name with which you want to register the certificate.
`-email` *email_address*	Specify an email address.
`-mailserver` *mail_server_address*	Specify the mail server address.
`-mailserverport` *mail_server_port*	Specify the mail server port number.
`-verbose`	Displays verbose output.
`-tlscert` *certificate_wallet*	Specify path of the wallet that has security certificates stored in it.

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 F-83). 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]
  [-p12certpath P12_file_absolute_path -certname custom_cert_name]
  [-email email_address -mailserver mail_server_address -mailserverport
mail_server_port]
```

Parameters

Table F-71 srvctl modify rhpclient Command Parameters

Parameter	Description
`-clientdata` *path_to_client_data*	Modify the path to the Fleet Patching and Provisioning Client data.
`-port` *rmi_port*	Modify the RMI port number used by the Fleet Patching and Provisioning Client.
`-diskgroup` *dg_name*	Modify the Oracle ASM disk group from which to create the Oracle ACFS file systems for image storage.
`-storage` *base_path*	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 *base_path*`/images` for all local Oracle ACFS storage.
`-p12certpath` *P12_file_absolute_path*	Specify absolute path to the P12 certificate file.
`-certname` *custom_cert_name*	Specify a name with which you want to register the certificate.
`-email` *email_address*	Modify an email address.
`-mailserver` *mail_server_address*	Modify the mail server address.
`-mailserverport` *mail_server_port*	Modify the mail server port number.
`-tlscert` *certificate_wallet*	Specify path of the wallet that has security certificates stored in it.

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 | -vault
vault_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} [-p12certpath p12_file_path -certname cert_name]] [-
enableHTTPS '{YES|NO}']
  [-dbtype {FPPDB|EXTERNAL} [-connstr connection_string [-dbadmin db_user]]]
[-port_range low_val-high_val]
  [-tmploc temporary_location] [-rhpsvip_address rhps_vip_address]
  [-memsettings {LOW|MEDIUM|HIGH}] [-verbose] [-ssh_port ssh_port_number]
```

> **Note:**
>
> If `-dbtype` is set to `EXTERNAL`, then `srvctl add rhpserver` prompts for the `sys` user password for Oracle FPP repository database.

Parameters

Table F-72 srvctl add rhpserver Command Parameters

Parameter	Description
`-storage base_path`	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 `base_path/images`.
`-diskgroup disk_group_name`	Specify a particular Oracle ASM disk group from which to create the Oracle ACFS file system for storing images.
`-vault vault_name`	Specify an Oracle Database vault name.
`-enableTLS {YES|NO}`	Specify this parameter to enable or disable TLS. **Note:** You must enable TLS to use custom security certificates.
`-p12certpath P12_file_absolute_path`	Specify absolute path to the P12 certificate file.
`-certname custom_cert_name`	Specify a name with which you want to register the certificate.
`-email email_address`	Specify an email address.

Table F-72 (Cont.) srvctl add rhpserver Command Parameters

Parameter	Description
-mailserver *mail_server_address*	Specify the mail server address.
-mailserverport *mail_server_port*	Specify the mail server port number.
-pl_port *RHP_progress_listener_port*	Specify the Fleet Patching and Provisioning progress listener port number.
-clport *RHP_copy_listener_port*	Specify the Fleet Patching and Provisioning copy listener port number.
-enableHTTPS '{YES|NO}'	Enable Fleet Patching and Provisioning transport layer security for HTTP.
-dbtype *{FPPDB|EXTERNAL}*	Specify the metadata repository type. The EXTERNAL option creates an Oracle Database repository.
-connstr *connection_string*	Specify the connection string for the database. This parameter is mandatory if you create an external metadata repository.
-dbadmin *db_user*	Specify a database user with DBA privileges on the external metadata repository. By default, SYS user is used. This parameter is not required if wallet is specified.
-port_range *low_val-high_val*	Specify the lower and upper port values for the ports to use for file transfer.
-tmploc*temporary_location*	Modify the temporary location for Oracle FPP Server installation processing.
-rhpsvip_address *rhps_vip_address*	Specify the VIP address for the Oracle FPP Server.
-memsettings {LOW|MEDIUM|HIGH}	Modify the Java heap memory settings. The following settings are available: • LOW: Minimum heap memory is set to 128MB and maximum heap memory is set to 384MB. • MEDIUM: Minimum heap memory is set to 3GB and maximum heap memory is set to 5GB. • HIGH: Minimum heap memory is set to 6GB and maximum heap memory is set to 8GB.
-ssh_port *ssh_port_number*	Modify the SSH port that Oracle FPP use for command execution.

Usage Notes

- You must be root user on Linux and UNIX platforms to run this command.
- The default user is the user who installed Oracle Clusterware, and this user gets added to the user list for the Fleet Patching and Provisioning Server resource.

Example

An example of this command is:

```
$ srvctl add rhpserver -storage /scratch/rhp_storage -diskgroup GHDGP_01 -
dbtype FPPDB -dbadmin cdbadmin
Enter user "sys" password:
```

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/rhp_storage
Disk Groups: GHDGP_01
Port number: 8896
Temporary Location:
Transfer port range:
Rapid Home Provisioning Server is enabled
Rapid Home Provisioning Server is individually enabled on nodes:
Rapid Home Provisioning Server is individually disabled on nodes:
Email address:
Mail server address:
Mail server port:
Transport Level Security enabled
HTTP Secure is enabled
Endpoint: rhps-vip:vip_address
Minimum Heap memory: 128, Maximum Heap memory: 384
SSH Port: 22
Connection Description:
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=host_name)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=pdb1)))
Fri Jul 14 07:30:39 UTC 2023
```

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 rhpserver -node clusterNode01
```

srvctl enable rhpserver

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 rhpserver [-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 rhpserver -node clusterNode03
```

srvctl modify rhpserver

Modifies the storage for a Fleet Patching and Provisioning Server.

Syntax

```
srvctl modify rhpserver [-port rmi_port [-force]] [-p12certpath
P12_file_absolute_path -certname custom_cert_name]
   [-email email_address -mailserver mail_server_address -mailserverport
mail_server_port]
   [-pl_port RHP_progress_listener_port] [-clport RHP_copy_listener_port] [-
resetcreds [-dbadmin db_admin_user_name] [-newpwd]]
   [-enableHTTPS '{YES|NO}'] [-port_range low_val-high_val] [-tmploc
temporary_location] [-rhpsvip_address rhps_vip_address]
   [-httpport rhpserver_http_port] [-memsettings {LOW|MEDIUM|HIGH}] [-
ssh_port ssh_port_number]
```

Parameters

Table F-73 srvctl modify rhpserver Command Parameters

Parameter	Description
`-port rmi_port`	Modify the RMI port number used by the Fleet Patching and Provisioning Server.
`-force`	Specify this parameter to stop and restart the resource to effect a change.
`-p12certpath P12_file_absolute_path`	Specify absolute path to the P12 certificate file.
`-certname custom_cert_name`	Specify a name with which you want to register the certificate.
`-email email_address`	Modify an email address.
`-mailserver mail_server_address`	Modify the mail server address.
`-mailserverport mail_server_port`	Modify the mail server port number.
`-pl_port RHP_progress_listener_port`	Modify the Fleet Patching and Provisioning progress listener port number.
`-clport RHP_copy_listener_port`	Modify the Fleet Patching and Provisioning copy listener port number.
`-resetcreds`	Resets Oracle FPP schema credentials in the metadata repository.
`-dbadmin`	Metadata repository database administrative user account name.
`-newpwd`	Specify a new password through standard input instead of automatically generating a password.
`-enableHTTPS '{YES|NO}'`	Enable Fleet Patching and Provisioning transport layer security for HTTP.
`-port_range low_val-high_val`	Modify the lower and upper port values for the ports to use for file transfer.

Table F-73 (Cont.) srvctl modify rhpserver Command Parameters

Parameter	Description
-tmploc*temporary_location*	Modify the temporary location for Oracle FPP Server installation processing.
-rhpsvip_address *rhps_vip_address*	Specify the VIP address for the Oracle FPP Server.
-httpport *rhpserver_http_port*	Specify the HTTP port number.
-memsettings {LOW|MEDIUM|HIGH}	Modify the Java heap memory settings. The following settings are available: • LOW: Minimum heap memory is set to 128MB and maximum heap memory is set to 384MB. • MEDIUM: Minimum heap memory is set to 3GB and maximum heap memory is set to 5GB. • HIGH: Minimum heap memory is set to 6GB and maximum heap memory is set to 8GB.
-ssh_port *ssh_port_number*	Modify the SSH port that Oracle FPP use for command execution.

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]
```

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, the `srvctl remove rhpserver` command prompts for the `sys` user password for the Oracle FPP repository database.

Parameters

Table F-74 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.

Example

An example of this command is:

```
$ srvctl remove rhpserver
Enter user "sys" password:
```

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 resource for Kernel Virtual Machines (VMs) to the Oracle Clusterware.

Syntax

```
srvctl add vm -name unique_resource_name [-node "node_list"] [-failover {YES|
NO}]
    [-starttimeout start_timeout] [-stoptimeout stop_timeout] [-checkinterval
check_interval]
    [-dbvm {YES|NO}] [-xmlpath domain_path] [-acfspath "acfs_path_list"] [-
autostart {ALWAYS|NEVER|RESTORE}]
    [-placement {balanced|restricted|favored}] [-description
"KVM_resource_description"]
```

Parameters

Table F-75 srvctl add vm Command Parameters

Parameter	Description
`-name resource_name`	Specify a unique name for the virtual machine resource. If the name is not unique, then the command results in an error.
`-node "node_list"`	Optionally, you can specify the 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.
`-failover {YES|NO}`	Specify whether or not to allow failover of KVM resources from the nodes on which they fail.
`-starttimeout start_timeout`	Specify a start timeout, in seconds.
`-stoptimeout stop_timeout`	Specify a stop timeout, in seconds. The default is 180 if you do not assign a value to the parameter.
`-checkinterval check_interval`	Specify an interval between checks, in seconds. The default is 10 if you do not assign a value to the parameter.
`-dbvm {YES|NO}`	Specify this flag to indicate that this VM resource belongs to a database server.
`-xmlpath xml_domain_path`	Specify the absolute path to the VM resource XML domain file.
`-acfspath "acfs_path_list"`	Specify the list of Oracle ACFS paths for VM resource usage. **Note:** This parameter is available only on the platforms on which Oracle ACFS is supported.

Table F-75 (Cont.) srvctl add vm Command Parameters

Parameter	Description
`-autostart {ALWAYS|NEVER|RESTORE}`	Autostart policy for the VM resource.
`-placement {balanced|restricted|favored}`	Specify how Oracle Clusterware selects a node on which to start the VM resource.
`-description` *`description`*	Provide a valid description for the VM resource.

Usage Notes

You must add virtual machine manager credentials before you add a virtual machine resource.

Example

An example of this command is:

```
srvctl add vm -name vmres -node node1 -failover YES -autostart ALWAYS -
placement balanced
```

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 [-node node_name]
```

Parameters

Table F-76 srvctl check vm Command Parameters

Parameter	Description
`-name` *`resource_name`*	Specify the name of the virtual machine resource for which you want to check the status.
`-node` *`node_name`*	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
Nodes: node1, node2
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 disabed 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 [-node node_name]
```

Parameters

Table F-77 srvctl disable vm Command Parameters

Parameter	Description
`-name unique_name`	Specify the unique name of the virtual machine resource you want to disable.
`-node node_name`	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
```

srvctl enable vm

Enables a virtual machine resource.

Syntax

```
srvctl enable vm -name unique_name [-node node_name]
```

Parameters

Table F-78 srvctl enable vm Command Parameters

Parameter	Description
`-name unique_name`	Specify the name of the virtual machine resource you want to enable.
`-node node_name`	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_resource_name [-node "node_list"] [-failover
{YES|NO}]
    [-starttimeout start_timeout] [-stoptimeout stop_timeout] [-checkinterval
check_interval]
```

```
    [-dbvm {YES|NO}] [-xmlpath domain_path] [-acfspath "acfs_path_list"] [-
autostart {ALWAYS|NEVER|RESTORE}]
    [-placement {balanced|restricted|favored}] [-description
"KVM_resource_description"]
```

Parameters

Table F-79 srvctl modify vm Command Parameters

Parameter	Description
`-name resource_name`	Specify a unique name for the virtual machine resource. If the name is not unique, then the command results in an error.
`-node "node_list"`	Optionally, you can specify the list of nodes to which to modify the virtual machine resource configuration. 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.
`-failover {YES|NO}`	Specify whether or not to allow failover of KVM resources from the nodes on which they fail.
`-starttimeout start_timeout`	Modify the start timeout, in seconds.
`-stoptimeout stop_timeout`	Modify the stop timeout, in seconds. The default is 180 if you do not assign a value to the parameter.
`-checkinterval check_interval`	Modify the interval between checks, in seconds. The default is 10 if you do not assign a value to the parameter.
`-dbvm {YES|NO}`	Specify this flag to indicate that this VM resource belongs to a database server.
`-xmlpath xml_domain_path`	Modify the absolute path to the VM resource XML domain file.
`-acfspath "acfs_path_list"`	Modify the list of Oracle ACFS paths for VM resource usage. **Note:** This parameter is available only on the platforms on which Oracle ACFS is supported.
`-autostart {ALWAYS|NEVER|RESTORE}`	Autostart policy for the VM resource.
`-placement {balanced|restricted|favored}`	Specify how Oracle Clusterware selects a node on which to start the VM resource.
`-description description`	Provide a valid description for the VM resource.

Example

An example of this command is:

```
$ srvctl modify vm -name vmres -node "node1,node2" -stoptimeout 93 -autostart
ALWAYS -placement balanced
```

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 {-srcnode source_node_name}
   -node destination_node_name]
```

Parameters

Table F-80 srvctl relocate vm Command Parameters

Parameter	Description
`-name unique_name`	Specify the unique name of the virtual machine resource you want to relocate.
`-srcnode source_node_name`	Alternatively, you can specify a node from which you want to relocate a virtual machine resource.
`-node destination_node_name`	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 -srcnode node1 -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 [-node node_name]
```

Parameters

Table F-81 srvctl start vm Command Parameters

Parameter	Description
`-name resource_name`	Specify the unique name of the virtual machine resource you want to start.
`-node node_name`	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 `-node` parameter to start a particular virtual machine resource on a particular node.

Example

An example of this command is:

```
$ srvctl start vm -name vmres -node node3
```

srvctl status vm

Displays the currently known state of the virtual machine resource.

Syntax

```
srvctl status vm -name unique_name [-node node_name]
```

Parameters

Table F-82 srvctl status vm Command Parameters

Parameter	Description
`-name unique_name`	Specify the unique name of the virtual machine resource for which you want to display the status.
`-node node_name`	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

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 stop vm

Stops a specific virtual machine resource on the specified nodes.

Syntax

```
srvctl stop vm -name unique_name [-node node_name] [-force]
```

Parameters

Table F-83 srvctl stop vm Command Parameters

Parameter	Description
`-name unique_name`	Specify the unique name of the virtual machine resource you want to stop.
`-node node_name`	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.
`-force`	Stop the VM resource forcefully.

Example

An example of this command is:

```
$ srvctl stop vm -name vmres1 -node node1
```

G

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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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.

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 run agent programs on multiple platforms.

Table G-1 lists and describes the Oracle data types used in the agent framework.

Table G-1 Oracle Data Types Used in the Agent Framework

Oracle Data Type	Description
ub4	Unsigned 32-bit integer
sb4	Signed 32-bit integer

Table G-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 processed 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 run 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 G-2 lists and describes entry point types and their equivalent type codes.

Table G-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 running 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 run application-specific functionality, such as monitoring external event sources.

The agent framework processes 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 mutexes. Other external C or C++ libraries included by the agent should also be multithread safe.

Deprecated APIs

Table G-3 lists the deprecated agent framework APIs and the corresponding replacement APIs for Oracle Clusterware.

Table G-3 Deprecated Agent Framework APIs

Deprecated API	Replacement
`clsagfw_exit()`	`clsagfw_exit2()`
`clsagfw_send_status()`	`clsagfw_send_status2()`

API Reference

Lists and describes the API functions for Oracle Clusterware.

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 runs 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 running 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 processed.

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 CRSCTL.
- **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 terminated.

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 performed 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 processed.

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 run.

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 entrypoints

Usage Notes

Agents can call this function in entrypoints to check if the current command is being retried. An agent process or Oracle Clusterware server process can terminate in the middle of running 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 run 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 run.

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 terminates 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 *fmtp, ...);
```

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

`fmtp`: 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 run 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 G-1 details an agent implementation to manage a resource that represents a file.

Example G-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_exitcb(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;
}
```

H

Oracle Clusterware C Application Program Interfaces

This appendix describes the Oracle Clusterware C application program interfaces (APIs).

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

> **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

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, and servers. 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

Describes programming interface (C API) to Oracle Clusterware configuration and operational requirements.

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 runs 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 H-1 describes an example of the callback mechanism.

Example H-1 Callback Mechanism

```
void myCallback(void *arg, const oratext *pId, const oratext *pMsg,
                clscrs_msgtype msgType)
{
    if (pMsg != NULL)
   {
       cout << pMsg << endl;
    }
}
```

Example H-2 describes how to use the callback mechanism in an interactive API.

Example H-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 Oracle 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 Oracle 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 H-3 describes using filters in an interactive CLSCRS API.

Example H-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 run 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 Oracle Clusterware), along with a demo called `crsapp.c`.

Deprecated CLSCRS APIs

Table H-1 lists the deprecated CLSCRS APIs and the corresponding replacement APIs for Oracle Clusterware.

Table H-1 Deprecated CLSCRS APIs

Deprecated API	Replacement
`clscrs_check_resource`	`clscrs_check_resource2`
`clscrs_entity`	`clscrs_entity_type`
`clscrs_fail_resource`	No replacement
`clscrs_msgf`	`clscrs_msgf2`
`clscrs_register_resource`	`clscrs_register_resource2`
`clscrs_relocate_resource`	`clscrs_relocate_resource2`
`clscrs_res_attr_count`	`clscrs_crsentity_attr_count`
`clscrs_res_create`	`clscrs_crsentity_create`
`clscrs_res_create`	`clscrs_crsentitylist_create`
`clscrs_res_destroy`	`clscrs_crsentity_destroy`
`clscrs_res_get_attr`	`clscrs_crsentity_get_attr`
`clscrs_res_get_attr_list`	`clscrs_crsentity_get_attr_list`
`clscrs_res_get_name`	`clscrs_crsentity_get_name`
`clscrs_res_get_node_list`	`clscrs_crsentity_get_node_list`

Table H-1 (Cont.) Deprecated CLSCRS APIs

Deprecated API	Replacement
clscrs_res_get_op_status	clscrs_crsentity_get_op_status
clscrs_res_get_reslist	clscrs_crsentity_get_crsentitylist
clscrs_res_set_attr	clscrs_crsentity_set_attr
clscrs_res_set_attr_list	clscrs_crsentity_set_attr_list
clscrs_res_set_reslist	clscrs_crsentity_set_crsentitylist
clscrs_reslist_append	clscrs_crsentitylist_append
clscrs_reslist_count	clscrs_crsentitylist_count
clscrs_reslist_delete_res	clscrs_crsentitylist_delete_crsentity
clscrs_reslist_destroy	clscrs_crsentitylist_destroy
clscrs_reslist_find	clscrs_crsentitylist_find
clscrs_reslist_first	clscrs_crsentitylist_first
clscrs_reslist_next	clscrs_crsentitylist_next
clscrs_start_resource	clscrs_start_resource2
clscrs_stat	clscrs_stat2
clscrs_stop_resource	clscrs_stop_resource2
clscrs_unregister_resource	clscrs_unregister_resource2

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 H-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_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_unregister_resource2`	Unregisters the resources in the input list of resource names.

Table H-2 (Cont.) Summary of Interactive CLSCRS APIs for Oracle Clusterware

C API	Description
clscrs_unregister_servercategory	Unregisters server categories in the input server category list.
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.
clscrs_whatif_relocate_resource	Simulates what happens if you relocate a resource.
clscrs_whatif_relocate_server	Simulates what happens if you relocate a server.
clscrs_whatif_set_activepolicy	Simulates what happens if you activate a policy.
clscrs_whatif_start_resource	Simulates what happens if you start a resource.
clscrs_whatif_stop_resource	Simulates what happens if you stop a resource.

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 H-3 Non-Interactive CLSCRS APIs

C API	Description
clscrs_action_getentity	Returns the entity for the action.
clscrs_action_getparams	Returns the list of parameters for the action.
clscrs_action_gettype	Returns the type for the action.
clscrs_actionlist_count	Counts the number of actions in the action list.
clscrs_actionlist_create	Creates an action list.
clscrs_actionlist_destroy	Destroys the action list.

Table H-3 (Cont.) Non-Interactive CLSCRS APIs

C API	Description
clscrs_actionlist_first	Returns the first action in an action list.
clscrs_actionlist_next	Returns the next action in an action list.
clscrs_actionlist_print	Prints the action list.
clscrs_actionlist_seqid	Returns the sequence ID for the action lists
clscrs_compfilter_create	Constructs a simple filter that compares two values.
clscrs_crsentity_attr_count	Counts the number of attributes for an entity.
clscrs_crsentity_create	Creates a new entity (allocates memory).
clscrs_crsentity_destroy	Destroys an entity and frees up memory.
clscrs_crsentity_get_attr	Obtains the value of an entity or server attribute.
clscrs_crsentity_get_attr_list	Obtains the attribute list for an entity, resource type, or server.
clscrs_crsentity_get_crsentitylist	Obtains the entity list for an entity.
clscrs_crsentity_get_name	Obtains the name of an entity.
clscrs_crsentity_get_node_list	Obtains a list of nodes currently hosting the entity.
clscrs_crsentity_get_op_status	Obtains the status of an operation for an entity.
clscrs_crsentity_get_registered	Obtains the registration status of an entity.
clscrs_crsentity_set_attr	Sets an attribute for an entity.
clscrs_crsentity_set_attr_list	Sets the attribute list for an entity, resource type, or server.
clscrs_crsentity_set_crsentitylist	Sets the resource list for an entity.
clscrs_crsentitylist_append	Adds an entity to an entity list.
clscrs_crsentitylist_count	Counts the number of entities in an entity list.
clscrs_crsentitylist_create	Creates a list of entities.
clscrs_crsentitylist_delete_crsentity	Deletes an entity matching a given name from an entity list.
clscrs_crsentitylist_destroy	Destroys an entity list and frees up memory.
clscrs_crsentitylist_find	Finds an entity in an entity list matching a given name.
clscrs_crsentitylist_first	Obtains the first entity on an entity list.
clscrs_crslist_next	Obtains the current next entity from an entity list.
clscrs_entity_id_create	Creates an entity identifier that identifies an Oracle Clusterware entity such as a resource, resource type, server group, and so on.
clscrs_entity_id_destroy	Frees the memory associated with an entity identifier created from clscrs_entity_id_create().
clscrs_exprfilter_create	Constructs an expression filter from comparison or expression filters, or both.
clscrs_filter_destroy	Frees the memory for a filter.
clscrs_get_entity_type	Obtains the entity type corresponding to the entity identifier provided.

Table H-3 (Cont.) Non-Interactive CLSCRS APIs

C API	Description
clscrs_get_fixed_attrlist	Obtains the list of attributes that correspond to an attribute group identifier.
clscrs_get_resource_instance_details	Obtains the resource instance details, such as resource name, cardinality, and degree, from the resource instance identifier that is used.
clscrs_getnodename	Obtains the node name.
clscrs_init_crs	Initializes a context for communications with Oracle Clusterware.
clscrs_sp_get	Obtains the name and value components of a stringpair.
clscrs_sp_get_value	Obtains the value component of a stringpair.
clscrs_sp_set	Changes the value part of a stringpair.
clscrs_splist_append	Adds a new stringpair (sp) to a stringpair list (splist).
clscrs_splist_count	Counts the number of stringpairs (sp) in a stringpair list (splist).
clscrs_splist_create	Creates a new stringpair list.
clscrs_splist_create_and_set	Creates a new stringpair list (splist) and set the name and value for the first stringpair in the list.
clscrs_splist_delete_sp	Deletes a stringpair (sp) from a stringpair list (splist).
clscrs_splist_destroy	Frees the memory for a stringpair list (splist).
clscrs_splist_find	Finds the value for a stringpair (sp) in a stringpair list (splist).
clscrs_splist_first	Obtains the first stringpair (sp) from a stringpair list (splist).
clscrs_splist_next	Obtains the current next stringpair (sp) from a stringpair list (splist). Current next stringpair is effectively the next stringpair in the stringpair list. The list iterator is stored within the API and is not exposed.
clscrs_splist_replace	Replaces the value for a stringpair (sp) in a stringpair list (splist).
clscrs_term_crs	Releases a context for communications with Oracle Clusterware.
clscrs_type_create	Creates a new resource type.
clscrs_type_get_attr	Obtains the value/properties of a resource type attribute.
clscrs_type_set_attr	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. 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
- **Servers**: Add, remove, and relocate
- **Policy**: Change active policy
- **Server category**: Modify

Reasoned command evaluation response plans are available for the following event categories:

- **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 run, 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 H-4 describes how you can use command evaluation APIs.

Example H-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(a
list,&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_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_crsentitylist *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 Oracle Clusterware.

`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_crsentitylist *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 Oracle Clusterware.
`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 Oracle Clusterware.
`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_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 Oracle Clusterware.

`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 Oracle Clusterware.

`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_crsentitylist *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.

Ouput

`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 Oracle Clusterware.

`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.

clscrs_servercategory_create

Creates a server category.

Syntax

```
clscrs_servercategory_create(clscrs_ctx *ctx, const oratext *sc_name,
                             clscrs_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 Oracle Clusterware.

`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 Oracle Clusterware.

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_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_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 Oracle Clusterware.
`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 manages several entities, such as resources, and so on.

However, the `clscrs_stat2` interface only allows retrieving (reading) entities by type, which means that a single retrieval can only return entities of a single type. Therefore, clients that need to get different types of entities and need to have a consistent view of the data structures to make further decisions have 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 and later releases provide 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 run 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_crsentity_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_crsentity_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 Oracle Clusterware.

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 runs 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.

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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

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.

OCRCONFIG Command Reference

Lists and describes OCRCONFIG commands.

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 mounted

    The name of the disk group must be preceded by a plus sign (+).

> **See Also:**
>
> - *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux* 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_base`*`/crsdata/`*`host_name`*`/olr/`, 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 -delbackup

Removes an OCR or OLR physical backup device or backup file.

Syntax

```
ocrconfig -delbackup {+diskgroup_name | file_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 backup location:

```
# ocrconfig -delbackup +bkpdg
```

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 Linux* 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
  > – "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 Linux* 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 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 /oradbocfs/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 2020/05/08 15:06:57 /gridbase/crsdata/dglnx6/olr/
backup_20200508_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 [/gridbase/crsdata/dglnx6/olr]
```

Troubleshooting Oracle Cluster Registry and Diagnostic Output

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.

Troubleshooting Oracle Cluster Registry

Table I-1 describes common OCR problems with corresponding resolution suggestions.

Table I-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 I-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 `ocrconfig` command with the `-replace` option.
OCR has a misconfiguration.	Run the `ocrconfig` command with the `-repair` option as described.
You are experiencing a severe performance effect from OCR processing or you want to remove OCR for other reasons.	Run the `ocrconfig` command with the `-replace` option as described.
OCR has failed and before you can fix it, the node must be rebooted with only one OCR.	Run the `ocrconfig` with the `-repair` 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 I-1 OCRCHECK Utility Sample Output

```
# ocrcheck

Status of Oracle Cluster Registry is as follows :
        Version                  :          4
        Total space (kbytes)     :     491684
        Used space (kbytes)      :      99128
        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 I-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 I-2 Using OCRCHECK to Obtain OLR Information

```
$ ocrcheck -local -config

Oracle Local Registry configuration is :
        Device/File Name         : /oraclebase/crsdata/dglnx6/olr/
dglnx6_21.olr
```

Example I-3 shows using the `ocrcheck -config -details` command to obtain the location of OCR in an Oracle ASM disk group.

Example I-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 I-4 shows using the `ocrcheck -details` command to obtain a greater level of detail for the OCR configuration.

Example I-4 Using OCRCHECK to Obtain More OCR Information

```
$ ocrcheck -details

Status of Oracle Cluster Registry is as follows :
        Version                  :          4
        Total space (kbytes)     :     491684
        Used space (kbytes)      :      99128
        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 I-5 shows using the `ocrcheck -details` command to obtain backup file information.

Example I-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.

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.

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.

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 I-2 describes the OCRDUMP utility options and option descriptions.

Table I-2 OCRDUMP Options and Option Descriptions

Options	Description
file_name	The name of a file to which you want OCRDUMP to write output. By default, OCRDUMP writes output to a predefined output file named OCRDUMPFILE. The *file_name* option redirects OCRDUMP output to a file that you specify.

Table I-2 (Cont.) OCRDUMP Options and Option Descriptions

Options	Description
`-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 `OCRDUMPFILE`.
`-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 `-local` option with this option to view the contents of an OLR backup file.
`backup_file_name`	The name of the backup file with the content you want to view. You can query the backups using the `ocrconfig -showbackup` command. If the backup file is located in an Oracle ASM disk group, then ensure that the disk group exists and is mounted.
`-keyname` *`key_name`*	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 `OCRDUMPFILE` 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 `BACKUP00.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 : PROCR_ALL_ACCESS, GROUP_PERMISSION : PROCR_READ,
OTHER_PERMISSION : PROCR_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 : PROCR_ALL_ACCESS, GROUP_PERMISSION : PROCR_READ,
OTHER_PERMISSION : PROCR_READ, USER_NAME : user, GROUP_NAME : group}
```

J

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.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
>
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

Troubleshooting an Incompatible Fleet Patching and Provisioning Client Resource

If you manually upgrade an Oracle Clusterware 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 12*c* (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 -clienetdata 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**. 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.

crsctl query calog

Query the cluster resource activity logs matching specific criteria.

Syntax

```
crsctl query calog
[-aftertime "timestamp"]
[-beforetime "timestamp"]
[-days "number_of_days"]
[-duration "time_interval" | -follow]
[-filter "filter_expression"]
[-processname "writer_process"]
[-processid "writer_process_id"]
[-node "entity_hostname"]
[-fullfmt | -xmlfmt]
```

Parameters

Table J-1 crsctl query calog Command Parameters

Parameter	Description
`-aftertime "timestamp"`	Displays the activities logged after a specific time. Specify the timestamp in the `YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM]` or `YYYY-MM-DD` or `YYYY-MM` or `YYYY` or `HH24:MI:SS[.FF] [TZH:TZM]` format. `TZH` and `TZM` stands for time zone hour and minute, and `FF` stands for microseconds. 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. Use this parameter with `-beforetime` to query the activities logged at a specific time interval.

Table J-1 (Cont.) crsctl query calog Command Parameters

Parameter	Description
`-beforetime "timestamp"`	Displays the activities logged before a specific time. Specify the timestamp in the `YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM]` or `YYYY-MM-DD` or `YYYY-MM` or `YYYY` or `HH24:MI:SS[.FF] [TZH:TZM]` format. `TZH` and `TZM` stands for time zone hour and minute, and `FF` stands for microseconds. 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. Use this parameter with `-aftertime` to query the activities logged at a specific time interval.
`-days "number_of_days"`	Displays the activities logged in the last number of days specified. The number of days are specified as an integer value.
`-duration "time_interval" | -follow`	Use `-duration` to specify a time interval that you want to query when you use the `-aftertime` parameter. Specify the timestamp in the `DD HH:MM:SS` format. Use `-follow` to display a continuous stream of activities as they occur.
`-filter "filter_expression"`	Query any number of fields in the cluster resource activity log using the `-filter` parameter. To specify multiple filters, use a comma-delimited list of filter expressions surrounded by double quotation marks ("").
`-processname "writer_process"`	Displays the activities logged by a specific process identified by name.
`-processid "writer_process_id"`	Displays the activities logged by a specific process identified by ID.
`-node "entity_hostname"`	Displays the activities logged by a specific host.
`-fullfmt | -xmlfmt`	To display cluster resource activity log data, choose full or XML format.

Cluster Resource Activity Log Fields

Query any number of fields in the cluster resource activity log using the `-filter` parameter.

Table J-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.

Table J-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
writer_process_name	The name of the process that is writing to the cluster resource activity log.	When you query a specific process, CRSCTL 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.
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.

Table J-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
`is_planned`	Confirms if the activity is planned or not. For example, if a user issues the command `crsctl stop crs` on a node, then the stack stops and resources bounce. Running the `crsctl stop crs` command generates activities and logged in the `calog`. Since this is a planned action, the `is_planned` field is set to true (1). Otherwise, the `is_planned` 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 `ora.***`, is started or stopped, for example, then all those activities are logged in the cluster resource activity log. Since `ora.***` is an Oracle entity, the `entity_isoraentity` field is set to true (1). Otherwise the `entity_isoraentity` 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 *server*, for which the cluster activity log is written. Entity types that can be used to filter activities • `resource` • `resource_type` • `resource_group` • `server_category` • `ohasd` - activities generated by `ohasd` and resources it manages • `crsd` - activities generated by `crsd` and resources it manages In addition, GI components can choose to use their own names for entities when they write to activity log.	Query all the activities that match a specific entity.
`entity_name`	The name of the entity, for example, *foo* for which the cluster activity log is written.	Query all the cluster activities that match a specific entity name.

Table J-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
`entity_hostname`	The name of the host, for example, `node1`, 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, *cluster1* 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-mb1
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-mb1
nls_severity            : INFO
----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-mb1</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-mb1</entity_clustername>
    <nls_severity>INFO</nls_severity>
  </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-mb1</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-mb1</entity_clustername>
    <nls_severity>INFO</nls_severity>
  </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 : node1 : INFO : 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(node1)=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         : node1
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     : node1
entity_isoraentity      : 1
entity_type             : resource
entity_name             : ora.testdb.db
entity_hostname         : node1
entity_clustername      : cluster1-mb1
nls_severity            : INFO
----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 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

To query all the activities that were written by a specific process by name:

```
$  crsctl query calog -processname crsd.bin

2016-11-16 01:07:53.558000 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

To query all the activities that were written by a specific process by ID:

```
$ crsctl query calog -processid 6538

2016-11-16 01:07:53.558000 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

To query all the activities that were written by a specific node:

```
$ crsctl query calog -node node2
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
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 12*c* 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 12*c* 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 CRSCTL uses this name structure: `crsctl_`*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 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 12*c* 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). 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 J-3 lists and describes the parameters used with the `diagcollection.pl` script.

Table J-3 diagcollection.pl Script Parameters

Parameter	Description
`--collect`	Use this parameter with any of the following arguments: • `--crs`: Use this argument to collect Oracle Clusterware diagnostic information • `--acfs`: Use this argument to collect Oracle ACFS diagnostic information **Note:** You can only use this argument on UNIX systems. • `--all`: Use this argument to collect all diagnostic information except CHM (OS) data **Note:** This is the default. • `--chmos`: Use this argument to collect the following CHM diagnostic information `--incidenttime time`: Use this argument to collect CHM (OS) data from the specified time **Note:** The time format is MM/DD/YYYYHH24:MM:SS. `--incidentduration time`: Use this argument with `--incidenttime` to collect CHM (OS) data for the duration after the specified time **Note:** The time format is HH:MM. If you do not use `--incidentduration`, then all CHM (OS) data after the time you specify in `--incidenttime` is collected. • `--adr location`: The Automatic Diagnostic Repository Command Interpreter (ADRCI) uses this argument to specify a location in which to collect diagnostic information for ADR **See Also:** *Oracle Database Utilities* for more information about ADRCI • `--aftertime time`: Use this argument with the `--adr` argument to collect archives after the specified time **Note:** The time format is YYYYMMDDHHMISS24. • `--beforetime time`: Use this argument with the `--adr` argument to collect archives before the specified time **Note:** The time format is YYYYMMDDHHMISS24. • `--crshome path`: Use this argument to override the location of the Oracle Clusterware home **Note:** The `diagcollection.pl` script typically derives the location of the Oracle Clusterware home from the system configuration (either the `olr.loc` file or the Windows registry), so this argument is not required.

Table J-3 (Cont.) diagcollection.pl Script Parameters

Parameter	Description
`--clean`	Use this parameter to clean up the diagnostic information gathered by the `diagcollection.pl` script. **Note:** You cannot use this parameter with `--collect`.
`--coreanalyze`	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 ADVM/ACFS drivers. A system restart is recommended.
- ACFS-9428 Failed to load ADVM/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 -responseFile /u01/app/23.0.0/grid/install/response/gridinstall.rsp` command as the user who 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.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*

Testing Zone Delegation

To test zone delegation, use this procedure.

> **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.example.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 a CRS daemon process:

```
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.
```

Alert Messages Using Diagnostic Record Unique IDs

Beginning with Oracle Database 11*g* 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.