

Oracle® Solaris Cluster Data Services Planning and Administration Guide

Copyright © 2000, 2012, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

This software or hardware and documentation may provide access to or information on 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. 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.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

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

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Preface	9
1 Planning for Oracle Solaris Cluster Data Services	13
Configuration Guidelines for Oracle Solaris Cluster Data Services	14
Identifying Data Service Special Requirements	14
Determining the Location of the Application Binaries	14
Verifying the <code>nsswitch.conf</code> File Contents	15
Planning the Cluster File System Configuration	15
Enabling Oracle Solaris SMF Services to Run Under the Control of Oracle Solaris Cluster	16
Relationship Between Resource Groups and Device Groups	17
Understanding <code>HASstoragePlus</code>	17
Determining Whether Your Data Service Requires <code>HASstoragePlus</code>	18
Considerations for Installing and Configuring a Data Service	19
Overview of the Installation and Configuration Process	19
Installation and Configuration Task Flow	20
Example of Configuring a Failover Data Service	20
Tools for Data Service Resource Administration	21
<code>clsetup</code> Utility	21
Oracle Solaris Cluster Maintenance Commands	21
Summary by Task of Tools for Administering Data Service Resources	22
Standard Properties	23
Cluster Properties	23
Resource Type Properties	23
Resource Properties	23
Resource Group Properties	23
Resource Property Attributes	24
Node List Properties	24

Legal RGM Names	24
RGM Legal Names	24
RGM Values	26
2 Administering Data Service Resources	27
Overview of Tasks for Administering Data Service Resources	28
Configuring and Administering Oracle Solaris Cluster Data Services	31
Registering a Resource Type	32
▼ How to Register a Resource Type	32
Upgrading a Resource Type	33
▼ How to Install and Register an Upgrade of a Resource Type	34
▼ How to Migrate Existing Resources to a New Version of the Resource Type	35
▼ How to Unregister Older Unused Versions of the Resource Type	39
Downgrading a Resource Type	40
▼ How to Downgrade a Resource to an Older Version of Its Resource Type	41
Creating a Resource Group	42
▼ How to Create a Failover Resource Group	42
▼ How to Create a Scalable Resource Group	43
Configuring Failover and Scalable Data Services on Shared File Systems	45
▼ How to Configure a Failover Application Using the <code>ScalMountPoint</code> Resource	46
▼ How to Configure a Scalable Application Using the <code>ScalMountPoint</code> Resource	47
Tools for Adding Resources to Resource Groups	48
▼ How to Add a Logical Hostname Resource to a Resource Group by Using the <code>clsetup</code> Utility	49
▼ How to Add a Logical Hostname Resource to a Resource Group Using the Command-Line Interface	51
▼ How to Add a Shared Address Resource to a Resource Group by Using the <code>clsetup</code> Utility	53
▼ How to Add a Shared Address Resource to a Resource Group Using the Command-Line Interface	55
▼ How to Add a Failover Application Resource to a Resource Group	57
▼ How to Add a Scalable Application Resource to a Resource Group	59
Bringing Resource Groups Online	61
▼ How to Bring Resource Groups Online	62
Switching Resource Groups to Preferred Primaries	63
▼ How to Switch Resource Groups to Preferred Primaries	63

Enabling a Resource	64
▼ How to Enable a Resource	64
Quiescing Resource Groups	65
▼ How to Quiesce a Resource Group	65
▼ How to Quiesce a Resource Group Immediately	66
Suspending and Resuming the Automatic Recovery Actions of Resource Groups	66
Immediately Suspending Automatic Recovery by Killing Methods	67
▼ How to Suspend the Automatic Recovery Actions of a Resource Group	67
▼ How to Suspend the Automatic Recovery Actions of a Resource Group Immediately	68
▼ How to Resume the Automatic Recovery Actions of a Resource Group	68
Disabling and Enabling Resource Monitors	68
▼ How to Disable a Resource Fault Monitor	69
▼ How to Enable a Resource Fault Monitor	69
Removing Resource Types	70
▼ How to Remove a Resource Type	70
Removing Resource Groups	71
▼ How to Remove a Resource Group	72
Removing Resources	73
▼ How to Remove a Resource	73
Switching the Current Primary of a Resource Group	74
▼ How to Switch the Current Primary of a Resource Group	74
Disabling Resources and Moving Their Resource Group Into the UNMANAGED State	76
▼ How to Disable a Resource and Move Its Resource Group Into the UNMANAGED State	76
Displaying Resource Type, Resource Group, and Resource Configuration Information	78
Changing Resource Type, Resource Group, and Resource Properties	78
▼ How to Change Resource Type Properties	79
▼ How to Change Resource Group Properties	80
▼ How to Change Resource Properties	81
▼ How to Change Resource Dependency Properties	82
▼ How to Modify a Logical Hostname Resource or a Shared Address Resource	85
Clearing the STOP_FAILED Error Flag on Resources	86
▼ How to Clear the STOP_FAILED Error Flag on Resources	86
Clearing the Start_failed Resource State	88
▼ How to Clear a Start_failed Resource State by Switching Over a Resource Group	88
▼ How to Clear a Start_failed Resource State by Restarting a Resource Group	90
▼ How to Clear a Start_failed Resource State by Disabling and Enabling a Resource	91

Upgrading a Preregistered Resource Type	93
Information for Registering the New Resource Type Version	93
Information for Migrating Existing Instances of the Resource Type	93
Reregistering Preregistered Resource Types After Inadvertent Deletion	94
▼ How to Reregister Preregistered Resource Types After Inadvertent Deletion	94
Adding or Removing a Node to or From a Resource Group	95
Adding a Node to a Resource Group	95
Removing a Node From a Resource Group	98
Synchronizing the Startups Between Resource Groups and Device Groups	103
Managed Entity Monitoring by HASStoragePlus	104
Additional Administrative Tasks to Configure HASStoragePlus Resources for a Zone Cluster	107
▼ How to Set Up the HASStoragePlus Resource Type for New Resources	107
▼ How to Set Up the HASStoragePlus Resource Type for Existing Resources	109
Configuring an HASStoragePlus Resource for Cluster File Systems	110
Sample Entries in /etc/vfstab for Cluster File Systems	110
▼ How to Set Up the HASStoragePlus Resource for Cluster File Systems	111
▼ How to Delete an HASStoragePlus Resource Type for Cluster File Systems	112
Enabling Highly Available Local File Systems	112
Configuration Requirements for Highly Available Local File Systems	113
Format of Device Names for Devices Without a Volume Manager	114
Sample Entries in /etc/vfstab for Highly Available Local File Systems	114
▼ How to Set Up the HASStoragePlus Resource Type by Using the clsetup Utility	115
▼ How to Set Up the HASStoragePlus Resource Type to Make File Systems Highly Available Other Than Solaris ZFS	118
▼ How to Set Up the HASStoragePlus Resource Type to Make a Local Solaris ZFS File System Highly Available	119
▼ How to Delete an HASStoragePlus Resource That Makes a Local Solaris ZFS Highly Available	122
Sharing a Highly Available Local File System Across Zone Clusters	122
Configuration Requirements for Sharing a Highly Available Local File System Directory to a Zone Cluster	123
▼ How to Set Up the HASStorage Plus Resource Type to Share a Highly Available Local File System Directory to a Zone Cluster	123
Modifying Online the Resource for a Highly Available Local File System	126
▼ How to Add File Systems Other Than Solaris ZFS to an Online HASStoragePlus Resource	127

- ▼ How to Remove File Systems Other Than Solaris ZFS From an Online HASStoragePlus Resource 129
- ▼ How to Add a Solaris ZFS Storage Pool to an Online HASStoragePlus Resource 131
- ▼ How to Remove a Solaris ZFS Storage Pool From an Online HASStoragePlus Resource .. 132
- Changing a ZFS Pool Configuration That is Managed by an HASStoragePlus Resource ... 133
- ▼ How to Change a ZFS Pool Configuration That is Managed by an HASStoragePlus Resource in an Offline State 133
- ▼ How to Change a ZFS Pool Configuration That is Managed by an Online HASStoragePlus Resource 134
- ▼ How to Recover From a Fault After Modifying the FileSystemMountPoints Property of an HASStoragePlus Resource 135
- ▼ How to Recover From a Fault After Modifying the Zpools Property of an HASStoragePlus Resource 136
- Changing the Cluster File System to a Local File System in an HASStoragePlus Resource 137
- ▼ How to Change the Cluster File System to Local File System in an HASStoragePlus Resource 137
- Upgrading the HASStoragePlus Resource Type 138
- Information for Registering the New Resource Type Version 138
- Information for Migrating Existing Instances of the Resource Type 138
- Distributing Online Resource Groups Among Cluster Nodes 139
- Resource Group Affinities 139
- Enforcing Collocation of a Resource Group With Another Resource Group 141
- Specifying a Preferred Collocation of a Resource Group With Another Resource Group 142
- Distributing a Set of Resource Groups Evenly Among Cluster Nodes 143
- Specifying That a Critical Service Has Precedence 144
- Delegating the Failover or Switchover of a Resource Group 145
- Combining Affinities Between Resource Groups 146
- Zone Cluster Resource Group Affinities 147
- Configuring the Distribution of Resource Group Load Across Nodes 148
- ▼ How to Configure Load Limits for a Node 149
- ▼ How to Set Priority for a Resource Group 150
- ▼ How to Set Load Factors for a Resource Group 151
- ▼ How to Set Preemption Mode for a Resource Group 152
- ▼ How to Concentrate Load Onto Fewer Nodes in the Cluster 153
- Enabling Oracle Solaris SMF Services to Run With Oracle Solaris Cluster 154
- ▼ Encapsulating an SMF Service Into a Failover Proxy Resource Configuration 155
- ▼ Encapsulating an SMF Service Into a Multi-Master Proxy Resource Configuration 157

- ▼ Encapsulating an SMF Service Into a Scalable Proxy Resource Configuration 160
- Tuning Fault Monitors for Oracle Solaris Cluster Data Services 163
 - Setting the Interval Between Fault Monitor Probes 164
 - Setting the Timeout for Fault Monitor Probes 164
 - Defining the Criteria for Persistent Faults 165
 - Specifying the Failover Behavior of a Resource 166

- Index** 167

Preface

Oracle Solaris Cluster Data Services Planning and Administration Guide explains how to install and configure Oracle Solaris Cluster data services.

Note – This Oracle Solaris Cluster release supports systems that use the SPARC and x86 families of processor architectures. In this document, “x86” refers to the larger family of x86 compatible products. Information in this document pertains to all platforms unless otherwise specified.

This document is intended for system administrators with extensive knowledge of Oracle software and hardware. Do not use this document as a planning or presales guide. Before reading this document, you should have already determined your system requirements and purchased the appropriate equipment and software.

The instructions in this book assume knowledge of the Oracle Solaris Operating System and expertise with the volume-manager software that is used with Oracle Solaris Cluster software.

Bash is the default shell for Oracle Solaris 11. Machine names shown with the Bash shell prompt are displayed for clarity.

Using UNIX Commands

This document contains information about commands that are specific to installing and configuring Oracle Solaris Cluster data services. The document does *not* contain comprehensive information about basic UNIX commands and procedures, such as shutting down the system, booting the system, and configuring devices. Information about basic UNIX commands and procedures is available from the following sources:

- Online documentation for the Oracle Solaris Operating System
- Oracle Solaris Operating System man pages
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Description	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

Related Documentation

Information about related Oracle Solaris Cluster topics is available in the documentation that is listed in the following table. All Oracle Solaris Cluster documentation is available at <http://www.oracle.com/technetwork/indexes/documentation/index.html>.

Topic	Documentation
Hardware installation and administration	<i>Oracle Solaris Cluster 4.1 Hardware Administration Manual</i> Individual hardware administration guides
Concepts	<i>Oracle Solaris Cluster Concepts Guide</i>
Software installation	<i>Oracle Solaris Cluster Software Installation Guide</i>
Data service installation and administration	<i>Oracle Solaris Cluster Data Services Planning and Administration Guide</i> and individual data service guides
Data service development	<i>Oracle Solaris Cluster Data Services Developer's Guide</i>
System administration	<i>Oracle Solaris Cluster System Administration Guide</i> <i>Oracle Solaris Cluster Quick Reference</i>
Software upgrade	<i>Oracle Solaris Cluster Upgrade Guide</i>
Error messages	<i>Oracle Solaris Cluster Error Messages Guide</i>
Command and function references	<i>Oracle Solaris Cluster Reference Manual</i> <i>Oracle Solaris Cluster Data Services Reference Manual</i> <i>Oracle Solaris Cluster Geographic Edition Reference Manual</i> <i>Oracle Solaris Cluster Quorum Server Reference Manual</i>
Compatible software	Oracle Solaris Cluster Compatibility Guide available at the Oracle Solaris Cluster Technical Resources page

Access to Oracle Support

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

Getting Help

If you have problems installing or using Oracle Solaris Cluster, contact your service provider and provide the following information.

- Your name and email address (if available)
- Your company name, address, and phone number
- The model number and serial number of your systems
- The release number of the operating environment (for example, Oracle Solaris 11)
- The release number of Oracle Solaris Cluster (for example, Oracle Solaris Cluster 4.1)

Use the following commands to gather information about your system for your service provider.

Command	Function
<code>prtconf -v</code>	Displays the size of the system memory and reports information about peripheral devices
<code>psrinfo -v</code>	Displays information about processors
<code>pkg list</code>	Reports which packages are installed
<code>prtdiag -v</code>	Displays system diagnostic information
<code>/usr/cluster/bin/clnode show-rev -v</code>	Displays Oracle Solaris Cluster release and package version information for each node

Also have available the contents of the `/var/adm/messages` file.

Planning for Oracle Solaris Cluster Data Services

This chapter provides planning information and guidelines to install and configure Oracle Solaris Cluster data services. This chapter contains the following sections.

- “Configuration Guidelines for Oracle Solaris Cluster Data Services” on page 14
- “Relationship Between Resource Groups and Device Groups” on page 17
- “Understanding HAS_{storagePlus}” on page 17
- “Considerations for Installing and Configuring a Data Service” on page 19
- “Overview of the Installation and Configuration Process” on page 19
- “Tools for Data Service Resource Administration” on page 21
- “Standard Properties” on page 23
- “Node List Properties” on page 24
- “Legal RGM Names” on page 24

For introductory information about data services, see *Oracle Solaris Cluster Concepts Guide*.

Oracle Solaris Cluster software can provide service only for those data services that are either supplied with the Oracle Solaris Cluster product or are created with the Oracle Solaris Cluster data services application programming interfaces (APIs).

If an Oracle Solaris Cluster data service is not provided for your application, consider developing a custom data service for the application. To develop a custom data service, use the Oracle Solaris Cluster data services APIs. For more information, see *Oracle Solaris Cluster Data Services Developer’s Guide*.

Note – Oracle Solaris Cluster does not provide a data service for the `sendmail(1M)` subsystem. The `sendmail` subsystem can run on the individual cluster nodes, but the `sendmail` functionality is not highly available. This restriction applies to all the `sendmail` functionality, including the functionality of mail delivery and mail routing, queuing, and retry.

Configuration Guidelines for Oracle Solaris Cluster Data Services

This section provides configuration guidelines for Oracle Solaris Cluster data services.

Identifying Data Service Special Requirements

Identify requirements for all of the data services **before** you begin Oracle Solaris and Oracle Solaris Cluster installation. Failure to do so might result in installation errors that require that you completely reinstall the Oracle Solaris and Oracle Solaris Cluster software.

For example, the Oracle Data Guard option of Oracle Solaris Cluster Support for Oracle Real Application Clusters has special requirements for the hostnames that you use in the cluster. HA for SAP also has special requirements. You must accommodate these requirements before you install Oracle Solaris Cluster software because you cannot change hostnames after you install Oracle Solaris Cluster software.

Note – Some Oracle Solaris Cluster data services are not supported for use in x86-based clusters. For more information, see the release notes for your release of Oracle Solaris Cluster at (<http://www.oracle.com/technetwork/indexes/documentation/index.html>).

Determining the Location of the Application Binaries

You can install the application software and application configuration files on one of the following locations.

- **The local disks of each cluster node** – Placing the software and configuration files on the individual cluster nodes provides the following advantage. You can upgrade application software later without shutting down the service.

The disadvantage is that you then have several copies of the software and configuration files to maintain and administer.
- **The cluster file system** – If you put the application binaries on the cluster file system, you have only one copy to maintain and manage. However, you must shut down the data service in the entire cluster to upgrade the application software. If you can spare a short period of downtime for upgrades, place a single copy of the application and configuration files on the cluster file system.

For information about how to create cluster file systems, see “[Planning Global Devices, Device Groups, and Cluster File Systems](#)” in *Oracle Solaris Cluster Software Installation Guide*.

- **Highly available local file system** – Using `HASstoragePlus`, you can integrate your local file system into the Oracle Solaris Cluster environment, making the local file system highly available. `HASstoragePlus` provides additional file system capabilities such as checks, mounts, and unmounts that enable Oracle Solaris Cluster to fail over local file systems. To fail over, the local file system must reside on global disk groups with affinity switchovers enabled.

For information about how to use the `HASstoragePlus` resource type, see [“Enabling Highly Available Local File Systems” on page 112](#).

Verifying the `nsswitch.conf` File Contents

The `nsswitch.conf` file is the configuration file for name-service lookups. This file determines the following information:

- The databases within the Oracle Solaris environment to use for name-service lookups
- The order in which the databases are to be consulted

Some data services require that you direct “group” lookups to “files” first. For these data services, change the “group” line in the `nsswitch.conf` file so that the “files” entry is listed first. See the documentation for the data service that you plan to configure to determine whether you need to change the “group” line. The `scinstall` utility automatically configures the `nsswitch.conf` file for you. If you manually modify the `nsswitch.conf` file, you must export the new `nsswitch` configuration information.

Planning the Cluster File System Configuration

Depending on the data service, you might need to configure the cluster file system to meet Oracle Solaris Cluster requirements. To determine whether any special considerations apply, see the documentation for the data service that you plan to configure.

For information about planning cluster file systems, see [“Planning Global Devices, Device Groups, and Cluster File Systems” in *Oracle Solaris Cluster Software Installation Guide*](#).

The resource type `HASstoragePlus` enables you to use a highly available local file system in an Oracle Solaris Cluster environment that is configured for failover. For information about setting up the `HASstoragePlus` resource type, see [“Enabling Highly Available Local File Systems” on page 112](#).

Enabling Oracle Solaris SMF Services to Run Under the Control of Oracle Solaris Cluster

The Service Management Facility (SMF) enables you to automatically start and restart SMF services during a node boot or service failure. This feature is similar to the Oracle Solaris Cluster Resource Group Manager (RGM), which facilitates high availability and scalability for cluster applications. SMF services and RGM features are complementary to each other.

Oracle Solaris Cluster includes three SMF proxy resource types that can be used to enable SMF services to run with Oracle Solaris Cluster in a failover, multi-master, or scalable configuration. The SMF proxy resource types enables you to encapsulate a set of interrelated SMF services into a single resource, *SMF proxy resource* to be managed by Oracle Solaris Cluster. In this feature, SMF manages the availability of SMF services on a single node. Oracle Solaris Cluster provides cluster-wide high availability and scalability of the SMF services.

For information about how to encapsulate these services, see [“Enabling Oracle Solaris SMF Services to Run With Oracle Solaris Cluster” on page 154](#).

You might require Oracle Solaris Cluster to make highly available an application other than NFS or DNS that is integrated with the Solaris Service Management Facility (SMF). To ensure that Oracle Solaris Cluster can restart or fail over the application correctly after a failure, you must disable SMF service instances for the application as follows:

- For any application other than NFS or DNS, disable the SMF service instance on all potential primary nodes for the Oracle Solaris Cluster resource that represents the application.
- If multiple instances of the application share any component that you require Oracle Solaris Cluster to monitor, disable all service instances of the application. Examples of such components are daemons, file systems, and devices.

Note – If you do not disable the SMF service instances of the application, both the Solaris SMF and Oracle Solaris Cluster might attempt to control the startup and shutdown of the application. As a result, the behavior of the application might become unpredictable.

For more information, see the following documentation:

- [“How to Disable a Service Instance” in *Managing Services and Faults in Oracle Solaris 11.1*](#)
- [Oracle Solaris Cluster Data Service for Network File System \(NFS\) Guide](#)
- [Oracle Solaris Cluster Concepts Guide](#)

Relationship Between Resource Groups and Device Groups

Oracle Solaris Cluster uses the concept of **node lists** for device groups and resource groups. Node lists are ordered lists of primary nodes, which are potential masters of the disk device group or resource group. Oracle Solaris Cluster uses a **failback policy** to determine the behavior of Oracle Solaris Cluster in response to the following set of conditions:

- A node that has failed and left the cluster rejoins the cluster.
- The node that is rejoining the cluster appears earlier in the node list than the current primary node.

If failback is set to True, the device group or resource group is switched off the current primary and switched onto the rejoining node, making the rejoining node the new primary.

For example, assume that you have a disk device group, `disk-group-1`, that has nodes `phys-schost-1` and `phys-schost-2` in its node list, with the failback policy set to Enabled. Assume that you also have a failover resource group, `resource-group-1`, which uses `disk-group-1` to hold its application data. When you set up `resource-group-1`, also specify `phys-schost-1` and `phys-schost-2` for the resource group's node list, and set the failback policy to True.

To ensure high availability of a scalable resource group, make the scalable resource group's node list a superset of the node list for the disk device group. This setting ensures that the nodes that are directly connected to the disks are also nodes that can run the scalable resource group. The advantage is that, when at least one cluster node connected to the data is up, the scalable resource group runs on that same node, making the scalable services available also.

For more information about the relationship between device groups and resource groups, see [“Device Groups” in *Oracle Solaris Cluster Concepts Guide*](#).

For information about how to set up device groups, see the following documentation:

- [“Planning Device Groups” in *Oracle Solaris Cluster Software Installation Guide*](#)

Understanding HASStoragePlus

The HASStoragePlus resource type can be used to configure the following options.

- Coordinate the boot order of disk devices and resource groups. Other resources in the resource group that contains the HASStoragePlus resource are brought online *only* after the disk device resources become available.
- With `AffinityOn` set to True, enforce collocation of resource groups and device groups on the same node. This enforced collocation enhances the performance of disk-intensive data services.
- Monitor entities managed by the HASP resource, including global devices, file systems, and ZFS storage pools.

In addition, HAStoragePlus is capable of mounting local and global file systems. For more information, see [“Planning the Cluster File System Configuration” on page 15](#).

Note – If the device group is switched to another node while the HAStoragePlus resource is online, AffinityOn has no effect. The resource group does **not** migrate with the device group. However, if the resource group is switched to another node, the setting of AffinityOn to True causes the device group to follow the resource group to the new node.

See [“Synchronizing the Startups Between Resource Groups and Device Groups” on page 103](#) for information about the relationship between device groups and resource groups.

See [“Enabling Highly Available Local File Systems” on page 112](#) for procedures for mounting of file systems such as Solaris ZFS in a local mode. The `SUNW.HAStoragePlus(5)` man page provides additional details.

Determining Whether Your Data Service Requires HAStoragePlus

The following types of data services require HAStoragePlus:

- Data services with nodes that are not directly connected to storage
- Data services that are disk intensive

Data Services With Nodes That Are Not Directly Connected to Storage

Some nodes in the node list of a data service's resource group might not be directly connected to the storage. In this situation, you must coordinate the boot order between the storage and the data service. To meet this requirement, configure the resource group as follows:

- Configure HAStoragePlus resources in the resource group.
- Set the dependency of the other data service resources to the HAStoragePlus resource.

Data Services That Are Disk Intensive

Some data services, such as Oracle Solaris Cluster HA for Oracle and Oracle Solaris Cluster HA for NFS are disk intensive. If your data service is disk intensive, ensure that the resource groups and device groups are colocated on the same node. To meet this requirement, perform the following tasks:

- Add an HAStoragePlus resource to your data service resource group
- Switch the HAStoragePlus resource online
- Set the dependency of your data service resources to the HAStoragePlus resource
- Set AffinityOn to True

Note – The failback settings must be identical for both the resource group and device groups.

Some data services are not disk intensive. For example, HA for DNS (which reads all of its files at startup) is not disk intensive. If your data service is **not** disk intensive, configuring the `HASStoragePlus` resource type is optional.

Considerations for Installing and Configuring a Data Service

Use the information in this section to plan the installation and configuration of any data service. The information in this section encourages you to think about the impact your decisions have on the installation and configuration of any data service. For specific considerations for a data service, see the documentation for the data service.

- Retries within the I/O subsystem during disk failures might cause applications whose data services are disk intensive to experience delays. Disk-intensive data services are I/O intensive and have a large number of disks configured in the cluster. An I/O subsystem might require several minutes to retry and recover from a disk failure. This delay can cause Oracle Solaris Cluster to fail over the application to another node, even though the disk might have eventually recovered on its own. To avoid failover during these instances, consider increasing the default probe timeout of the data service. If you need more information or help with increasing data service timeouts, contact your local support engineer.
- For better performance, install and configure your data service on the cluster nodes with direct connection to the storage.
- Client applications that run on cluster nodes should not map to logical IP addresses of an HA data service. After a failover, these logical IP addresses might no longer exist, leaving the client without a connection.

Overview of the Installation and Configuration Process

Use the following procedures to install and configure a data service.

- Install the data service packages from the installation medium on which the packages are supplied.
- Install and configure the application to run in the cluster environment.
- Configure the resources and resource groups that the data service uses. When you configure a data service, specify the resource types, resources, and resource groups that the Resource Group Manager (RGM) is to manage. The documentation for the individual data services describes these procedures.

Installation and Configuration Task Flow

The following table summarizes the tasks for installing and configuring Oracle Solaris Cluster data services. The table also provides cross-references to detailed instructions for performing the tasks.

TABLE 1-1 Tasks for Installing and Configuring Oracle Solaris Cluster Data Services

Task	Instructions
Install the Oracle Solaris and Oracle Solaris Cluster software, including data service software	Oracle Solaris Cluster Software Installation Guide
Set up multihost disks	Oracle Solaris Cluster Software Installation Guide
Plan resources and resource groups	“Creating a Resource Group” on page 42
Decide the location for application binaries, and configure the <code>nsswitch.conf</code> file	“Determining the Location of the Application Binaries” on page 14 “Verifying the <code>nsswitch.conf</code> File Contents” on page 15
Install and configure the application software	The appropriate Oracle Solaris Cluster data services book
Register and configure the data service	The appropriate Oracle Solaris Cluster data services book

Example of Configuring a Failover Data Service

This example summarizes how to set up the resource types, resources, and resource groups that a failover data service for the Oracle application requires. For complete instructions for configuring the data service for the Oracle application, see [Oracle Solaris Cluster Data Service for Oracle Guide](#).

The principal difference between this example and an example of a scalable data service is as follows: In addition to the failover resource group that contains the network resources, a scalable data service requires a separate resource group (*scalable resource group*) for the application resources.

The Oracle application has two components—a server and a listener. Oracle supplies the Oracle Solaris Cluster HA for Oracle data service, and therefore these components have already been mapped into Oracle Solaris Cluster resource types. Both of these resource types are associated with resources and resource groups.

Because this example is a failover data service, the example uses logical hostname network resources, which are the IP addresses that fail over from a primary node to a secondary node.

Place the logical hostname resources into a failover resource group, and then place the Oracle server resources and listener resources into the same resource group. This ordering enables all of the resources to fail over as a group.

For Oracle Solaris Cluster HA for Oracle to run on the cluster, you must define the following objects:

- LogicalHostname resource type – This resource type is built in, and therefore you do not need to explicitly register the resource type.
- Oracle resource types – Oracle Solaris Cluster HA for Oracle defines two Oracle resource types: a database server and a listener.
- Logical hostname resources – These resources host the IP addresses that fail over in a node failure.
- Oracle resources – You must specify two resource instances for Oracle Solaris Cluster HA for Oracle— a server and a listener.
- Failover resource group – This container is composed of the Oracle server and listener and logical hostname resources that will fail over as a group.

Tools for Data Service Resource Administration

This section describes the tools that you can use to perform installation and configuration tasks.

c1setup Utility

The `c1setup(1CL)` utility is a menu-driven interface that you can use for general Oracle Solaris Cluster administration. You can also use this utility to configure data service resources and resource groups. Select option 2 from the `c1setup` main menu to launch the Resource Group Menu.

Oracle Solaris Cluster Maintenance Commands

You can use the Oracle Solaris Cluster maintenance commands to register and configure data service resources. See the procedure for how to register and configure your data service in the book for the data service. If, for example, you are using Oracle Solaris Cluster HA for Oracle, see “Registering and Configuring HA for Oracle” in *Oracle Solaris Cluster Data Service for Oracle Guide*.

For more information about how to use the commands to administer data service resources, see Chapter 2, “Administering Data Service Resources.”

Summary by Task of Tools for Administering Data Service Resources

The following table summarizes the tasks that the `clsetup` utility can perform to administer data service resources. For more information about these tasks and for details about how to use the command line to complete related procedures, see [Chapter 2, “Administering Data Service Resources.”](#)

TABLE 1-2 Tasks for Administering Data Service Resources

Task
Register a resource type
Create a resource group
Add a resource to a resource group
Suspend the automatic recovery actions of a resource group
Resume the automatic recovery actions of a resource group
Bring a resource group online
Remove a resource group
Remove a resource
Switch the current primary of a resource group
Enable a resource
Disable a resource
Move a resource group to the unmanaged state
Display resource type, resource group, and resource configuration information
Change resource properties
Set a resource dependency
Clear the <code>STOP_FAILED</code> error flag on resources
Clear the <code>START_FAILED</code> resource state for a resource
Add a node to a resource group

Standard Properties

You can specify the following standard cluster, resource type, resource, and resource group properties when configuring data services. Resource property attributes are also available to change system-defined properties and create extension properties.

This section lists the following properties:

- Cluster Properties
- Resource Type Properties
- Resource Properties
- Resource Group Properties
- Resource Property Attributes

Cluster Properties

Cluster properties are used to manage data services. For information about cluster properties, see the [cluster\(1CL\)](#) man page.

Resource Type Properties

Resource type properties cannot be updated by administrative utilities with the exception of `Installed_nodes` and `RT_system`. `Installed_nodes` cannot be declared in the RTR file and can only be set by the cluster administrator. `RT_system` can be assigned an initial value in the RTR file, and can also be set by the cluster administrator.

For information about each resource type property that is defined by Oracle Solaris Cluster software, see the [rt_properties\(5\)](#) man page.

Resource Properties

For information about each resource property that is defined by Oracle Solaris Cluster software, see the [r_properties\(5\)](#) man page.

Resource Group Properties

For information about each resource group property that is defined by Oracle Solaris Cluster software, see the [rg_properties\(5\)](#) man page.

Resource Property Attributes

You can use resource property attributes to change system-defined properties or to create extension properties. For information about each property, see the [property_attributes\(5\)](#) man page.

Node List Properties

You can specify the following node list properties when configuring data services:

- `Installed_nodes` Property – See the [rt_properties\(5\)](#) man page for more information.
- `NodeList` Property – See the [rg_properties\(5\)](#) man page for more information.
- `Auxnodelist` Property – See the [clressharedaddress\(1CL\)](#) man page for more information.

Legal RGM Names

This section lists the requirements for legal characters for Resource Group Manager (RGM) names and values.

This section covers the following topics:

- “RGM Legal Names” on page 24
- “RGM Values” on page 26

RGM Legal Names

RGM names fall into the following categories:

- Resource group names
- Resource type names
- Resource names
- Property names
- Enumeration literal names

Rules for Names Except Resource Type Names

Except for resource type names, all names must comply with these rules:

- Names must be in ASCII.
- Names must start with a letter.
- Names can contain uppercase and lowercase letters, digits, dashes (-), and underscores (_).

- The maximum number of characters that you can use in a name is 255.

Format of Resource Type Names

The format of the complete name of a resource type depends on the resource type, as follows:

- If the resource type's resource type registration (RTR) file contains the `#$upgrade` directive, the format is as follows:

vendor-id.base-rt-name:rt-version

- If the resource type's RTR file does *not* contain the `#$upgrade` directive, the format is as follows:

vendor-id.base-rt-name

A period separates *vendor-id* and *base-rt-name*. A colon separates *base-rt-name* and *rt-version*.

The variable elements in this format are as follows:

<i>vendor-id</i>	Specifies the vendor ID prefix, which is the value of the <code>Vendor_id</code> resource type property in the RTR file. If you are developing a resource type, choose a vendor ID prefix that uniquely identifies the vendor, such as your company's stock ticker symbol.
<i>base-rt-name</i>	Specifies the base resource type name, which is the value of the <code>Resource_type</code> resource type property in the RTR file.
<i>rt-version</i>	Specifies the version suffix, which is the value of the <code>RT_version</code> resource type property in the RTR file. The version suffix is <i>only</i> part of the complete resource type name if the RTR file contains the <code>#\$upgrade</code> directive.

Note – If only one version of a base resource type name is registered, you do not have to use the complete name in administrative commands. You can omit the vendor ID prefix, the version number suffix, or both.

For more information, see [“Resource Type Properties” on page 23](#).

EXAMPLE 1-1 Complete Name of a Resource Type With the `#$upgrade` Directive

This example shows the complete name of a resource type for which properties in the RTR file are set, as follows:

- `Vendor_id=ORCL`
- `Resource_type=sample`
- `RT_version=2.0`

The complete name of the resource type that is defined by this RTR file is as follows:

EXAMPLE 1-1 Complete Name of a Resource Type With the #\$\$upgrade Directive *(Continued)*

```
ORCL.sample:2.0
```

EXAMPLE 1-2 Complete Name of a Resource Type Without the #\$\$upgrade Directive

This example shows the complete name of a resource type for which properties in the RTR file are set, as follows:

- Vendor_id=ORCL
- Resource_type=abc

The complete name of the resource type that is defined by this RTR file is as follows:

```
ORCL.abc
```

RGM Values

RGM values fall into two categories: property values and description values. Both categories share the same rules:

- Values must be in ASCII.
- The maximum length of a value is 4 megabytes minus 1, that is, 4,194,303 bytes.
- Values cannot contain the following characters:
 - Null
 - Newline
 - Comma (,)
 - Semicolon (;)

Administering Data Service Resources

This chapter describes how to use the Oracle Solaris Cluster maintenance commands to manage resources, resource groups, and resource types within the cluster. To determine if you can use other tools to complete a procedure, see [“Tools for Data Service Resource Administration”](#) on page 21.

For overview information about resource types, resource groups, and resources, see [Chapter 1, “Planning for Oracle Solaris Cluster Data Services,”](#) and *Oracle Solaris Cluster Concepts Guide*.

This chapter contains the following sections.

- [“Overview of Tasks for Administering Data Service Resources”](#) on page 28
- [“Configuring and Administering Oracle Solaris Cluster Data Services”](#) on page 31
- [“Registering a Resource Type”](#) on page 32
- [“Upgrading a Resource Type”](#) on page 33
- [“Downgrading a Resource Type”](#) on page 40
- [“Creating a Resource Group”](#) on page 42
- [“Configuring Failover and Scalable Data Services on Shared File Systems”](#) on page 45
- [“Tools for Adding Resources to Resource Groups”](#) on page 48
- [“Bringing Resource Groups Online”](#) on page 61
- [“Switching Resource Groups to Preferred Primaries”](#) on page 63
- [“Enabling a Resource”](#) on page 64
- [“Quiescing Resource Groups”](#) on page 65
- [“Suspending and Resuming the Automatic Recovery Actions of Resource Groups”](#) on page 66
- [“Disabling and Enabling Resource Monitors”](#) on page 68
- [“Removing Resource Types”](#) on page 70
- [“Removing Resource Groups”](#) on page 71
- [“Removing Resources”](#) on page 73
- [“Switching the Current Primary of a Resource Group”](#) on page 74
- [“Disabling Resources and Moving Their Resource Group Into the UNMANAGED State”](#) on page 76

- “Displaying Resource Type, Resource Group, and Resource Configuration Information” on page 78
- “Changing Resource Type, Resource Group, and Resource Properties” on page 78
- “Clearing the STOP_FAILED Error Flag on Resources” on page 86
- “Clearing the Start_failed Resource State” on page 88
- “Upgrading a Preregistered Resource Type” on page 93
- “Reregistering Preregistered Resource Types After Inadvertent Deletion” on page 94
- “Adding or Removing a Node to or From a Resource Group” on page 95
- “Synchronizing the Startups Between Resource Groups and Device Groups” on page 103
- “Configuring an HAStoragePlus Resource for Cluster File Systems” on page 110
- “Enabling Highly Available Local File Systems” on page 112
- “Sharing a Highly Available Local File System Across Zone Clusters” on page 122
- “Modifying Online the Resource for a Highly Available Local File System” on page 126
- “Changing the Cluster File System to a Local File System in an HAStoragePlus Resource” on page 137
- “Upgrading the HAStoragePlus Resource Type” on page 138
- “Distributing Online Resource Groups Among Cluster Nodes” on page 139
- “Configuring the Distribution of Resource Group Load Across Nodes” on page 148
- “Enabling Oracle Solaris SMF Services to Run With Oracle Solaris Cluster” on page 154
- “Tuning Fault Monitors for Oracle Solaris Cluster Data Services” on page 163

Overview of Tasks for Administering Data Service Resources

The following table summarizes the tasks for installing and configuring Oracle Solaris Cluster data services. The table also provides cross-references to detailed instructions for performing the tasks.

TABLE 2-1 Tasks for Administering Data Service Resources

Task	Instructions
Register a resource type	“How to Register a Resource Type” on page 32
Upgrade a resource type	“How to Install and Register an Upgrade of a Resource Type” on page 34
Downgrade a resource type	“How to Downgrade a Resource to an Older Version of Its Resource Type” on page 41
Create failover or scalable resource groups	“How to Create a Failover Resource Group” on page 42 “How to Create a Scalable Resource Group” on page 43

TABLE 2-1 Tasks for Administering Data Service Resources (Continued)

Task	Instructions
Add logical hostnames or shared addresses and data service resources to resource groups	<p>“How to Add a Logical Hostname Resource to a Resource Group by Using the <code>clsetup</code> Utility” on page 49</p> <p>“How to Add a Logical Hostname Resource to a Resource Group Using the Command-Line Interface” on page 51</p> <p>“How to Add a Shared Address Resource to a Resource Group by Using the <code>clsetup</code> Utility” on page 53</p> <p>“How to Add a Shared Address Resource to a Resource Group Using the Command-Line Interface” on page 55</p> <p>“How to Add a Failover Application Resource to a Resource Group” on page 57</p> <p>“How to Add a Scalable Application Resource to a Resource Group” on page 59</p>
Enable resources and resource monitors, manage the resource group, and bring the resource group and its associated resources online	<p>“How to Enable a Resource” on page 64</p> <p>“How to Bring Resource Groups Online” on page 62</p> <p>“How to Switch Resource Groups to Preferred Primaries” on page 63</p>
Quiesce a resource group	<p>“How to Quiesce a Resource Group” on page 65</p> <p>“How to Quiesce a Resource Group Immediately” on page 66</p>
Suspend and resume automatic recovery actions of a resource group	<p>“How to Suspend the Automatic Recovery Actions of a Resource Group” on page 67</p> <p>“How to Suspend the Automatic Recovery Actions of a Resource Group Immediately” on page 68</p> <p>“How to Resume the Automatic Recovery Actions of a Resource Group” on page 68</p>
Disable and enable resource monitors independent of the resource	<p>“How to Disable a Resource Fault Monitor” on page 69</p> <p>“How to Enable a Resource Fault Monitor” on page 69</p>
Remove resource types from the cluster	“How to Remove a Resource Type” on page 70
Remove resource groups from the cluster	“How to Remove a Resource Group” on page 72
Remove resources from resource groups	“How to Remove a Resource” on page 73
Switch the primary for a resource group	“How to Switch the Current Primary of a Resource Group” on page 74
Disable resources and move their resource group into the UNMANAGED state	“How to Disable a Resource and Move Its Resource Group Into the UNMANAGED State” on page 76

TABLE 2-1 Tasks for Administering Data Service Resources (Continued)

Task	Instructions
Display resource type, resource group, and resource configuration information	“Displaying Resource Type, Resource Group, and Resource Configuration Information” on page 78
Change resource type, resource group, and resource properties	“How to Change Resource Type Properties” on page 79 “How to Change Resource Group Properties” on page 80 “How to Change Resource Properties” on page 81
Clear error flags for failed Resource Group Manager (RGM) processes	“How to Clear the STOP_FAILED Error Flag on Resources” on page 86
Clear the Start_failed resource state	“How to Clear a Start_failed Resource State by Switching Over a Resource Group” on page 88 “How to Clear a Start_failed Resource State by Restarting a Resource Group” on page 90 “How to Clear a Start_failed Resource State by Disabling and Enabling a Resource” on page 91
Reregister the built-in resource types LogicalHostname and SharedAddress	“How to Reregister Preregistered Resource Types After Inadvertent Deletion” on page 94
Update the network interface ID list for the network resources, and update the node list for the resource group	“Adding a Node to a Resource Group” on page 95
Remove a node from a resource group	“Removing a Node From a Resource Group” on page 98
Set up HAStoragePlus for resource groups to synchronize the startups between those resource groups and device groups	“How to Set Up the HAStoragePlus Resource Type for New Resources” on page 107 “How to Set Up the HAStoragePlus Resource Type for Existing Resources” on page 109 “How to Set Up the HAStoragePlus Resource for Cluster File Systems” on page 111 “How to Set Up the HAStoragePlus Resource Type by Using the clsetup Utility” on page 115
Set up the HAStoragePlus to make a local Solaris ZFS highly available	“How to Set Up the HAStoragePlus Resource Type to Make a Local Solaris ZFS File System Highly Available” on page 119
Modify online the resource for a highly available local file system	“Modifying Online the Resource for a Highly Available Local File System” on page 126
Change the cluster file system to local file system in an HAStoragePlus resource	“Changing the Cluster File System to a Local File System in an HAStoragePlus Resource” on page 137
Upgrade the built-in resource types LogicalHostname and SharedAddress	“Upgrading a Resource Type” on page 33

TABLE 2-1 Tasks for Administering Data Service Resources (Continued)

Task	Instructions
Upgrade the HAStoragePlus resource type	“Upgrading a Resource Type” on page 33
Distribute online resource groups among cluster nodes	“Distributing Online Resource Groups Among Cluster Nodes” on page 139
Distribute resource-group load across cluster nodes	“Configuring the Distribution of Resource Group Load Across Nodes” on page 148
Enable Solaris SMF services to run with Oracle Solaris Cluster	“Enabling Oracle Solaris SMF Services to Run With Oracle Solaris Cluster” on page 154
Tune fault monitors for Oracle Solaris Cluster data services	“Tuning Fault Monitors for Oracle Solaris Cluster Data Services” on page 163

Note – The procedures in this chapter describe how to use the Oracle Solaris Cluster maintenance commands to complete these tasks. Other tools also enable you to administer your resources. See [“Tools for Data Service Resource Administration” on page 21](#) for details about these options.

Configuring and Administering Oracle Solaris Cluster Data Services

Configuring an Oracle Solaris Cluster data service involves the following tasks.

- Registering a resource type
- Upgrading a resource type
- Creating resource groups
- Adding resources to resource groups
- Bringing resources online

Use the procedures in this chapter to update your data service configuration after the initial configuration. For example, to change resource type, resource group, and resource properties, go to [“Changing Resource Type, Resource Group, and Resource Properties” on page 78](#).

Registering a Resource Type

A resource type provides specification of common properties and callback methods that apply to all of the resources of the given type. You must register a resource type before you create a resource of that type. For details about resource types, see [Chapter 1, “Planning for Oracle Solaris Cluster Data Services.”](#)

An administrator can register a resource type for a zone cluster by specifying a resource type registration (RTR) file that resides inside the zone cluster. In other words, the file must be under the zone root path. The RTR file inside the zone cluster cannot have the `Global_zone` property set to `TRUE`. The RTR file inside the zone cluster cannot be of type `RTR_LOGICAL_HOSTNAME` or `RTR_SHARED_ADDRESS`.

Note – If you want to register a resource type in a zone cluster that uses the Trusted Extensions feature of Oracle Solaris, and you want to set the `Global_zone` resource-type property to `TRUE`, you must place the RTR file in the `/usr/cluster/lib/rgm/rtreg` directory of the global cluster.

The administrator can also register a resource type for a zone cluster from the location `/usr/cluster/lib/rgm/rtreg`. The administrator in the zone cluster cannot modify any RTR files in this directory. This enables registering system resource types for a zone cluster, even when the RTR file has one of the properties that cannot be set directly from the zone cluster. This process provides a secure way of delivering system resource types.

▼ How to Register a Resource Type

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the name for the resource type that you plan to register. The resource type name is an abbreviation for the data service name.

- 1 **On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**

- 2 **Register the resource type.**

```
# clresourcetype register resource-type
resource-type    Specifies name of the resource type to add.
```

- 3 **Verify that the resource type has been registered.**

```
# clresourcetype show
```

Example 2-1 Registering a Resource Type

The following example registers the `SUNW.oracle_server:8` resource type, which represents the HA Oracle Server application in an Oracle Solaris Cluster configuration.

```
# clresourcetype register SUNW.oracle_server:8
# clresourcetype show SUNW.oracle_server:8

Resource Type:                SUNW.oracle_server:8
RT_description:                Resource type for Oracle Server
RT_version:                    8
API_version:                   2
RT_basedir:                    /opt/SUNWscor/oracle_server
Single_instance:               False
Proxy:                          False
Init_nodes:                    All potential masters
Installed_nodes:               <All>
Failover:                       True
Pkglist:                       <NULL>
RT_system:                     False
Global_zone:                   False
```

Next Steps After registering resource types, you can create resource groups and add resources to the resource group. For details, see [“Creating a Resource Group” on page 42](#).

See Also The following man pages:

- [clresourcetype\(1CL\)](#)
- [clresourcegroup\(1CL\)](#)
- [clresource\(1CL\)](#)

Upgrading a Resource Type

Upgrading a resource type enables you to use new features that are introduced in the new version of the resource type. A new version of a resource type might differ from a previous version in the following ways.

- Default settings of resource type properties might change.
- New extension properties of the resource type might be introduced.
- Existing extension properties of the resource type might be withdrawn.
- The set of standard properties that are declared for the resource type might change.
- The attributes of resource properties such as `min`, `max`, `arraymin`, `arraymax`, `default`, and `tunability` might change.
- The set of declared methods might differ.
- The implementation of methods or the fault monitor might change.

Upgrading a resource type involves the tasks that are explained in the following sections:

1. “How to Install and Register an Upgrade of a Resource Type” on page 34
2. “How to Migrate Existing Resources to a New Version of the Resource Type” on page 35

▼ How to Install and Register an Upgrade of a Resource Type

The instructions that follow explain how to use the `clresource` command to perform this task. However, you are not restricted to using the `clresource` command for this task. Instead of the `clresource` command, you can use the Resource Group option of the `clsetup` command to perform this task. See [clsetup\(1CL\)](#) for more information.

Before You Begin Consult the documentation for the resource type to determine what you must do before installing the upgrade package on a node. One action from the following list will be required:

- You must reboot the node in noncluster mode.
- You may leave the node running in cluster mode, but you must turn off monitoring of all instances of the resource type.
- You may leave the node running in cluster mode and leave monitoring turned on for all instances of the resource type.

If you must reboot the node in noncluster mode, prevent a loss of service by performing a rolling upgrade. In a rolling upgrade, you install the package on each node individually while leaving the remaining nodes running in cluster mode.

- 1 **On a cluster member, assume the root role that provides `soLaris.cluster.modify` RBAC authorization.**
- 2 **Install the package for the resource type upgrade on all cluster nodes where instances of the resource type are to be brought online.**
- 3 **Register the new version of the resource type.**

To ensure that the correct version of the resource type is registered, you must specify the following information:

- The resource type name
- The resource type registration (RTR) file that defines the resource type

```
# clresourcetype register -f path-to-new-rtr-file resource-type-name
```

The format of the resource type name is as follows:

```
vendor-id.base-rt-name:rt-version
```

For an explanation of this format, see [“Format of Resource Type Names” on page 25](#).

4 Display the newly registered resource type.

```
# clresourcetype show resource-type-name
```

5 If necessary, set the Installed_nodes property to the nodes where the package for the resource type upgrade is installed.

You must perform this step if the package for the resource type upgrade is not installed on all cluster nodes.

The `nodeList` property of all resource groups that contain instances of the resource type must be a subset of the `Installed_nodes` property of the resource type.

```
# clresourcetype set -n installed-node-list resource-type
```

`-n installed-node-list` Specifies the names of nodes on which this resource type is installed.

▼ How to Migrate Existing Resources to a New Version of the Resource Type

The instructions that follow explain how to use the `clresource` command to perform this task. However, you are not restricted to using the `clresource` command for this task. Instead of the `clresource` command, you can use the Resource Group option of the `clsetup` command to perform this task. See [clsetup\(1CL\)](#) for more information.

Before You Begin Consult the instructions for upgrading the resource type to determine when you can migrate resources to a new version of the resource type.

- Any time
- Only when the resource is unmonitored
- Only when the resource is offline
- Only when the resource is disabled
- Only when the resource group is unmanaged

The instructions might state that you cannot upgrade your existing version of the resource. If you cannot migrate the resource, consider the following alternatives:

- Deleting the resource and replacing it with a new resource of the upgraded version
- Leaving the resource at the old version of the resource type

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 For each resource of the resource type that is to be migrated, change the state of the resource or its resource group to the appropriate state.

- If you can migrate the resource at any time, no action is required.
- If you can migrate the resource only when the resource is unmonitored, type the following command:

```
# clresource unmonitor resource
```

- If you can migrate the resource only when the resource is offline, type the following command:

```
# clresource disable resource
```

Note – If other resources depend on the resource that you are migrating, this step fails. In this situation, consult the error message that is printed to determine the names of the dependent resources. Then repeat this step, specifying a comma-separated list that contains the resource that you are migrating and any dependent resources.

- If you can migrate the resource only when the resource is disabled, type the following command:

```
# clresource disable resource
```

Note – If other resources depend on the resource that you are migrating, this step fails. In this situation, consult the error message that is printed to determine the names of the dependent resources. Then repeat this step, specifying a comma-separated list that contains the resource that you are migrating and any dependent resources.

- If you can migrate the resource only when the resource group is unmanaged, type the following commands:

```
# clresource disable -g resource-group +  
# clresourcegroup offline resource-group  
# clresourcegroup unmanage resource-group
```

The replaceable items in these commands are as follows:

resource-group Specifies the resource group that is to be unmanaged

3 For each resource of the resource type that is to be migrated, change the `Type_version` property to the new version.

If necessary, set other properties of the same resource to appropriate values in the same command. To set these properties, specify the `-p` option in the command.

To determine whether you are required to set other properties, consult the instructions for upgrading the resource type. You might be required to set other properties for the following reasons:

- An extension property has been introduced in the new version of the resource type.
- The default value of an existing property has been changed in the new version of the resource type.

```
# clresource set -p Type_version=new-version \
[-p extension-property=new-value] [-p standard-property=new-value] resource
```

Note – If the existing version of the resource type does not support upgrades to the new version, this step fails.

4 Restore the previous state of the resource or resource group by reversing the command that you typed in [Step 2](#).

- **If you can migrate the resource at any time, no action is required.**

Note – After migrating a resource that can be migrated at any time, the resource probe might not display the correct resource type version. In this situation, disable and re-enable the resource's fault monitor to ensure that the resource probe lists the correct resource type version.

- **If you can migrate the resource only when the resource is unmonitored, type the following command:**

```
# clresource monitor resource
```

- **If you can migrate the resource only when the resource is offline, type the following command:**

```
# clresource enable resource
```

Note – If you disabled in [Step 2](#) other resources that depend on the resource that you are migrating, enable the dependent resources also.

- **If you can migrate the resource only when the resource is disabled, type the following command:**

```
# clresource enable resource
```

Note – If you disabled in [Step 2](#) other resources that depend on the resource that you are migrating, enable the dependent resources also.

- **If you can migrate the resource only when the resource group is unmanaged, type the following commands:**

```
# clresource enable -g resource-group +
# clresourcegroup manage resource-group
# clresourcegroup online resource-group
```

Example 2–2 Migrating a Resource That Can Be Migrated Only When Offline

This example shows the migration of a resource that can be migrated only when the resource is offline. The new resource type package contains methods that are located in new paths. Because the methods are not overwritten during the installation, the resource does not need to be disabled until after the upgraded resource type is installed.

The characteristics of the resource in this example are as follows:

- The new resource type version is 2.0.
- The resource name is myresource.
- The resource type name is myrt.
- The new RTR file is in /opt/XYZmyrt/etc/XYZ.myrt.
- No dependencies on the resource that is to be migrated exist.
- The resource that is to be migrated can be taken offline while leaving the containing resource group online.

This example assumes that the upgrade package is already installed on all cluster nodes according to the supplier's directions.

```
# clresourcetype register -f /opt/XYZmyrt/etc/XYZ.myrt myrt
# clresource disable myresource
# clresource set -p Type_version=2.0 myresource
# clresource enable myresource
```

Example 2-3 Migrating a Resource That Can Be Migrated Only When Unmonitored

This example shows the migration of a resource that can be migrated only when the resource is unmonitored. The new resource type package contains only the monitor and RTR file. Because the monitor is overwritten during installation, monitoring of the resource must be disabled before the upgrade package is installed.

The characteristics of the resource in this example are as follows:

- The new resource type version is 2.0.
- The resource name is `myresource`.
- The resource type name is `myrt`.
- The new RTR file is in `/opt/XYZmyrt/etc/XYZ.myrt`.

The following operations are performed in this example.

1. Before the upgrade package is installed, the following command is run to disable monitoring of the resource:

```
# clresource unmonitor myresource
```

2. The upgrade package is installed on all cluster nodes according to the supplier's directions.

3. To register the new version of the resource type, the following command is run:

```
# clresourcetype register -f /opt/XYZmyrt/etc/XYZ.myrt myrt
```

4. To change the `Type_version` property to the new version, the following command is run:

```
# clresource set -p Type_version=2.0 myresource
```

5. To enable monitoring of the resource after its migration, the following command is run:

```
# clresource monitor myresource
```

▼ How to Unregister Older Unused Versions of the Resource Type

If you have migrated all existing resources to the newest version of the resource type and have no further need for the older version of that resource type, unregister the older versions.

- 1 On a cluster member, assume the root role that provides `soLaris.cluster.modify` RBAC authorization.

- 2 Retrieve a list of the registered resource type versions and unregister the version you no longer wish to use.

```
# clresourcetype list | grep myrt
```

```
XYZ.myrt:1.0  
XYZ.myrt:2.0
```

```
# clresourcetype unregister XYZ.myrt:1.0
```

- 3 If the resource type has the `RT_system` property set to `TRUE`, you must set it to `FALSE` before you can remove the resource type.

If the following text appears after you run `clresourcetype unregister` from [Step 2](#), you must set the property to `TRUE` by typing the commands that appear below the output below.

```
clrt: (C944871) Operation not allowed on system rescue type <XYZ.myrt:1.0>
```

```
# clresourcetype set -p RT_system=FALSE XYZ.myrt:1.0
```

```
# clresourcetype unregister XYZ.myrt:1.0
```

By default, the `RT_system` property by default is set to `TRUE` on the pre-installed resource types `LogicalHostname` and `SharedAddress`. If you have upgraded to a newer version of one of these pre-installed resource types, you will have to set `RT_system` to `FALSE` before unregistering the older version. In this example, all of the existing `LogicalHostname` resources have been migrated to version 4.0 of `LogicalHostname`.

```
# clresourcetype list
```

```
...  
SUNW.LogicalHostname:3  
SUNW.LogicalHostname:4  
...
```

```
# clresourcetype set -p RT_system=FALSE SUNW.LogicalHostname:3
```

```
# clresourcetype unregister SUNW.LogicalHostname:3
```

Downgrading a Resource Type

You can downgrade a resource to an older version of its resource type. The conditions for downgrading a resource to an older version of the resource type are more restrictive than the conditions for upgrading to a newer version of the resource type. The resource group that contains the resource must be unmanaged.

▼ How to Downgrade a Resource to an Older Version of Its Resource Type

The instructions that follow explain how to use the `clresource` command to perform this task. However, you are not restricted to using the `clresource` command for this task. Instead of the `clresource` command, you can use the Resource Group option of the `clsetup` command to perform this task. See [clsetup\(1CL\)](#) for more information.

- 1 **On a cluster member, assume the root role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**
- 2 **Switch offline the resource group that contains the resource that you are downgrading.**
`clresourcegroup offline resource-group`
- 3 **Disable all resources in the resource group that contains the resource that you are downgrading.**
`clresource disable -g resource-group +`
- 4 **Unmanage the resource group that contains the resource that you are downgrading.**
`clresourcegroup unmanage resource-group`
- 5 **If necessary, reregister the old version of the resource type to which you are downgrading.**
 Perform this step only if the version to which you are downgrading is no longer registered. If the version to which you are downgrading is still registered, omit this step.
`clresourcetype register resource-type-name`
- 6 **For the resource that you are downgrading, set the `Type_version` property to the old version to which you are downgrading.**
 If necessary, edit other properties of the same resource to appropriate values in the same command.
`clresource set -p Type_version=old-version resource-todowngrade`
- 7 **Enable all the resources that you disabled in [Step 3](#).**
`# clresource enable -g resource-group +`
- 8 **Bring to a managed state the resource group that contains the resource that you downgraded.**
`# clresourcegroup manage resource-group`
- 9 **Bring online the resource group that contains the resource that you downgraded.**
`# clresourcegroup online resource-group`

Creating a Resource Group

A resource group contains a set of resources, all of which are brought online or offline together on a given node or set of nodes. You must create an empty resource group before you place resources into it.

The two resource group types are **failover** and **scalable**. A failover resource group can be online on only one node at any time, while a scalable resource group can be online on multiple nodes simultaneously.

The following procedures explain how to use the `clresourcegroup` command to create a resource group. See the `clresourcegroup(1CL)` man page for more information.

For conceptual information about resource groups, see [Chapter 1, “Planning for Oracle Solaris Cluster Data Services,”](#) and *Oracle Solaris Cluster Concepts Guide*.

▼ How to Create a Failover Resource Group

A failover resource group contains the following types of resources:

- Network address resources, which are instances of the built-in resource types `LogicalHostname` and `SharedAddress`
- Failover resources, which are the data service application resources for a failover data service

The network address resources and their dependent data service resources move between cluster nodes when data services fail over or are switched over.

Note – Perform this procedure from any cluster node.

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Create the failover resource group.

```
# clresourcegroup create [-n nodelist] resource-group
```

-n nodelist Specifies a comma-separated, ordered list of nodes that can master this resource group.

This list is optional. If you omit this list, the resource group is created on all nodes in the cluster.

resource-group Specifies your choice of the name of the failover resource group to add. This name must begin with an ASCII character.

3 Verify that the resource group has been created.

```
# clresourcegroup show resource-group
```

Example 2-4 Creating a Failover Resource Group

This example shows the creation of the failover resource group `resource-group-1`. The cluster nodes `phys-schost-1` and `phys-schost-2` can master this resource group.

```
# clresourcegroup create -n phys-schost1,phys-schost-2 resource-group-1
# clresourcegroup show -v resource-group-1

=== Resource Groups and Resources ===

Resource Group:                resource-group1
RG_description:                <NULL>
RG_mode:                       Failover
RG_state:                      Unmanaged
RG_project_name:              default
RG_affinities:                <NULL>
RG_SLM_type:                  manual
Auto_start_on_new_cluster:    True
Failback:                     False
Nodelist:                     phys-schost-1 phys-schost-2
Maximum primaries:            1
Desired primaries:            1
RG_dependencies:              <NULL>
Implicit_network_dependencies: True
Global_resources_used:        <All>
Pingpong_interval:            3600
Pathprefix:                   <NULL>
RG_System:                    False
Suspend_automatic_recovery:   False
```

Next Steps After you create a failover resource group, you can add application resources to this resource group. See “[Tools for Adding Resources to Resource Groups](#)” on page 48 for the procedure.

See Also The `clresourcegroup(1CL)` man page.

▼ How to Create a Scalable Resource Group

A scalable resource group is used with scalable services. The shared address feature is the Oracle Solaris Cluster networking facility that enables the multiple instances of a scalable service to appear as a single service. You must first create a failover resource group that contains the shared addresses on which the scalable resources depend. Next, create a scalable resource group, and add scalable resources to that group. Each instance of the scalable service must run on a different cluster node.

Note – Perform this procedure from any cluster node.

- 1 On a cluster member assume the root role that provides solaris.cluster.modify RBAC authorization.**
- 2 Create the failover resource group that holds the shared addresses that the scalable resource is to use.**
- 3 Create the scalable resource group.**

```
# clresourcegroup create -S [-p Maximum primaries=m] [-p Desired primaries=n] \
[-n nodelist] resource-group
```

-S Specifies that the resource group is to be multi-mastered. If the *-p Maximum primaries* and *-p Desired primaries* options are omitted, both properties are set to the number of nodes in the resource group's node list.

*-p Maximum primaries=*m** Specifies the maximum number of active primaries for this resource group.

*-p Desired primaries=*n** Specifies the number of active primaries on which the resource group should attempt to start.

-n nodelist Specifies a comma-separated, ordered list of nodes in which this resource group is to be available.

This list is optional. If you omit this list, the resource group is created on all nodes in the cluster.

resource-group Specifies your choice of the name of the scalable resource group to add. This name must begin with an ASCII character.

- 4 Verify that the scalable resource group has been created.**

```
# clresourcegroup show resource-group
```

Example 2-5 Creating a Scalable Resource Group

This example shows the creation of the scalable resource group *resource-group-1*. This resource group is to be hosted in the global cluster of nodes *phys-schost-1* and *phys-schost-2*. The scalable resource group depends on the failover resource group *resource-group-2*, which contains the shared address resources.

```
# clresourcegroup create -S \
-p Maximum primaries=2 \
-p Desired primaries=2 \
-p RG_dependencies=resource-group-2 \
```

```

-n phys-schost-1, phys-schost-2 \
resource-group-1

# clresourcegroup show resource-group-1

=== Resource Groups and Resources ===

Resource Group:                resource-group-1
RG_description:                 <NULL>
RG_mode:                        Scalable
RG_state:                       Unmanaged
RG_project_name:                default
RG_affinities:                  <NULL>
Auto_start_on_new_cluster:      True
Failback:                       False
Nodelist:                       phys-schost-1 phys-schost-2
Maximum primaries:              2
Desired primaries:              2
RG_dependencies:                resource-group2
Implicit_network_dependencies:   True
Global_resources_used:          <All>
Pingpong_interval:              3600
Pathprefix:                     <NULL>
RG_System:                      False
Suspend_automatic_recovery:     False

```

Next Steps After you have created a scalable resource group, you can add scalable application resources to the resource group. See [“How to Add a Scalable Application Resource to a Resource Group” on page 59](#) for details.

See Also The `clresourcegroup(1CL)` man page.

Configuring Failover and Scalable Data Services on Shared File Systems

After the NAS device is installed and configured, you can use the `ScaLMountPoint` resource to configure failover and scalable applications.

An instance of the `ScaLMountPoint` resource type represents the mount point of one of the following type of file systems:

- File systems on a network-attached storage (NAS) device

The NAS device and the file system must already be configured for use with Oracle Solaris Cluster.

The `ScaLMountPoint` resource type is a scalable resource type. An instance of this resource type is online on each node in the node list of the resource group that contains the resource.

▼ How to Configure a Failover Application Using the ScalMountPoint Resource

Before You Begin This procedure provides the long forms of the Oracle Solaris Cluster commands. Most commands also have short forms. Except for the forms of the command names, the commands are identical.

To perform this procedure, assume the root role that provides `solaris.cluster.read` and `solaris.cluster.modify` RBAC (role-based access control) authorization.

- 1 **Create a scalable resource group containing the ScalMountPoint resource for the NAS NFS file system.**

```
# clrg create -p RG_mode=Scalable \
-p Desired_primaries=num_active_primary \
-p Maximum_primaries=max_num_active_primary scalmp-rg
```

```
# clrt register SUNW.ScalMountPoint
```

```
# clrs create -g scalmp-rg -t SUNW.ScalMountPoint \
-p TargetFileSystem=nas_device:path \
-p FileSystemType=nas \
-p MountPointDir=fs_mountpoint scalmp-rs
```

```
# clrg online -eM scalmp-rg
```

- 2 **Create a failover resource group that contains the failover application resource.**

```
# clrg create -p rg_affinities=++scalmp-rg app-fo-rg
```

The failover application resource group must have a strong positive affinity upon the resource group created in [Step 1](#).

```
# clrt register app_resource_type
```

```
# clrs create -g app-fo-rg -t app_resource_type \
-p Resource_dependencies_offline_restart=scalmp-rs \
...
app-fo-rs
```

The failover application resource must have an offline restart dependency upon the ScalMountPoint resource created in [Step 1](#).

```
# clrg online -eM app-fo-rg
```

▼ How to Configure a Scalable Application Using the ScalMountPoint Resource

Before You Begin This procedure provides the long forms of the Oracle Solaris Cluster commands. Most commands also have short forms. Except for the forms of the command names, the commands are identical.

To perform this procedure, assume the root role that provides `solaris.cluster.read` and `solaris.cluster.modify` RBAC authorization.

- 1 Create a scalable resource group containing the ScalMountPoint resource for the NAS NFS file system.

```
# clrg create -p RG_mode=Scalable \
-p Desired primaries=num_active_primary \
-p Maximum primaries=max_num_active_primary scalmp-rg

# clrt register SUNW.ScalMountPoint

# clrs create -g scalmp-rg -t SUNW.ScalMountPoint \
-p TargetFileSystem=nas_device:path \
-p FileSystemType=nas \
-p MountPointDir=fs_mountpoint scalmp-rs

# clrg online -eM scalmp-rg
```

- 2 Create a scalable resource group that contains the application resource.

```
# clrg create -p RG_mode=Scalable \
-p Maximum primaries=max_num_active_primary \
-p Desired primaries=num_active_primary \
-p rg_affinities=++scalmp-rg app-rg
```

The application resource group must have a strong positive affinity upon the resource group created in [Step 1](#).

```
# clrt register app_resource_type

# clrs create -g app-rg -t app_resource_type \
...
-p Scalable=True \
-p resource_dependencies_offline_restart=scalmp-rs app-rs

# clrg online -eM app-rg
```

The application resource must have an offline restart dependency upon the ScalMountPoint resources created in [Step 1](#).

Tools for Adding Resources to Resource Groups

A resource is an instantiation of a resource type. You must add resources to a resource group before the RGM can manage the resources. This section describes the following three resource types.

- Logical hostname resources
- Shared-address resources
- Data service (application) resources

Oracle Solaris Cluster provides the following tools for adding resources to resource groups:

- The `clsetup` utility. See the `clsetup(1CL)` man page for more information.
- Oracle Solaris Cluster maintenance commands.

You can use the `clsetup` utility or the Oracle Solaris Cluster maintenance commands to add the logical hostname resources and shared-address resources to the resource group.

The `clsetup` utility enables you to add resources to the resource group interactively. Configuring these resources interactively reduces the possibility for configuration errors that might result from command syntax errors or omissions. The `clsetup` utility ensure that all required resources are created and that all required dependencies between resources are set.

Always add logical hostname resources and shared address resources to failover resource groups. Add data service resources for failover data services to failover resource groups. Failover resource groups contain both the logical hostname resources and the application resources for the data service. Scalable resource groups contain only the application resources for scalable services. The shared address resources on which the scalable service depends must reside in a separate failover resource group. You must specify dependencies between the scalable application resources and the shared address resources for the data service to scale across cluster nodes.

Note – The DEPRECATED flag marks the logical hostname or shared address resource as a deprecated address. These addresses are not suitable for outbound requests since they can migrate to a different cluster node due to a failover or switchover.

For more information about resources, see *Oracle Solaris Cluster Concepts Guide* and Chapter 1, “Planning for Oracle Solaris Cluster Data Services.”

▼ How to Add a Logical Hostname Resource to a Resource Group by Using the `clsetup` Utility

The following instructions explain how to add a logical hostname resource to a resource group by using the `clsetup` utility. Perform this procedure from one node only.

This procedure provides the long forms of the Oracle Solaris Cluster maintenance commands. Most commands also have short forms. Except for the forms of the command names, the commands are identical.

Before You Begin Ensure that the following prerequisites are met:

- An entry for each logical hostname that is to be made available by the resource is added to the name service database.
- If you are using IP Networking Multipathing (IPMP) groups, the groups are configured on the nodes where the logical hostname resource can be brought online.

Ensure that you have the following information:

- The hostnames that you plan to add to the resource group

1 Assume the root role on any cluster node.

2 Start the `clsetup` utility.

```
# clsetup
```

The `clsetup` main menu is displayed.

3 Type the number for the option for data services and press Return.

The Data Services menu is displayed.

4 Type the number for the option for configuring the Logical Hostname resource and press Return.

The `clsetup` utility provides the list of prerequisites for performing this task.

5 Verify that the prerequisites are met, and press Return to continue.

The `clsetup` utility provides a list of the cluster nodes where the logical hostname resource can be brought online.

6 Select the nodes where the logical hostname resource can be brought online.

- To accept the default selection of all listed nodes in an arbitrary order, type `a` and press Return.

- **To select a subset of the listed nodes, type a comma-separated or space-separated list of the numbers that correspond to the nodes and press Return.**
 - **To select all nodes in a particular order, type a comma-separated or space-separated ordered list of the numbers that correspond to the nodes and press Return.**

Ensure that the nodes are listed in the order in which the nodes are to appear in the logical hostname resource group's node list. The first node in the list is the primary node of this resource group.
- 7 To confirm your selection of nodes, type `d` and press Return.**

The `clsetup` utility provides a screen where you can specify the logical hostname that the resource is to make available.
 - 8 Type the logical hostname that this resource is to make available and press Return.**

The `clsetup` utility lists the names of the Oracle Solaris Cluster objects that the utility will create.
 - 9 If you require a different name for any Oracle Solaris Cluster object, change the name as follows.**
 - a. Type the number for the name that you are changing and press Return.**

The `clsetup` utility provides a screen where you can specify the new name.
 - b. At the New Value prompt, type the new name and press Return.**

The `clsetup` utility returns you to the list of the names of the Oracle Solaris Cluster objects that the utility will create.
 - 10 To confirm your selection of Oracle Solaris Cluster object names, type `d` and press Return.**

The `clsetup` utility provides information about the Oracle Solaris Cluster configuration that the utility will create.
 - 11 To create the configuration, type `c` and press Return.**

The `clsetup` utility provides a progress message to indicate that the utility is running commands to create the configuration. When configuration is complete, the `clsetup` utility lists the commands that the utility ran to create the configuration.
 - 12 (Optional) Type `q` and press Return repeatedly until you quit the `clsetup` utility.**

If you prefer, you can leave the `clsetup` utility running while you perform other required tasks before using the utility again. If you choose to quit `clsetup`, the utility recognizes your existing logical hostname resource group when you restart the utility.

13 Verify that the logical hostname resource has been created.

Use the `clresource(1CL)` utility for this purpose. By default, the `clsetup` utility assigns the name `node_name-rg` to the resource group.

```
# clresource show node_name-rg
```

▼ How to Add a Logical Hostname Resource to a Resource Group Using the Command-Line Interface

Note – When you add a logical hostname resource to a resource group, the extension properties of the resource are set to their default values. To specify a nondefault value, you must modify the resource after you add the resource to a resource group. For more information, see [“How to Modify a Logical Hostname Resource or a Shared Address Resource” on page 85](#).

Perform this procedure from any cluster node.

Before You Begin

- Ensure that you have the following information.
 - The name of the failover resource group to which you are adding the resource
 - The hostnames that you plan to add to the resource group
- Ensure that the `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Add the logical hostname resource to the resource group.

```
# clreslogicalhostname create -g resource-group -h hostnamelist, ... [-N netiflist] resource
-g resource-group    Specifies the name of the resource group in which this resource resides.
-h hostnamelist, ... Specifies a comma-separated list of UNIX hostnames (logical
                    hostnames) by which clients communicate with services in the resource
                    group.
```

You must specify the fully qualified name with the `-h` option if you require a fully qualified hostname.

```
-N netiflist        Specifies an optional, comma-separated list that identifies the IPMP
                    groups that are on each node. Each element in netiflist must be in the
                    form of netif@node. netif can be given as an IPMP group name, such
                    as sc_ipmp0. The node can be identified by the node name or node ID,
                    such as sc_ipmp0@1 or sc_ipmp@phys-schost-1.
```

Note – Oracle Solaris Cluster does not support the use of the adapter name for `netif`.

resource Specifies an optional resource name of your choice. You cannot use the fully qualified name in the resource name.

3 Verify that the logical hostname resource has been added.

```
# clresource show resource
```

Example 2-6 Adding a Logical Hostname Resource to a Resource Group

This example shows the addition of logical hostname resource (`resource-1`) to a resource group (`resource-group-1`).

```
# clreslogicalhostname create -g resource-group-1 -h schost-1 resource-1
# clresource show resource-1
```

```
=== Resources ===
```

```
Resource:                resource-1
Type:                    SUNW.LogicalHostname:2
Type_version:           2
Group:                   resource-group-1
R_description:
Resource_project_name:   default
Enabled{phys-schost-1}:  True
Enabled{phys-schost-2}:  True
Monitored{phys-schost-1}: True
Monitored{phys-schost-2}: True
```

Example 2-7 Adding Logical Hostname Resources That Identify IPMP Groups

This example shows the addition of the following logical host name resources to the resource group `nfs-fo-rg`:

- A resource that is named `cs23-rs`, which identifies the IPMP group `sc_ipmp0` on node 1 and node 2
- A resource that is named `cs24-rs`, which identifies the IPMP group `sc_ipmp1` on node 1 and node 2

```
# clreslogicalhostname create -g nfs-fo-rg -h cs23-rs -N sc_ipmp0@1,sc_ipmp0@2 cs23-rs
# clreslogicalhostname create -g nfs-fo-rg -h cs24-rs -N sc_ipmp1@1,sc_ipmp1@2 cs24-rs
```

Next Steps After you add logical hostname resources, see [“How to Bring Resource Groups Online” on page 62](#) to bring the resources online.

Troubleshooting Adding a resource causes the Oracle Solaris Cluster software to validate the resource. If the validation fails, the `clreslogicalhostname` command prints an error message and exits. To determine why the validation failed, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `clreslogicalhostname` command.

See Also The `clreslogicalhostname(1CL)` man page.

▼ How to Add a Shared Address Resource to a Resource Group by Using the `clsetup` Utility

The following instructions explain how to add a shared address resource to a resource group by using the `clsetup` utility. Perform this procedure from any cluster node.

This procedure provides the long forms of the Oracle Solaris Cluster maintenance commands. Most commands also have short forms. Except for the forms of the command names, the commands are identical.

Before You Begin Ensure that the following prerequisites are met:

- The shared address that is to be made available by the resource has an entry in a name service database.
- If you are using IP Networking Multipathing (IPMP) groups, the groups are configured on the nodes where the shared address resource can be brought online.

Ensure that you have the following information:

- The hostnames that you plan to add to the resource group.

1 Assume the root role on any cluster node.

2 Start the `clsetup` utility.

```
# clsetup
```

The `clsetup` main menu is displayed.

3 Type the number for the option for data services and press Return.

The Data Services menu is displayed.

4 Type the number for the option for configuring the shared address resource and press Return.

The `clsetup` utility provides the list of prerequisites for performing this task.

5 Verify that the prerequisites are met, and press Return to continue.

The `clsetup` utility lists the cluster nodes where the shared address resource can be brought online.

6 Select the nodes where the shared address resource can be brought online.

- To accept the default selection of all listed nodes in an arbitrary order, type `a` and press Return.
- To select a subset of the listed nodes, type a comma-separated or space-separated list of the numbers that correspond to the nodes and press Return.
- To select all nodes in a particular order, type a comma-separated or space-separated ordered list of the numbers that correspond to the nodes and press Return.

7 To confirm your selection of nodes, type `d` and press Return.

The `clsetup` utility provides a screen where you can specify the shared address that the resource is to make available.

8 Type the shared address that this resource is to make available and press Return.

The `clsetup` utility lists the names of the Oracle Solaris Cluster objects that the utility will create.

9 If you require a different name for any Oracle Solaris Cluster object, change the name as follows.

a. Type the number for the name that you are changing and press Return.

The `clsetup` utility provides a screen where you can specify the new name.

b. At the New Value prompt, type the new name and press Return.

The `clsetup` utility returns you to the list of the names of the Oracle Solaris Cluster objects that the utility will create.

10 To confirm your selection of Oracle Solaris Cluster object names, type `d` and press Return.

The `clsetup` utility provides information about the Oracle Solaris Cluster configuration that the utility will create.

11 To create the configuration, type `c` and Press Return.

The `clsetup` utility provides a progress message to indicate that the utility is running commands to create the configuration. When configuration is complete, the `clsetup` utility lists the commands that the utility ran to create the configuration.

12 (Optional) Type `q` and press Return repeatedly until you quit the `clsetup` utility.

If you prefer, you can leave the `clsetup` utility running while you perform other required tasks before using the utility again. If you choose to quit `clsetup`, the utility recognizes your existing shared address resource group when you restart the utility.

13 Verify that the shared address resource has been created.

Use the `clresource(1CL)` utility for this purpose. By default, the `clsetup` utility assigns the name `node_name-rg` to the resource group.

```
# clresource show node_name-rg
```

▼ How to Add a Shared Address Resource to a Resource Group Using the Command-Line Interface

Note – When you add a shared address resource to a resource group, the extension properties of the resource are set to their default values. To specify a non–default value, you must modify the resource after you add the resource to a resource group. For more information, see [“How to Modify a Logical Hostname Resource or a Shared Address Resource” on page 85](#).

Perform this procedure from any cluster node.

Before You Begin

- Ensure that you have the following information.
 - The name of the resource group into which you are adding the resource. This group must be a failover resource group that you created previously.
 - The hostnames that you plan to add to the resource group.
- Ensure that the `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**2 Add the shared address resource to the resource group.**

```
# clressharedaddress create -g resource-group -h hostnamelist, ... \
[-X auxnodelist] [-N netiflist] resource
```

`-g resource-group` Specifies the resource group name.

`-h hostnamelist, ...` Specifies a comma-separated list of shared address hostnames.

`-X auxnodelist` Specifies a comma-separated list of node names or IDs that identify the cluster nodes that can host the shared address but never serve as primary if failover occurs. These nodes are mutually exclusive, with the

nodes identified as potential masters in the resource group's node list. If no auxiliary node list is explicitly specified, the list defaults to the list of all cluster node names that are not included in the node list of the resource group that contains the shared address resource.

-N *netiflist*

Specifies an optional, comma-separated list that identifies the IPMP groups that are on each node. Each element in *netiflist* must be in the form of *netif@node*. *netif* can be given as an IPMP group name, such as *sc_ipmp0*. The node can be identified by the node name or node ID, such as *sc_ipmp0@1* or *sc_ipmp@phys-schost-1*.

Note – Oracle Solaris Cluster does not support the use of the adapter name for *netif*.

resource

Specifies an optional resource name of your choice.

3 Verify that the shared address resource has been added and validated.

```
# clresource show resource
```

Example 2–8 Adding a Shared Address Resource to a Resource Group

This example shows the addition of a shared address resource (*resource-1*) to a resource group (*resource-group-1*).

```
# clressharedaddress create -g resource-group-1 -h schost-1 resource-1
# clresource show resource-1
```

```
=== Resources ===
```

```
Resource:                resource-1
Type:                    SUNW.SharedAddress:2
Type_version:           2
Group:                   resource-group-1
R_description:
Resource_project_name:   default
Enabled{phys-schost-1}:  False
Enabled{phys-schost-2}:  False
Monitored{phys-schost-1}: True
Monitored{phys-schost-2}: True
```

Next Steps After you add a shared address resource, use the procedure [“How to Bring Resource Groups Online” on page 62](#) to enable the resource.

Troubleshooting Adding a resource causes the Oracle Solaris Cluster software to validate the resource. If the validation fails, the `clressharedaddress` command prints an error message and exits. To

determine why the validation failed, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `clressharedress(1CL)` man page.

See Also The `clressharedress(1CL)` man page.

▼ How to Add a Failover Application Resource to a Resource Group

A failover application resource is an application resource that uses logical hostnames that you previously created in a failover resource group.

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the following information.

- The name of the failover resource group to which you are adding the resource
- The name of the resource type for the resource
- The logical hostname resources that the application resource uses, which are the logical hostnames that you previously included in the same resource group

Note – This procedure also applies to proxy resources.

- 1 **On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**
- 2 **Add a failover application resource to the resource group.**

```
# clresource create -g resource-group -t resource-type \
[-p "extension-property[{node-specifier}]"=value, ...] [-p standard-property=value, ...] resource
-g resource-group
    Specifies the name of a failover resource group. This resource group must already exist.
-t resource-type
    Specifies the name of the resource type for the resource.
-p "extension-property[{node-specifier}]"=value, ...
    Specifies a comma-separated list of extension properties that you are setting for the resource.
    The extension properties that you can set depend on the resource type. To determine which
    extension properties to set, see the documentation for the resource type.
```

The *node-specifier* is an *optional* qualifier to the `-p` and `-x` options. This qualifier indicates that the extension property or properties on *only* the specified node or nodes are to be set when the resource is created. The specified extension properties on other nodes in the cluster are not set. If you do not include *node-specifier*, the specified extension properties on all nodes in the cluster are set. You can specify a node name or a node identifier for *node-specifier*. Examples of the syntax of *node-specifier* include the following:

```
-p "myprop{phys-schost-1}"
```

The braces (`{ }`) indicate that you are setting the specified extension property on only node `phys-schost-1`. For most shells, the double quotation marks (`"`) are required.

```
-p standard-property=value, ...
```

Specifies a comma-separated list of standard properties that you are setting for the resource.

The standard properties that you can set depend on the resource type. To determine which standard properties to set, see the following man pages: [rt_properties\(5\)](#), [cluster\(1CL\)](#), [rg_properties\(5\)](#), [r_properties\(5\)](#), and [property_attributes\(5\)](#).

resource

Specifies your choice of the name of the resource to add.

The resource is created in the enabled state.

3 Verify that the failover application resource has been added and validated.

```
# clresource show resource
```

Example 2-9 Adding a Failover Application Resource to a Resource Group

This example shows the addition of a resource (`resource-1`) to a resource group (`resource-group-1`). The resource depends on logical hostname resources (`schost-1`, `schost-2`), which must reside in the same failover resource groups that you defined previously.

```
# clresource create -g resource-group-1 -t resource-type-1 \  
-p Resource_dependencies=schost-1,schost2 resource-1 \  
# clresource show resource-1
```

```
=== Resources ===
```

Resource:	resource-1
Type:	resource-type-1
Type_version:	
Group:	resource-group-1
R_description:	
Resource_project_name:	default
Enabled{phys-schost-1}:	False
Enabled{phys-schost-2}:	False
Monitored{phys-schost-1}:	True
Monitored{phys-schost-2}:	True

Next Steps After you add a failover application resource, use the procedure “[How to Bring Resource Groups Online](#)” on page 62 to enable the resource.

Troubleshooting Adding a resource causes the Oracle Solaris Cluster software to validate the resource. If the validation fails, the `clresource` command prints an error message and exits. To determine why the validation failed, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `clresource` command.

See Also The `clresource(1CL)` man page.

▼ How to Add a Scalable Application Resource to a Resource Group

A scalable application resource is an application resource that uses the network load balancing features of Oracle Solaris Cluster software. The scalable application resource is in a multi-mastered resource group and has a dependency on one or more shared-address resources. The shared-address resources are in a failover resource group.

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the following information.

- The name of the scalable resource group to which you are adding the resource
- The name of the resource type for the resource
- The shared address resources that the scalable service resource uses, which are the shared addresses that you previously included in a failover resource group

Note – This procedure also applies to proxy resources.

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Add a scalable application resource to the resource group.

```
# clresource create -S -g resource-group -t resource-type \
-p Resource_dependencies=network-resource[,network-resource...] \
-p Scalable=True
[-p "extension-property[{node-specifier}]=value, ...] [-p standard-property=value, ...] resource
-S
```

Specifies that the resource group is to be multi-mastered. If the `-p Maximum primaries` and `-p Desired primaries` options are omitted, both properties are set to the number of nodes in the resource group's node list.

`-g resource-group`

Specifies the name of a scalable service resource group that you previously created.

- t *resource-type*
Specifies the name of the resource type for this resource.
- p Resource_dependencies= *network-resource*[,*network-resource*...]
Specifies the list of network resources (shared addresses) on which this resource depends.
- p Scalable=True
Specifies that this resource uses the network load balancing feature of Oracle Solaris Cluster software.
- p "*extension-property*{*node-specifier*}"=*value*, ...
Specifies a comma-separated list of extension properties that you are setting for the resource. The extension properties that you can set depend on the resource type. To determine which extension properties to set, see the documentation for the resource type.

The *node-specifier* is an *optional* qualifier to the -p and -x options. This qualifier indicates that the extension property or properties on *only* the specified node or nodes are to be set when the resource is created. The specified extension properties on other nodes in the cluster are not set. If you do not include *node-specifier*, the specified extension properties on all nodes in the cluster are set. You can specify a node name or a node identifier for *node-specifier*. Examples of the syntax of *node-specifier* include the following:

-p "myprop{phys-schost-1}"

The braces ({}) indicate that you are setting the specified extension property on only node *phys-schost-1*. For most shells, the double quotation marks (") are required.

- p *standard-property*=*value*, ...
Specifies a comma-separated list of standard properties that you are setting for the resource. The standard properties that you can set depend on the resource type. For scalable services, you typically set the *Port_list*, *Load_balancing_weights*, and *Load_balancing_policy* properties. To determine which standard properties to set, see the following man pages: [cluster\(1CL\)](#), [rt_properties\(5\)](#), [rg_properties\(5\)](#), [r_properties\(5\)](#), and [property_attributes\(5\)](#).

resource

Specifies your choice of the name of the resource to add.

The resource is created in the enabled state.

3 Verify that the scalable application resource has been added and validated.

```
# clresource show resource
```

Example 2–10 Adding a Scalable Application Resource to a Resource Group

This example shows the addition of a resource (*resource-1*) to a resource group (*resource-group-1*). Note that *resource-group-1* depends on the failover resource group that contains the network addresses that are in use (*schost-1* and *schost-2* in the following

example). The resource depends on shared address resources (schost-1, schost-2), which must reside in one or more failover resource groups that you defined previously.

```
# clresource create -S -g resource-group-1 -t resource-type-1 \
-p Resource_dependencies=schost-1,schost-2 resource-1 \
-p Scalable=True
# clresource show resource-1

=== Resources ===

Resource:                resource-1
Type:                    resource-type-1
Type_version:
Group:                   resource-group-1
R_description:
Resource_project_name:   default
Enabled{phys-schost-1}:  False
Enabled{phys-schost-2}:  False
Monitored{phys-schost-1}: True
Monitored{phys-schost-2}: True
```

Next Steps After you add a scalable application resource, follow the procedure “[How to Bring Resource Groups Online](#)” on page 62 to enable the resource.

Troubleshooting Adding a resource causes the Oracle Solaris Cluster software to validate the resource. If the validation fails, the `clresource` command prints an error message and exits. To determine why the validation failed, check the `sys log` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `clresource` command.

See Also The `clresource(1CL)` man page.

Bringing Resource Groups Online

To enable resources to begin providing HA services, you must perform the following operations:

- Bring resource groups online
- Enable the resources in their resource groups
- Enable the resource monitors
- Make the resource groups managed

You can perform these tasks individually or by using a single command.

After you bring a resource group online, it is configured and ready for use. If a resource or node fails, the RGM switches the resource group online on alternate nodes to maintain availability of the resource group.

▼ How to Bring Resource Groups Online

Perform this task from any cluster node.

- 1 On a cluster member, assume the root role that provides `soLaris.cluster.admin` RBAC authorization.

- 2 Type the command to bring online the resource groups.

- If you have intentionally disabled a resource or a fault monitor that must remain disabled, type the following command:

```
# clresourcegroup online -eM rg-list
```

rg-list Specifies a comma-separated list of the names of the resource groups to bring online. The resource groups must exist. The list may contain one resource group name or more than one resource group name.

- If you require the resources and their fault monitors to be enabled when the resource groups are brought online, type the following command:

```
# clresourcegroup online -eM rg-list
```

rg-list Specifies a comma-separated list of the names of the resource groups to bring online. The resource groups must exist. The list can contain one resource group name or more than one resource group name.

Note – If any resource group that you are bringing online declares a strong affinity for other resource groups, this operation might fail. For more information, see [“Distributing Online Resource Groups Among Cluster Nodes”](#) on page 139.

- 3 Verify that each resource group that you specified in [Step 2](#) is online.

The output from this command indicates on which nodes each resource group is online.

```
# clresourcegroup status
```

Example 2–11 Bringing Online a Resource Group

This example shows how to bring online the resource group `resource-group-1` and verify its status. All resources in this resource and their fault monitors are also enabled.

```
# clresourcegroup online -eM resource-group-1
# clresourcegroup status
```

Next Steps If you brought resource groups online *without* enabling their resources and fault monitors, enable the fault monitors of any resources that you require to be enabled. For more information, see “[How to Enable a Resource Fault Monitor](#)” on page 69.

See Also The `clresourcegroup(1CL)` man page.

Switching Resource Groups to Preferred Primaries

The `clresourcegroup remaster` command can switch the resource group offline from current primaries in order to switch the resource group online on preferred nodes. Similar to the `clresourcegroup online` command, the RGM computes the most preferred set of nodes based on the resource group's `Nodelist` property, but also takes into account the following factors:

- The `RG_affinities` property settings
- The `Load_factors` property settings, compared against node load limits and current loads
- The failure history of each resource group

▼ How to Switch Resource Groups to Preferred Primaries

Perform this task from any cluster node.

- 1 On a cluster member, assume the root role that provides `solaris.cluster.admin` RBAC authorization.
- 2 Type the command to switch the resource groups to their preferred primaries.
 - If you have intentionally disabled a resource or a fault monitor that must remain disabled, type the following command:


```
# clresourcegroup remaster rg-list
```

rg-list Specifies a comma-separated list of the names of the resource groups to switch to the preferred primaries. The resource groups must exist. The list may contain one resource group name or more than one resource group name.
 - If you require the resources and their fault monitors to be enabled when the resource groups are switched to the preferred primary, type the following command:


```
# clresourcegroup remaster -emM rg-list
```

rg-list Specifies a comma-separated list of the names of the resource groups to switch to the preferred primaries. The resource groups must exist. The list can contain one resource group name or more than one resource group name.

- 3 **Verify that each resource group that you specified in Step 2 has been switched to the preferred primary.**

The output from this command indicates the new nodes for the resource groups that were switched.

```
# clresourcegroup status
```

Enabling a Resource

You can enable a resource that you neglected to enable when you brought online a resource group.

▼ How to Enable a Resource

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have created and have the name of the resource that you intend to enable.

- 1 **On a cluster member, assume the root role that provides `solaris.cluster.admin` RBAC authorization.**
- 2 **Enable the resource.**

```
# clresource enable [-n nodelist] resource
```

-n nodelist Specifies a comma-separated, ordered list of nodes on which to enable the resource.

This list is optional. If you omit this list, the resource is enabled on all nodes in its resource group's node list.

Note – If you specify more than one node with the *-n* option, you can specify only one resource.

resource Specifies the name of the resource that you want to enable.

- 3 **Verify that the resource has been enabled.**

```
# clresource status
```

The output from this command indicates the state of the resource that you have enabled.

See Also The `clresource(1CL)` man page.

Quiescing Resource Groups

To stop a resource group from continuously switching from one node to another when a START or STOP method fails, bring it to a quiescent state. To bring a resource group to a quiescent state, you issue the `clresourcegroup quiesce` command.

When you quiesce a resource group, resource methods that are executing are allowed to run until they are completed. If a serious problem occurs, you might need to quiesce a resource group immediately. To do so, you specify the `-k` command option, which kills the following methods:

- `Prenet_start`
- `Start`
- `Monitor_start`
- `Monitor_stop`
- `Stop`
- `Postnet_stop`

Note – The `Init`, `Finis Boot`, and `Update` methods are not killed when you specify this command option.

However, if you immediately quiesce a resource group by killing methods, you might leave one of its resources in an error state such as `Start_failed` or `Stop_failed`. You must clear these error states yourself.

▼ How to Quiesce a Resource Group

1 Assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Quiesce the resource group.

```
# clresourcegroup quiesce resource-group
```

▼ How to Quiesce a Resource Group Immediately

- 1 Assume the root role that provides `soLaris.cLuster.modify` RBAC authorization.
- 2 Immediately quiesce the resource group.

```
# clresourcegroup quiesce -k resource-group
```

The `Prenet_start`, `Start`, `Monitor_start`, `Monitor_stop`, `Stop`, and `Postnet_stop` methods that are associated with the resource group are killed immediately. The resource group is brought to a quiescent state.

The `clresourcegroup quiesce -k` command blocks until the specified resource group has reached a quiescent state.

Suspending and Resuming the Automatic Recovery Actions of Resource Groups

You can temporarily suspend the automatic recovery actions of a resource group. You might need to suspend the automatic recovery of a resource group to investigate and fix a problem in the cluster. Or, you might need to perform maintenance on resource group services.

To suspend the automatic recovery actions of a resource group, you issue the `clresourcegroup suspend` command. To resume automatic recovery actions, issue the `clresourcegroup resume` command.

When you suspend the automatic recovery actions of a resource group, you also bring the resource group to a quiescent state.

A suspended resource group is *not* automatically restarted or failed over until you explicitly issue the command that resumes automatic recovery. Whether online or offline, suspended data services remain in their current state. You can still manually switch the resource group to a different state on specified nodes. You can also still enable or disable individual resources in the resource group.

A dependency or affinity is suspended and not enforced when you suspend the automatic recovery actions of a resource group that does one of the following:

- Contains a resource that has a restart dependency on another resource
- Declares a strong positive or negative affinity for another resource group

When you suspend one of these categories of resource groups, Oracle Solaris Cluster provides a warning that the dependency or affinity is suspended as well.

Note – Setting the `RG_system` property does not affect your ability to suspend or resume the automatic recovery actions of a resource group. However, if you suspend a resource group for which the `RG_system` property is set to `TRUE`, a warning message is produced. The `RG_system` property specifies that a resource group contains critical system services. If set to `TRUE`, the `RG_system` property prevents users from inadvertently stopping, deleting, or modifying a resource group or its resources.

Immediately Suspending Automatic Recovery by Killing Methods

When you suspend the automatic recovery actions of a resource group, resource methods that are executing are allowed to run until they are completed. If a serious problem occurs, you might need to suspend the automatic recovery actions of a resource group immediately. To do so, you specify the `-k` command option, which kills the following methods:

- `Prenet_start`
- `Start`
- `Monitor_start`
- `Monitor_stop`
- `Stop`
- `Postnet_stop`

Note – The `Init`, `Finis Boot`, and `Update` methods are not killed when you include this command option.

However, if you immediately suspend automatic recovery actions by killing methods, you might leave one of its resources in an error state such as `Start_failed` or `Stop_failed`. You must clear these error states yourself.

▼ How to Suspend the Automatic Recovery Actions of a Resource Group

- 1 Assume the `root` role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Suspend the automatic recovery actions of the resource group.

```
# clresourcegroup suspend resource-group
```

The resource group that you specify is not automatically started, restarted, or failed over until you resume automatic recovery actions. See [“How to Resume the Automatic Recovery Actions of a Resource Group”](#) on page 68.

▼ How to Suspend the Automatic Recovery Actions of a Resource Group Immediately

- 1 Assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Immediately suspend the automatic recovery actions of the resource group.

```
# clresourcegroup suspend -k resource-group
```

The `Prenet_start`, `Start`, `Monitor_start`, `Monitor_stop`, `Stop`, and `Postnet_stop` methods that are associated with the resource group are killed immediately. Automatic recovery actions of the resource group is suspended. The resource group is *not* automatically started, restarted, or failed over until you resume automatic recovery actions. See [“How to Resume the Automatic Recovery Actions of a Resource Group”](#) on page 68.

The `clresourcegroup suspend -k` command blocks until the specified resource group has reached a quiescent state.

▼ How to Resume the Automatic Recovery Actions of a Resource Group

- 1 Assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Resume the automatic recovery actions of the resource group.

```
# clresourcegroup resume resource-group
```

The resource group that you specify is automatically started, restarted, or failed over.

Disabling and Enabling Resource Monitors

The procedures in this section explain how to disable or enable resource fault monitors, not the resources themselves. A resource can continue to operate normally while its fault monitor is disabled. However, if the fault monitor is disabled and a data service fault occurs, automatic fault recovery is not initiated.

See the `clresource(1CL)` man page for additional information.

Note – Perform these procedures from any cluster node.

▼ How to Disable a Resource Fault Monitor

- 1 On any cluster member, assume the `root` role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Disable the resource fault monitor.

```
# clresource unmonitor [-n nodelist] resource
```

`-n nodelist` Specifies a comma-separated, ordered list of nodes on which to unmonitor the resource.

This list is optional. If you omit this list, the resource is unmonitored on all nodes in its resource group's node list.

Note – If you specify more than one node with the `-n` option, you can specify only one resource.

`resource` Specifies the name of the resource or resources.

- 3 Run the `clresource` command on each cluster node and check for monitored fields (RS Monitored) to verify that the resource fault monitor has been disabled.

```
# clresource show -v
```

Example 2–12 Disabling a Resource Fault Monitor

```
# clresource unmonitor resource-1
# clresource show -v
...
RS Monitored: no...
```

▼ How to Enable a Resource Fault Monitor

- 1 On any cluster member, assume the `root` role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Enable the resource fault monitor.

```
# clresource monitor [-n nodelist] resource
```

-n nodelist Specifies a comma-separated, ordered list of nodes on which to monitor the resource.

This list is optional. If you omit this list, the resource is monitored on all nodes in its resource group's node list.

Note – If you specify more than one node with the *-n* option, you can specify only one resource.

resource Specifies the name of the resource or resources.

- 3 **Run the `clresource` command on each cluster node and check for monitored fields (RS Monitored) to verify that the resource fault monitor has been enabled.**

```
# clresource show -v
```

Example 2–13 Enabling a Resource Fault Monitor

```
# clresource monitor resource-1
# clresource show -v
...
RS Monitored: yes...
```

Removing Resource Types

You do not need to remove resource types that are not in use. However, if you want to remove a resource type, follow this procedure.

Note – Perform this procedure from any cluster node.

▼ How to Remove a Resource Type

Removing a resource type involves disabling and removing all resources of that type in the cluster before unregistering the resource type.

Before You Begin To identify all instances of the resource type that you are removing, type the following command:

```
# clresourcetype show -v
```

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Disable each resource of the resource type that you are removing.

```
# clresource disable resource
```

resource Specifies the name of the resource to disable.
- 3 Remove each resource of the resource type that you are removing.

```
# clresource delete resource
```

resource Specifies the name of the resource to remove.
- 4 Unregister the resource type.

```
# clresourcetype unregister resource-type
```

resource-type Specifies the name of the resource type to unregister.
- 5 Verify that the resource type has been removed.

```
# clresourcetype show
```

Example 2–14 Removing a Resource Type

This example shows how to disable and remove all of the resources of a resource type (`resource-type-1`) and then unregister the resource type. In this example, `resource-1` is a resource of the resource type `resource-type-1`.

```
# clresource disable resource-1
# clresource delete resource-1
# clresourcetype unregister resource-type-1
```

See Also The following man pages:

- [clresource\(1CL\)](#)
- [clresourcetype\(1CL\)](#)

Removing Resource Groups

To remove a resource group, you must first remove all of the resources from the resource group.

Note – Perform this procedure from any cluster node.

▼ How to Remove a Resource Group

Before You Begin To identify all resources in the resource group that you are removing, type the following command:

```
# clresource show -v
```

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Run the following command to switch the resource group offline.

```
# clresourcegroup offline resource-group
```

resource-group Specifies the name of the resource group to take offline.

3 Disable all of the resources in the resource group that you are removing.

```
# clresource disable resource
```

resource Specifies the name of the resource to disable.

4 Remove all of the resources from the resource group.

For each resource, type the following command.

```
# clresource delete resource
```

resource Specifies the name of the resource to be removed.

5 Remove the resource group.

```
# clresourcegroup delete resource-group
```

resource-group Specifies the name of the resource group to be removed.

6 Verify that the resource group has been removed.

```
# clresourcegroup show
```

Example 2–15 Removing a Resource Group

This example shows how to remove a resource group (`resource-group-1`) after you have removed its resource (`resource-1`).

```
# clresourcegroup offline resource-group-1
# clresource disable resource-1
# clresource delete resource-1
# clresourcegroup delete resource-group-1
```

See Also The following man pages:

- `clresource(1CL)`
- `clresourcegroup(1CL)`

Removing Resources

Disable the resource before you remove it from a resource group.

Note – Perform this procedure from any cluster node.

▼ How to Remove a Resource

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Disable the resource that you are removing.

```
# clresource disable resource
```

resource Specifies the name of the resource to disable.
- 3 Remove the resource.

```
# clresource delete resource
```

resource Specifies the name of the resource to remove.
- 4 Verify that the resource has been removed.

```
# clresource show
```

Example 2-16 Removing a Resource

This example shows how to disable and remove a resource (`resource-1`).

```
# clresource disable resource-1  
# clresource delete resource-1
```

See Also `clresource(1CL)`

Switching the Current Primary of a Resource Group

Use the following procedure to switch over a resource group from its current primary to another node that is to become the new primary.

▼ How to Switch the Current Primary of a Resource Group

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that the following conditions are met:

- You have the following information:
 - The name of the resource group that you are switching over.
 - The names of the nodes where the resource group is to be brought online or to remain online.
- The nodes where the resource group is to be brought online or to remain online are in the cluster.
- These nodes have been set up to be potential masters of the resource group that you are switching.

To see a list of potential primaries for the resource group, type the following command:

```
# clresourcegroup show -v
```

1 On a cluster member, assume the root role that provides solaris.cluster.modify RBAC authorization.

2 Switch the resource group to a new set of primaries.

```
# clresourcegroup switch [-n nodelist] resource-group
```

-n nodelist Specifies a comma-separated, ordered list of global-cluster nodes that can master this resource group. The resource group is switched offline on all of the other nodes.

This list is optional. If you omit this list, the resource group is switched on all nodes in the resource group's node list.

resource-group Specifies the name of the resource group to switch.

Note – If any resource group that you are switching declares a strong affinity for other resource groups, the attempt to switch might fail or be delegated. For more information, see [“Distributing Online Resource Groups Among Cluster Nodes” on page 139](#).

3 Verify that the resource group has been switched to the new primary.

The output from this command indicates the state of the resource group that has been switched over.

```
# clresourcegroup status
```

Example 2–17 Switching a Resource Group to a New Primary

This example shows how to switch the resource group `resource-group-1` from its current primary `phys-schost-1` to the potential primary `phys-schost-2`.

1. To verify that the resource group is online on `phys-schost-1`, the following command is run:

```
phys-schost-1# clresourcegroup status
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
-----	-----	-----	-----
resource-group1	phys-schost-1	No	Online
	phys-schost-2	No	Offline

2. To perform the switch, the following command is run:

```
phys-schost-1# clresourcegroup switch -n phys-schost-2 resource-group-1
```

3. To verify that the group is switched to be online on `phys-schost-2`, the following command is run:

```
phys-schost-1# clresourcegroup status
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
-----	-----	-----	-----
resource-group1	phys-schost-1	No	Offline
	phys-schost-2	No	Online

See Also The [clresourcegroup\(1CL\)](#) page.

Disabling Resources and Moving Their Resource Group Into the UNMANAGED State

At times, you must bring a resource group into the UNMANAGED state before you perform an administrative procedure on it. Before you move a resource group into the UNMANAGED state, you must disable all of the resources that are part of the resource group and bring the resource group offline.

See the [clresourcegroup\(1CL\)](#) man page for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Disable a Resource and Move Its Resource Group Into the UNMANAGED State

Note – When a shared address resource is disabled, the resource might still be able to respond to ping commands from some hosts. To ensure that a disabled shared address resource cannot respond to ping commands, you must bring the resource's resource group to the UNMANAGED state. See the [ping\(1M\)](#) man page for more information.

Before You Begin Ensure that you have the following information.

- The name of each resource to be disabled
- The name of the resource group to move into the UNMANAGED state

To determine the resource and resource group names that you need for this procedure, type:

```
# clresourcegroup show -v
```

- 1 On any cluster member, assume the root role that provides `solaris.cluster.admin` RBAC authorization.
- 2 Disable all resources in the resource group.

```
# clresource disable [-n nodelist] -g resource-group +
```

`-n nodelist` Specifies a comma-separated, ordered list of nodes on which to disable the resource.

This list is optional. If you omit this list, the resource is disabled on all nodes in its resource group's node list.

Note – If you specify more than one node with the `-n` option, you can specify only one resource.

3 Switch the resource group offline.

```
# clresourcegroup offline resource-group
resource-group
```

Specifies the name of the resource group to take offline.

4 Move the resource group into the UNMANAGED state.

```
# clresourcegroup unmanage resource-group
resource-group
```

Specifies the name of the resource group to move into the UNMANAGED state.

5 Verify that the resources are disabled and that the resource group is in the UNMANAGED state.

```
# clresourcegroup show resource-group
```

Example 2–18 Disabling a Resource and Moving Its Resource Group Into the UNMANAGED State

This example shows how to disable the resource (`resource-1`) and then move the resource group (`resource-group-1`) into the UNMANAGED state.

```
# clresource disable resource-1
# clresourcegroup offline resource-group-1
# clresourcegroup unmanage resource-group-1
# clresourcegroup show resource-group-1

=== Resource Groups and Resources ===

Resource Group:                resource-group-1
RG_description:                <NULL>
RG_mode:                      Failover
RG_state:                     Unmanaged
Fallback:                     False
Nodelist:                     phys-schost-1 phys-schost-2

--- Resources for Group resource-group-1 ---

Resource:                      resource-1
Type:                          SUNW.LogicalHostname:2
Type_version:                  2
Group:                         resource-group-1
R_description:
Resource_project_name:        default
Enabled{phys-schost-1}:       False
Enabled{phys-schost-2}:       False
Monitored{phys-schost-1}:     True
Monitored{phys-schost-2}:     True
```

See Also The following man pages:

- [clresource\(1CL\)](#)
- [clresourcegroup\(1CL\)](#)

Displaying Resource Type, Resource Group, and Resource Configuration Information

Before you perform administrative procedures on resources, resource groups, or resource types, view the current configuration settings for these objects.

Note – You can view configuration settings for resources, resource groups, and resource types from any cluster node.

You can also use the `clresourcetype`, `clresourcegroup`, and `clresource` commands to check status information about specific resource types, resource groups, and resources. For example, the following command specifies that you want to view specific information about the resource `apache-1` only.

```
# clresource show apache-1
```

For more information, see the following man pages:

- [clresourcetype\(1CL\)](#)
- [clresourcegroup\(1CL\)](#)
- [clresource\(1CL\)](#)

Changing Resource Type, Resource Group, and Resource Properties

Oracle Solaris Cluster defines standard properties for configuring resource types, resource groups, and resources. These standard properties are described in the following sections:

- “Resource Type Properties” on page 23
- “Resource Properties” on page 23
- “Resource Group Properties” on page 23

Resources also have extension properties, which are predefined for the data service that represents the resource. For a description of the extension properties of a data service, see the documentation for the data service.

To determine whether you can change a property, see the Tunable entry for the property in the description of the property.

The following procedures describe how to change properties for configuring resource types, resource groups, and resources.

▼ How to Change Resource Type Properties

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the following information.

- The name of the resource type to change.
- The name of the resource type property to change. For resource types, you can change only certain properties. To determine whether you can change a property, see the Tunable entry for the property in the [rt_properties\(5\)](#) man page.

Note – You cannot change the `Installed_nodes` property explicitly. To change this property, specify the `-n installed-node-list` option of the `clresourcetype` command.

- 1 **On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**
- 2 **Run the `clresourcetype` command to determine the name of the resource type that you need for this procedure.**

```
# clresourcetype show -v
```

- 3 **Change the resource type property.**

For resource types, you can change only certain properties. To determine whether you can change a property, see the Tunable entry for the property in the [rt_properties\(5\)](#) man page.

```
# clresourcetype set -n installed-node-list \  
[-p property=new-value]resource-type
```

`-n installed-node-list` Specifies the names of nodes on which this resource type is installed.

`-p property=new-value` Specifies the name of the standard property to change and the new value of the property.

You cannot change the `Installed_nodes` property explicitly. To change this property, specify the `-n installed-node-list` option of the `clresourcetype` command.

4 Verify that the resource type property has been changed.

```
# clresourcetype show resource-type
```

Example 2–19 Changing a Resource Type Property

This example shows how to change the SUNW.apache property to define that this resource type is installed on the cluster nodes `phys-schost-1` and `phys-schost-2`.

```
# clresourcetype set -n phys-schost-1,phys-schost-2 SUNW.apache
# clresourcetype show SUNW.apache
```

```
Resource Type:                SUNW.apache:4.2
RT_description:               Apache Web Server on Oracle Solaris Cluster
RT_version:                   4.2
API_version:                  2
RT_basedir:                   /opt/SUNWscapc/bin
Single_instance:              False
Proxy:                         False
Init_nodes:                   All potential masters
Installed_nodes:              All
Failover:                     False
Pkglist:                      <NULL>
RT_system:                    False
Global_zone:                  False
```

▼ How to Change Resource Group Properties

This procedure explains how to change resource group properties. For a description of resource group properties, see the [rg_properties\(5\)](#) man page.

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the following information.

- The name of the resource group to change
- The name of the resource group property to change and its new value

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Change the resource group property.

```
# clresourcegroup set -p property=new-value resource-group
-p property      Specifies the name of the property to change
resource-group  Specifies the name of the resource group
```

3 Verify that the resource group property has been changed.

```
# clresourcegroup show resource-group
```

Example 2–20 Changing a Resource Group Property

This example shows how to change the Failback property for the resource group (resource-group-1).

```
# clresourcegroup set-p Failback=True resource-group-1
# clresourcegroup show resource-group-1
```

▼ How to Change Resource Properties

This procedure explains how to change extension properties and standard properties of a resource.

- For a description of standard resource properties, see the [r_properties\(5\)](#) man page.
- For a description of the extension properties of a resource, see the documentation for the resource's resource type.

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the following information.

- The name of the resource with the property to change
- The name of the property to change

1 On a cluster member, assume the root role that provides solaris.cluster.modify RBAC authorization.**2 View the current resource property settings.**

```
# clresource show -v resource
```

3 Change the resource property.

```
# clresource set -p standard-property=new-value | -p "extension-property \  
[{node-specifier}]"=new-value resource
```

```
-p standard-property=new-value
```

Specifies the name of the standard property to change.

```
-p "extension-property[{node-specifier}]"=new-value
```

Specifies the name of the extension property to change.

The *node-specifier* is an *optional* qualifier to the `-p` and `-x` options. This qualifier indicates that the extension property or properties on *only* the specified node or nodes are to be set when the resource is created. The specified extension properties on other nodes in the cluster are not set. If you do not include *node-specifier*, the specified extension properties on all nodes in the cluster are set. You can specify a node name or a node identifier for *node-specifier*. Examples of the syntax of *node-specifier* include the following:

```
-p "myprop{phys-schost-1}"
```

The braces (`{}`) indicate that you are setting the specified extension property on only node `phys-schost-1`. For most shells, the double quotation marks (`"`) are required.

Note – The extension property that you specify with *node-specifier* must be declared in the RTR file as a per-node property.

resource

Specifies the name of the resource.

4 Verify that the resource property has been changed.

```
# clresource show -v resource
```

Example 2–21 Changing a Standard Resource Property

This example shows how to change the system-defined `Start_timeout` property for the resource (`resource-1`).

```
# clresource set -p start_timeout=30 resource-1  
# clresource show -v resource-1
```

Example 2–22 Changing an Extension Resource Property

This example shows how to change an extension property (`Log_level`) for the resource (`resource-1`).

```
# clresource set -p Log_level=3 resource-1  
# clresource show -v resource-1
```

▼ How to Change Resource Dependency Properties

This procedure explains how to set a resource dependency property. The RGM supports dependencies of one resource upon another. You can specify per-node resource dependencies, which might differ for each per-node instance of a resource. Per-node instances are instances of the resource that are online simultaneously (in a multi-mastered resource group) or disjointly in time (in a failover resource group) on different nodes. For a description of resource properties, see the [r_properties\(5\)](#) man page.

You can use the `clsetup` utility or the CLI to set a resource dependency. The following procedure shows the steps for the `clsetup` utility.

1 Assume the root role on any cluster node.

2 Start the `clsetup` utility.

```
# clsetup
```

The `clsetup` main menu is displayed.

Note – If you want to use the CLI to set a per-node dependency on a subset of cluster nodes, specify each per-node dependency in the following form: *resourcename@nodename*.

3 Type the number for the option for resource groups and press Return.

The Resource Group menu is displayed.

4 Type the number for the option for changing the properties of a resource and press Return.

The Change Properties of a Resource screen provides a description of this task.

5 Type yes and press Return.

A menu of options for this task is displayed.

6 Type the number for the option for changing standard resource properties and press Return.

A menu of options for this task is displayed.

7 Type the number for the option for the resource whose properties you want to change and press Return.

Only those standard properties that can be changed while the resource is in this state are shown. You might have to disable the resource in order to change certain properties. Check the [r_properties\(5\)](#) man page for more information on setting standard resource properties.

You can choose to change the `resource_dependencies`, `resource_dependencies_weak`, `resource_dependencies_restart`, or `resource_dependencies_offline_restart` properties.

8 Press Return when the resource state is displayed.

9 Type the number for the option for the dependency property you want to change and press Return.

The current property name, type, description, and value are displayed.

10 Type the new values for the dependency list and press Return.

Specify each resource on which this resource is to depend using the following format: *resource-name*, *resource-name{qualifier}*, or *resource-name@node*. See the text on the screen for more information.

For example, you can change the value of the `resource_dependencies` property from `rs1` to `rs1@mynode1,rs2@mynode2,rs3`.

11 Type yes and press Return.**12 Verify that the dependency was set by pressing Return after the *Command completed successfully* message appears.**

The new value you typed appears in the Current Setting column for the property.

Example 2–23 Changing Resource Dependency Properties

The following example shows how to use the `clresource` command to set a per-node resource dependency that is dependent on two different logical hostname resources. The example uses a scalable resource called `gds-rs` and sets the dependency of `gds-rs` on `trancos-3-rs` on `ptrancos1` and `trancos-4-rs` on `ptrancos2`.

```
ptrancos1# clresource set -p resource_dependencies=trancos-3-rs@ptrancos1, \
trancos-4-rs@ptrancos2 gds-rs
ptrancos1# clresource show -p resource_dependencies gds-rs

=== Resources ===

Resource:                                gds-rs
Resource_dependencies: trancos-3-rs@ptrancos1 trancos-4-rs@ptrancos2

--- Standard and extension properties ---
```

Example 2–24 Viewing Resource Dependency Properties

The following example shows how to use the `scha_resource_get` command to retrieve a per-node resource dependency that is dependent on two different logical hostname resources. To set a per-node resource dependency, you must use the `clresource set` command. The example uses a scalable resource called `gds-rs` and sets the dependency of `gds-rs` on `trancos-3-rs` on `ptrancos1` and `trancos-4-rs` on `ptrancos2`.

From the `ptrancos1` node:

```
ptrancos1(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES -R gds-rs
trancos-3-rs
ptrancos1(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos1
trancos-3-rs
ptrancos1(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos2
trancos-4-rs
```

```
ptrancos1(/root)$ scha_resource_get -Q -O RESOURCE_DEPENDENCIES -R gds-rs
trancos-3-rs@ptrancos1
trancos-4-rs@ptrancos2
ptrancos1(/root)$ scha_resource_get -O NETWORK_RESOURCES_USED -R gds-rs
trancos-3-rs
```

From the ptrancos2 node:

```
ptrancos2(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES -R gds-rs
trancos-4-rs
ptrancos2(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos1
trancos-3-rs
ptrancos2(/root)$ scha_resource_get -O RESOURCE_DEPENDENCIES_NODE -R gds-rs ptrancos2
trancos-4-rs
ptrancos2(/root)$ scha_resource_get -Q -O RESOURCE_DEPENDENCIES -R gds-rs
trancos-3-rs@ptrancos1
trancos-4-rs@ptrancos2
ptrancos2(/root)$ scha_resource_get -O NETWORK_RESOURCES_USED -R gds-rs
trancos-4-rs
```

▼ How to Modify a Logical Hostname Resource or a Shared Address Resource

By default, logical hostname resources and shared address resources use name services for name resolution. You might configure a cluster to use a name service that is running on the same cluster. During the failover of a logical hostname resource or a shared address resource, a name service that is running on the cluster might also be failing over. If the logical hostname resource or the shared address resource uses the name service that is failing over, the resource fails to fail over.

Note – Configuring a cluster to use a name server that is running on the same cluster might impair the availability of other services on the cluster.

To prevent such a failure to fail over, modify the logical hostname resource or the shared address resource to bypass name services. To modify the resource to bypass name services, set the `CheckNameService` extension property of the resource to `false`. You can modify the `CheckNameService` property at any time.

Note – If your version of the resource type is earlier than 2, you must upgrade the resource type before you attempt to modify the resource. For more information, see [“Upgrading a Preregistered Resource Type” on page 93](#).

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify RBAC` authorization.

2 Change the resource property.

```
# clresource set -p CheckNameService=false resource
```

-p CheckNameService=false Sets the CheckNameService extension property of the resource to false.

resource Specifies the name of the logical hostname resource or shared address resource that you are modifying.

Clearing the STOP_FAILED Error Flag on Resources

When the `Failover_mode` resource property is set to `NONE` or `SOFT`, a failure of the resource's `STOP` method causes the following effects:

- The individual resource goes into the `STOP_FAILED` state.
- The resource group that contains the resource goes into the `ERROR_STOP_FAILED` state.

In this situation, you cannot perform the following operations:

- Bringing online the resource group on any node
- Adding resources to the resource group
- Removing resources from the resource group
- Changing the properties of the resource group
- Changing the properties of resources in the resource group

▼ How to Clear the STOP_FAILED Error Flag on Resources

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the following information.

- The name of the node where the resource is `STOP_FAILED`
- The name of the resource and resource group that are in `STOP_FAILED` state

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**
- 2 Identify which resources have gone into the `STOP_FAILED` state and on which nodes.**

```
# clresource status
```

3 Manually stop the resources and their monitors on the nodes on which they are in STOP_FAILED state.

This step might require that you kill processes or run commands that are specific to resource types or other commands.

4 Clear the STOP_FAILED error flag on the resources.

```
# clresource clear -f STOP_FAILED -n nodelist resource
```

-f STOP_FAILED Specifies the flag name.

-n *nodelist* Specifies a comma-separated list of the names of the nodes where the resource is in the STOP_FAILED state. The list may contain one node name or more than one node name.

resource Specifies the name of the resource.

5 Check the resource group state on the nodes where you cleared the STOP_FAILED flag in Step 4.

```
# clresourcegroup status
```

The resource group state should now be OFFLINE or ONLINE.

The resource group remains in the ERROR_STOP_FAILED state in the following combination of circumstances:

- The resource group was being switched offline when the STOP method failure occurred.
- The resource that failed to stop had a dependency on other resources in the resource group.

6 If the resource group remains in the ERROR_STOP_FAILED state, correct the error as follows.

a. Switch the resource group offline on the appropriate nodes.

```
# clresourcegroup offline resource-group
```

resource-group Specifies the name of the resource group to switch offline.

b. Switch the resource group to the ONLINE state.

See Also The following man pages:

- [clresource\(1CL\)](#)
- [clresourcegroup\(1CL\)](#)

Clearing the Start_failed Resource State

The Start_failed resource state indicates that a Start or Prenet_start method failed or timed out on a resource, but its resource group came online anyway. The resource group comes online even though the resource has been placed in a faulted state and might not be providing service. This state can occur if the resource's Failover_mode property is set to None or to another value that prevents the failover of the resource group.

Unlike the Stop_failed resource state, the Start_failed resource state does *not* prevent you or the Oracle Solaris Cluster software from performing actions on the resource group. You need only to execute a command that restarts the resource.

Use any one of the following procedures to clear this condition.

▼ How to Clear a Start_failed Resource State by Switching Over a Resource Group

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that the following conditions are met:

- You have the following information:
 - The name of the resource group that you are switching over
 - The name of the node on which to switch over the resource group
- The nodes where the resource group is to be brought online or to remain online are in the cluster.

1 On a cluster member, assume the root role that provides solaris.cluster.modify RBAC authorization.

2 Switch the resource group to the new node.

```
# clresourcegroup switch [-n nodelist] resource-group
```

-n nodelist Specifies a comma-separated, ordered list of nodes that can master this resource group. This resource group is switched offline on all of the other nodes.

This list is optional. If you omit this list, the resource group is switched on all nodes in the resource group's node list.

resource-group Specifies the name of the resource group to switch.

Note – If any resource group that you are switching declares a strong affinity for other resource groups, the attempt to switch might fail or be delegated. For more information, see [“Distributing Online Resource Groups Among Cluster Nodes” on page 139](#).

- 3 **Verify that the resource group has been switched to the new node and that the Start_failed resource state is cleared.**

```
# clresourcegroup status
```

The output from this command indicates the state of the resource and the resource group that has been switched over.

Example 2–25 Clearing a Start_failed Resource State by Switching Over a Resource Group

This example shows how to clear a Start_failed resource state that has occurred on the rscon resource in the resource-group-1 resource group. The command clears this condition by switching the resource group to the cluster node phys-schost-2.

1. To verify that the resource is in the Start_failed resource state on phys-schost-1, the following command is run:

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Message
rscon	phys-schost-1	Faulted	Faulted
	phys-schost-2	Offline	Offline
hastor	phys-schost-1	Online	Online
	phys-schost-2	Offline	Offline

2. To perform the switch, the following command is run:

```
# clresourcegroup switch -n phys-schost-2 resource-group-1
```

3. To verify that the resource group is switched to be online on phys-schost-2 and that the Start_failed resource status is cleared, the following command is run:

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Message
rscon	phys-schost-1	Offline	Offline
	phys-schost-2	Online	Online
hastor	phys-schost-1	Online	Online
	phys-schost-2	Offline	Offline

See Also The `clresourcegroup(1CL)` man page.

▼ How to Clear a Start_failed Resource State by Restarting a Resource Group

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that the following conditions are met:

- You have the following information:
 - The name of the resource group that you are restarting
 - The name of the node on which to restart the resource group
- The nodes where the resource group is to be brought online or to remain online are cluster nodes.

1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

2 Restart the resource group.

```
# clresourcegroup restart -n node resource-group
```

-n node Specifies the name of the node on which the resource group is to be restarted. This resource group is switched offline on all of the other nodes.

resource-group Specifies the name of the resource group to restart.

3 Verify that the resource group has been restarted on the new node and that the Start_failed resource state is cleared.

```
# clresourcegroup status
```

The output from this command indicates the state of the resource and the resource group that has been restarted.

Example 2–26 Clearing a Start_failed Resource State by Restarting a Resource Group

This example shows how to clear a Start_failed resource state that has occurred on the `rscon` resource in the `resource-group-1` resource group. The command clears this condition by restarting the resource group on the cluster node `phys-schost-1`.

1. Verify that the resource is in the Start_failed resource state on `phys-schost-1`.

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Message
rscon	phys-schost-1 phys-schost-2	Faulted Offline	Faulted Offline
hastor	phys-schost-1 phys-schost-2	Online Offline	Online Offline

- Restart the resource.

```
# clresourcegroup restart -n phys-schost-1 -g resource-group-1
```

- Verify that the resource group is restarted on phys-schost-1 and that the Start_failed resource status is cleared.

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Message
rscon	phys-schost-1	Offline	Offline
rscon	phys-schost-2	Online	Online
hastor	phys-schost-1	Online	Online
hastor	phys-schost-2	Offline	Offline

See Also The `clresourcegroup(1CL)` man page.

▼ How to Clear a Start_failed Resource State by Disabling and Enabling a Resource

Note – Perform this procedure from any cluster node.

Before You Begin Ensure that you have the name of the resource that you are disabling and enabling.

- On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- Disable and then enable the resource.

```
# clresource disable resource
```

```
# clresource enable resource
```

resource Specifies the name of the resource.

- 3 **Verify that the resource has been disabled and enabled and that the Start_failed resource state is cleared.**

```
# clresource status
```

The output from this command indicates the state of the resource that has been disabled and re-enabled.

Example 2–27 Clearing a Start_failed Resource State by Disabling and Enabling a Resource

This example shows how to clear a Start_failed resource state that has occurred on the rscon resource by disabling and enabling the resource.

1. Verify that the resource is in the Start_failed resource state.

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Message
rscon	phys-schost-1 phys-schost-2	Faulted Offline	Faulted Offline
hastor	phys-schost-1 phys-schost-2	Online Offline	Online Offline

2. Disable and re-enable the resource.

```
# clresource disable rscon
```

```
# clresource enable rscon
```

3. Verify that the resource is re-enabled and that the Start_failed resource status is cleared.

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Message
rscon	phys-schost-1 phys-schost-2	Online Offline	Online Offline
hastor	phys-schost-1 phys-schost-2	Online Offline	Online Offline

See Also The [clresource\(1CL\)](#) man page.

Upgrading a Preregistered Resource Type

In a previous version of the cluster software, the following preregistered resource types were enhanced:

- `SUNW.LogicalHostname`, which represents a logical hostname
- `SUNW.SharedAddress`, which represents a shared address

The purpose of these enhancements was to enable you to modify logical hostname resources and shared address resources to bypass name services for name resolution.

Upgrade these resource types if all conditions in the following list apply:

- You are upgrading from an earlier version of Oracle Solaris Cluster.
- You need to use the new features of the resource types.

For general instructions that explain how to upgrade a resource type, see [“Upgrading a Resource Type” on page 33](#). The information that you need to complete the upgrade of the preregistered resource types is provided in the subsections that follow.

Information for Registering the New Resource Type Version

To determine the version of the resource type that is registered, use one command from the following list:

- `clresourcetype list`
- `clresourcetype list -v`

EXAMPLE 2-28 Registering a New Version of the `SUNW.LogicalHostname` Resource Type

This example shows the command for registering version 4 of the `SUNW.LogicalHostname` resource type during an upgrade.

```
# clresourcetype register SUNW.LogicalHostname:4
```

Information for Migrating Existing Instances of the Resource Type

The information that you need to migrate an instance of a preregistered resource type is as follows:

- You can perform the migration at any time.

- If you need to use the new features of the preregistered resource type, the required value of the `Type_version` property is 2.
- If you are modifying the resource to bypass name services, set the `CheckNameService` extension property of the resource to `false`.

EXAMPLE 2-29 Migrating a Logical Hostname Resource

This example shows the command for migrating the logical hostname resource `lhostrs`. As a result of the migration, the resource is modified to bypass name services for name resolution.

```
# clresource set -p CheckNameService=false -p Type_version=2 lhostrs
```

Reregistering Preregistered Resource Types After Inadvertent Deletion

The resource types `SUNW.LogicalHostname` and `SUNW.SharedAddress` are preregistered. All of the logical hostname and shared address resources use these resource types. You never need to register these two resource types, but you might inadvertently delete them. If you have deleted resource types inadvertently, use the following procedure to reregister them.

Note – If you are upgrading a preregistered resource type, follow the instructions in [“Upgrading a Preregistered Resource Type” on page 93](#) to register the new resource type version.

Perform this procedure from any cluster node.

▼ How to Reregister Preregistered Resource Types After Inadvertent Deletion

- Reregister the resource type.

```
# clresourcetype register SUNW.resource-type
```

resource-type Specifies the resource type to add (reregister). The resource type can be either `SUNW.LogicalHostname` or `SUNW.SharedAddress`.

Example 2-30 Reregistering a Preregistered Resource Type After Inadvertent Deletion

This example shows how to reregister the `SUNW.LogicalHostname` resource type.

```
# clresourcetype register SUNW.LogicalHostname
```

See Also The `clresourcetype(1CL)` man page.

Adding or Removing a Node to or From a Resource Group

The procedures in this section enable you to perform the following tasks.

- Configuring a cluster node to be an additional master of a resource group
- Removing a node from a resource group

The procedures are slightly different, depending on whether you plan to add or remove the node to or from a failover or scalable resource group.

Failover resource groups contain network resources that both failover and scalable services use. Each IP subnetwork connected to the cluster has its own network resource that is specified and included in a failover resource group. The network resource is either a logical hostname or a shared address resource. Each network resource includes a list of IPMP groups that it uses. For failover resource groups, you must update the complete list of IPMP groups for each network resource that the resource group includes (the `netiflist` resource property).

The procedure for scalable resource groups involves the following steps:

1. Repeating the procedure for failover groups that contain the network resources that the scalable resource uses
2. Changing the scalable group to be mastered on the new set of hosts

For more information, see the `clresourcegroup(1CL)` man page.

Note – Run either procedure from any cluster node.

Adding a Node to a Resource Group

The procedure to follow to add a node to a resource group depends on whether the resource group is a scalable resource group or a failover resource group. For detailed instructions, see the following sections:

- “How to Add a Node to a Scalable Resource Group” on page 96
- “How to Add a Node to a Failover Resource Group” on page 96

You must supply the following information to complete the procedure.

- The names and node IDs of all of the cluster nodes
- The names of the resource groups to which you are adding the node

- The name of the IPMP group that is to host the network resources that are used by the resource group on all of the nodes

Also, be sure to verify that the new node is already a cluster member.

▼ How to Add a Node to a Scalable Resource Group

- 1 For each network resource that a scalable resource in the resource group uses, make the resource group where the network resource is located run on the new node.

See [Step 1](#) through [Step 5](#) in the following procedure for details.

- 2 Add the new node to the list of nodes that can master the scalable resource group (the `nodeList` resource group property).

This step overwrites the previous value of `nodeList`, and therefore you must include all of the nodes that can master the resource group here.

```
# clresourcegroup set [-n nodeList] resource-group
```

`-n nodeList` Specifies a comma-separated, ordered list of nodes that can master this resource group. This resource group is switched offline on all of the other nodes..

This list is optional. If you omit this list, the `NodeList` property is set to all nodes in the cluster.

`resource-group` Specifies the name of the resource group to which the node is being added.

- 3 (Optional) Update the scalable resource's `Load_balancing_weights` property to assign a weight to the node that you are adding to the resource group.

Otherwise, the weight defaults to 1. See the `clresourcegroup(1CL)` man page for more information.

▼ How to Add a Node to a Failover Resource Group

- 1 Display the current node list and the current list of IPMP groups that are configured for each resource in the resource group.

```
# clresourcegroup show -v resource-group | grep -i nodeList
# clresourcegroup show -v resource-group | grep -i netiflist
```

Note – The output of the command line for `nodeList` and `netiflist` identifies the nodes by node name. To identify node IDs, run the command `clnode show -v | grep -i node-id`.

2 Update `netiflist` for the network resources that the node addition affects.

This step overwrites the previous value of `netiflist`, and therefore you must include all the IPMP groups here.

```
# clresource set -p netiflist=netiflist network-resource
```

`-p netiflist=netiflist` Specifies a comma-separated list that identifies the IPMP groups that are on each node. Each element in *netiflist* must be in the form of `netif@node`. `netif` can be given as an IPMP group name, such as `sc_ipmp0`. The node can be identified by the node name or node ID, such as `sc_ipmp0@1` or `sc_ipmp@phys-schost-1`.

network-resource Specifies the name of the network resource (logical hostname or shared address) that is being hosted on the *netiflist* entries.

3 If the `HASStoragePlus AffinityOn` extension property equals `True`, add the node to the appropriate disk set or device group.

▪ If you are using Solaris Volume Manager, use the `metaset` command.

```
# metaset -s disk-set-name -a -h node-name
```

`-s disk-set-name` Specifies the name of the disk set on which the `metaset` command is to work

`-a` Adds a drive or host to the specified disk set

`-h node-name` Specifies the node to be added to the disk set

4 Update the node list to include all of the nodes that can now master this resource group.

This step overwrites the previous value of `nodelist`, and therefore you must include all of the nodes that can master the resource group here.

```
# clresourcegroup set [-n nodelist] resource-group
```

`-n nodelist` Specifies a comma-separated, ordered list of nodes that can master this resource group. This resource group is switched offline on all the other nodes.

This list is optional. If you omit this list, the `NodeList` property is set to all nodes in the cluster.

resource-group Specifies the name of the resource group to which the node is being added.

5 Verify the updated information.

```
# clresourcegroup show -v resource-group | grep -i nodelist
# clresourcegroup show -v resource-group | grep -i netiflist
```

Example 2–31 Adding a Node to a Resource Group

This example shows how to add a cluster node `phys-schost-2` to the resource group `resource-group-1` that contains the logical hostname `schost-2`.

```
# clresourcegroup show -v resource-group-1 | grep -i nodelist
Nodelist:      phys-schost-1 phys-schost-3
# clresourcegroup show -v resource-group-1 | grep -i netiflist
Res property name: NetIfList
Res property class: extension
List of IPMP
interfaces on each node
Res property type: stringarray
Res property value: sc_ipmp0@1 sc_ipmp0@3
```

Only nodes 1 and 3 have been assigned IPMP groups. You must add an IPMP group for node 2.

```
# clresource set -p netiflist=sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3 schost-2
# metaset -s red -a -h phys-schost-2
# clresourcegroup set -n phys-schost-1,phys-schost-2,phys-schost-3 resource-group-1
# clresourcegroup show -v resource-group-1 | grep -i nodelist
Nodelist:      phys-schost-1 phys-schost-2
                phys-schost-3
# clresourcegroup show -v resource-group-1 | grep -i netiflist
Res property value: sc_ipmp0@1 sc_ipmp0@2
                   sc_ipmp0@3
```

Removing a Node From a Resource Group

The procedure to follow to remove a node from a resource group depends on whether the resource group is a scalable resource group or a failover resource group. For detailed instructions, see the following sections:

- [“How to Remove a Node From a Scalable Resource Group” on page 99](#)
- [“How to Remove a Node From a Failover Resource Group” on page 100](#)
- [“How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources” on page 101](#)

Note – If the node that you want to remove appears in a per-node resource dependency, you must remove that node from the per-node dependency before you can remove it from the resource group. For more information, see [“How to Change Resource Dependency Properties” on page 82](#).

To complete the procedure, you must supply the following information.

- Node names and node IDs of all of the cluster nodes

```
# clnode show -v | grep -i "Node ID"
```

- The name of the resource group or the names of the resource groups from which you plan to remove the node

```
# clresourcegroup show | grep "NodeList"
```

- Names of the IPMP groups that are to host the network resources that are used by the resource groups on all of the nodes

```
# clresourcegroup show -v | grep "NetIfList.*value"
```

Additionally, be sure to verify that the resource group **is not mastered** on the node that you are removing. If the resource group **is mastered** on the node that you are removing, run the `clresourcegroup` command to switch the resource group offline from that node. The following `clresourcegroup` command brings the resource group offline from a given node, provided that *new-masters* does not contain that node.

```
# clresourcegroup switch -n new-masters resource-group
```

-n new-masters Specifies the nodes that is now to master the resource group.

resource-group Specifies the name of the resource group that you are switching. This resource group is mastered on the node that you are removing.

For more information, see the `clresourcegroup(1CL)` man page.



Caution – If you plan to remove a node from all the resource groups, and you use a scalable services configuration, first remove the node from the scalable resource groups. Then remove the node from the failover groups.

▼ How to Remove a Node From a Scalable Resource Group

A scalable service is configured as two resource groups, as follows.

- One resource group is a scalable group that contains the scalable service resource.
- One resource group is a failover group that contains the shared address resources that the scalable service resource uses.

Additionally, the `RG_dependencies` property of the scalable resource group is set to configure the scalable group with a dependency on the failover resource group. For information about this property, see the `rg_properties(5)` man page.

For details about scalable service configuration, see *Oracle Solaris Cluster Concepts Guide*.

Removing a node from the scalable resource group causes the scalable service to no longer be brought online on that node. To remove a node from the scalable resource group, perform the following steps.

- 1 **Remove the node from the list of nodes that can master the scalable resource group (the `nodeList` resource group property).**

```
# clresourcegroup set [-n nodelist] scalable-resource-group
```

`-n nodelist` Specifies a comma-separated, ordered list of nodes that can master this resource group. This resource group is switched offline on all the other nodes.

This list is optional. If you omit this list, the `NodeList` property is set to all nodes in the cluster.

`scalable-resource-group` Specifies the name of the resource group from which the node is being removed.

- 2 **(Optional) Remove the node from the failover resource group that contains the shared address resource.**

For details, see [“How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources”](#) on page 101.

- 3 **(Optional) Update the `Load_balancing_weights` property of the scalable resource to remove the weight of the node that you are removing from the resource group.**

See Also The `clresourcegroup(1CL)` man page.

▼ How to Remove a Node From a Failover Resource Group

Perform the following steps to remove a node from a failover resource group.



Caution – If you plan to remove a node from all of the resource groups, and you use a scalable services configuration, first remove the node from the scalable resource groups. Then use this procedure to remove the node from the failover groups.

If the failover resource group contains shared address resources that scalable services use, see [“How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources”](#) on page 101.

- 1 **Update the node list to include all of the nodes that can now master this resource group.**

This step removes the node and overwrites the previous value of the node list. Be sure to include all of the nodes that can master the resource group here.

```
# clresourcegroup set [-n nodelist] failover-resource-group
```

<code>-n nodelist</code>	Specifies a comma-separated, ordered list of nodes that can master this resource group. This resource group is switched offline on all the other nodes.
	This list is optional. If you omit this list, the <code>NodeList</code> property is set to all nodes in the cluster.
<code>failover-resource-group</code>	Specifies the name of the resource group from which the node is being removed.

2 Display the current list of IPMP groups that are configured for each resource in the resource group.

```
# clresourcegroup show -v failover-resource-group | grep -i netiflist
```

3 Update `netiflist` for network resources that the removal of the node affects.

This step overwrites the previous value of `netiflist`. Be sure to include all of the IPMP groups here.

```
# clresource set -p netiflist=netiflist network-resource
```

Note – The output of the preceding command line identifies the nodes by node name. Run the command line `clnode show -v | grep -i "Node ID"` to find the node ID.

<code>-p netiflist=netiflist</code>	Specifies a comma-separated list that identifies the IPMP groups that are on each node. Each element in <code>netiflist</code> must be in the form of <code>netif@node</code> . <code>netif</code> can be given as an IPMP group name, such as <code>sc_ipmp0</code> . The node can be identified by the node name or node ID, such as <code>sc_ipmp0@1</code> or <code>sc_ipmp@phys-schost-1</code> .
<code>network-resource</code>	Specifies the name of the network resource that is hosted on the <code>netiflist</code> entries.

Note – Oracle Solaris Cluster does not support the use of the adapter name for `netif`.

4 Verify the updated information.

```
# clresourcegroup show -v failover-resource-group | grep -i nodelist
# clresourcegroup show -v failover-resource-group | grep -i netiflist
```

▼ How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources

In a failover resource group that contains shared address resources that scalable services use, a node can appear in the following locations.

- The node list of the failover resource group
- The `auxnodelist` of the shared address resource

To remove the node from the node list of the failover resource group, follow the procedure “[How to Remove a Node From a Failover Resource Group](#)” on page 100.

To modify the `auxnodelist` of the shared address resource, you must remove and recreate the shared address resource.

If you remove the node from the failover group’s node list, you can continue to use the shared address resource on that node to provide scalable services. To continue to use the shared address resource, you must add the node to the `auxnodelist` of the shared address resource. To add the node to the `auxnodelist`, perform the following steps.

Note – You can also use the following procedure to **remove** the node from the `auxnodelist` of the shared address resource. To remove the node from the `auxnodelist`, you must delete and recreate the shared address resource.

- 1 **Switch the scalable service resource offline.**
- 2 **Remove the shared address resource from the failover resource group.**
- 3 **Create the shared address resource.**

Add the node ID or node name of the node that you removed from the failover resource group to the `auxnodelist`.

```
# clressharedaddress create -g failover-resource-group \
  -X new-auxnodelist shared-address
```

failover-resource-group The name of the failover resource group that used to contain the shared address resource.

new-auxnodelist The new, modified `auxnodelist` with the desired node added or removed.

shared-address The name of the shared address.

Example – Removing a Node From a Resource Group

This example shows how to remove a node (`phys-schost-3`) from a resource group (`resource-group-1`) that contains a logical hostname resource (`schost-1`).

```
# clresourcegroup show -v resource-group-1 | grep -i nodelist
Nodelist:      phys-schost-1 phys-schost-2
                phys-schost-3
# clresourcegroup set -n phys-schost-1,phys-schost-2 resource-group-1
# clresourcegroup show -v resource-group-1 | grep -i netiflist
```

```
( Res property name: NetIfList
Res property class: extension
( List of IPMP
interfaces on each node
( Res property type: stringarray
  Res property value: sc_ipmp0@1 sc_ipmp0@2
                    sc_ipmp0@3

(sc_ipmp0@3 is the IPMP group to be removed.)

# clresource set -p netiflist=sc_ipmp0@1,sc_ipmp0@2 schost-1
# clresourcegroup show -v resource-group-1 | grep -i nodelist
Nodelist:      phys-schost-1 phys-schost-2
# clresourcegroup show -v resource-group-1 | grep -i netiflist
Res property value: sc_ipmp0@1 sc_ipmp0@2
```

Synchronizing the Startups Between Resource Groups and Device Groups

After a cluster boots or services fail over to another node, global devices and local and cluster file systems might require time to become available. However, a data service can run its START method before global devices and local and cluster file systems come online. If the data service depends on global devices or local and cluster file systems that are not yet online, the START method times out. In this situation, you must reset the state of the resource groups that the data service uses and restart the data service manually.

To avoid these additional administrative tasks, use the HASStoragePlus resource type. Add an instance of HASStoragePlus to all resource groups whose data service resources depend on global devices or local and cluster file systems. Instances of these resource types perform the following operations:

- Forcing the START method of the other resources in the same resource group to wait until global devices and local and cluster file systems become available.

If an application resource is configured on top of an HASStoragePlus resource, the application resource must define the offline restart dependency on the underlying HASStoragePlus resource. This ensures that the application resource comes online after the dependent HASStoragePlus resource comes online, and goes offline before the HASStoragePlus resource goes offline.

The following command creates an offline restart dependency from an application resource to a HASP resource:

```
# clrs set -p Resource_dependencies_offline_restart=hasp_rs applicaton_rs
```

To create an HASStoragePlus resource, see [“How to Set Up the HASStoragePlus Resource Type for New Resources”](#) on page 107.

Managed Entity Monitoring by HAStoragePlus

All entities that are managed by the HAStoragePlus resource type are monitored. The SUNWHASStoragePlus resource type provides a fault monitor to monitor the health of the entities managed by the HASP resource, including global devices, file systems, and ZFS storage pools. The fault monitor runs fault probes on a regular basis. If one of the entities becomes unavailable, the resource is restarted or a failover to another node is performed. If more than one entity is monitored, the fault monitor probes them all at the same time. Ensure that all configuration changes to the managed entities are completed before you enable monitoring.

Note – Version 9 of the HAStoragePlus resource fault monitor probes the devices and file systems it manages by reading and writing to the file systems. If a read operation is blocked by any software on the I/O stack and the HASStoragePlus resource is required to be online, the user must disable the fault monitor. For example, you must unmonitor the HASStoragePlus resource managing the Availability Suite Remote Replication volumes because Availability Suite from Oracle blocks reading from any bitmap volume or any data volume in the NEED SYNC state. The HASStoragePlus resource managing the Availability Suite volumes must be online at all times.

For more information on the properties that enable monitoring for managed entities, see the [SUNW.HASStoragePlus\(5\)](#) man page.

For instructions on enabling and disabling monitoring for managed entities, see “[How to Enable a Resource Fault Monitor](#)” on page 69.

Depending on the type of managed entity, the fault monitor probes the target by reading or writing to it. If more than one entity is monitored, the fault monitor probes them all at the same time.

TABLE 2-2 What the Fault Monitor Verifies

Monitored Entity	What the Fault Monitor Verifies
Global device	<ul style="list-style-type: none"> ■ The device group is online or degraded. ■ The device is readable.
Raw device group	<ul style="list-style-type: none"> ■ The device group is online or degraded. ■ For each device of the device group, its path (<code>/dev/global/rdisk/device</code>) is available. ■ Partitions of every device are readable.

TABLE 2-2 What the Fault Monitor Verifies (Continued)

Monitored Entity	What the Fault Monitor Verifies
Solaris Volume Manager device group	<ul style="list-style-type: none"> ■ The device group is online or degraded. ■ The path of the metaset (<code>/dev/md/metaset</code>) is valid. ■ The Solaris Volume Manager reported status from the primary of the device group: <ul style="list-style-type: none"> ■ The unmirrored metadevice is not in any of the following error states: Needs Maintenance, Last Erred, or Unavailable. ■ At least one submirror of a mirror is not in an error state. An error with some, but not all submirrors, is treated as partial error. ■ The unmirrored metadevice is readable from the primary. ■ Some submirrors of a mirror are readable. An error with some, but not all, submirrors is treated as partial error.
File systems (including UFS and PxFS)	<ul style="list-style-type: none"> ■ The file system is mounted. ■ Every device under the file system is readable. ■ The file system is readable, if the <code>I00ption</code> property is set to <code>ReadOnly</code>. ■ The file system is writable, if the <code>I00ption</code> property is set to <code>ReadWrite</code>. ■ If the file system is mounted read-only but the <code>I00ption</code> property is set to <code>ReadWrite</code>, the fault monitor issues a warning and then tries to read it (rather than write to it). ■ To avoid having the <code>HASoragePlus</code> resource go offline when a file system hits its quota, set the <code>I00ption</code> to <code>ReadOnly</code>. The <code>ReadOnly</code> option ensures that the fault monitor will not attempt to write to the file system.

TABLE 2-2 What the Fault Monitor Verifies (Continued)

Monitored Entity	What the Fault Monitor Verifies
ZFS storage pool	<ul style="list-style-type: none"> ■ The pool status is OK or Degraded. ■ Each non-legacy file system is mounted. ■ Each non-legacy file system is readable, if the <code>I00ption</code> property is set to <code>ReadOnly</code>. ■ Each non-legacy file system is writable, if the <code>I00ption</code> property is set to <code>ReadWrite</code>. ■ If a non-legacy file system is mounted read-only but the <code>I00ption</code> property is set to <code>ReadWrite</code>, the fault monitor issues a warning and then tries to read it (rather than write to it). ■ To avoid having the <code>HASStoragePlus</code> resource go offline when a file system hits its quota, set the <code>I00ption</code> to <code>ReadOnly</code>. The <code>ReadOnly</code> option ensures that the fault monitor will not attempt to write to the file system. <p>Note – When all connections to a top-level ZFS storage device are lost, queries about the ZFS storage pool or associated file system will hang. To prevent the fault monitor from hanging, you must set the <code>fail_mode</code> property of the ZFS storage pool to <code>panic</code>.</p>

For instructions on enabling a resource fault monitor, see [“How to Enable a Resource Fault Monitor”](#) on page 69.

Troubleshooting Monitoring for Managed Entities

If monitoring is not enabled on the managed entities, perform the following troubleshooting steps:

1. Ensure that the `hasstorageplus_probe` process is running.
2. Look for error messages on the console.
3. Enable debug messages to the `syslog` file.

```
# mkdir -p /var/cluster/rgm/rt/SUNW.HASStoragePlus:9
# echo 9 > /var/cluster/rgm/rt/SUNW.HASStoragePlus:9/logLevel
```

You should also check the `/etc/syslog.conf` file to ensure that messages with the `daemon.debug` facility level are logged to the `/var/adm/messages` file. Add the `daemon.debug` entry to the `/var/adm/messages` action if it is not already present.

Additional Administrative Tasks to Configure HAStoragePlus Resources for a Zone Cluster

When you configure HAStoragePlus resources for a zone cluster, you need to perform the following additional tasks before performing the steps for global cluster:

- While configuring file systems like UFS in file system mount points, the file systems need to be configured to the zone cluster. For more information about configuring a file system to a zone cluster, see [“How to Add a Local File System to a Specific Zone-Cluster Node”](#) in *Oracle Solaris Cluster Software Installation Guide*.
- While configuring global devices in global device paths, the devices need to be configured to the zone cluster. For more information about configuring global devices to a zone cluster, see [“Adding Storage Devices to a Zone Cluster”](#) in *Oracle Solaris Cluster Software Installation Guide*.
- While configuring the ZFS file systems using Zpools, the ZFS pool needs to be configured to the zone cluster. For more information about configuring a ZFS file system to a zone cluster, see [“How to Add a ZFS Storage Pool to a Zone Cluster”](#) in *Oracle Solaris Cluster Software Installation Guide*.

▼ How to Set Up the HAStoragePlus Resource Type for New Resources

In the following example, the resource group `resource-group-1` contains the following data services.

- HA for Oracle iPlanet Web Server (formerly Sun Java System Web Server), which depends on `/global/resource-group-1`
- HA for Oracle, which depends on `/dev/global/dsk/d5s2`
- HA for NFS, which depends on `dsk/d6`

Note – To create an HAStoragePlus resource with Oracle Solaris ZFS as a highly available local file system see [“How to Set Up the HAStoragePlus Resource Type to Make a Local Solaris ZFS File System Highly Available”](#) on page 119 section.

To create an HAStoragePlus resource `hastorageplus-1` for new resources in `resource-group-1`, read [“Synchronizing the Startups Between Resource Groups and Device Groups”](#) on page 103 and then perform the following steps.

To create an HAStoragePlus resource, see [“Enabling Highly Available Local File Systems”](#) on page 112.

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.

- 2 Create the resource group `resource-group-1`.

```
# clresourcegroup create resource-group-1
```

- 3 Determine whether the resource type is registered.

The following command prints a list of registered resource types.

```
# clresourcetype show | egrep Type
```

- 4 If you need to, register the resource type.

```
# clresourcetype register SUNW.HASStoragePlus
```

- 5 Create the `HASStoragePlus` resource `hasstorageplus-1`, and define the filesystem mount points and global device paths.

```
# clresource create -g resource-group-1 -t SUNW.HASStoragePlus \  
-p GlobalDevicePaths=/dev/global/dsk/d5s2,dsk/d6 \  
-p FilesystemMountPoints=/global/resource-group-1 hasstorageplus-1
```

`GlobalDevicePaths` can contain the following values.

- Global device group names, such as `nfs-dg`, `dsk/d5`
- Paths to global devices, such as `/dev/global/dsk/d1s2`, `/dev/md/nfsdg/dsk/d10`

`FilesystemMountPoints` can contain the following values.

- Mount points of local or cluster file systems, such as `/local-fs/nfs`, `/global/nfs`

Note – `HASStoragePlus` has a `Zpools` extension property that is used to configure ZFS file system storage pools and a `ZpoolsSearchDir` extension property that is used to specify the location to search for the devices of ZFS file system storage pools. The default value for the `ZpoolsSearchDir` extension property is `/dev/dsk`. The `ZpoolsSearchDir` extension property is similar to the `-d` option of the `zpool (1M)` command.

The resource is created in the enabled state.

- 6 Add the resources (Oracle iPlanet Web Server (formerly Sun Java System Web Server), Oracle, and NFS) to `resource-group-1`, and set their dependency to `hasstorageplus-1`.

For example, for Oracle iPlanet Web Server (formerly Sun Java System Web Server), run the following command.

```
# clresource create -g resource-group-1 -t SUNW.iws \  
-p Confdir_list=/global/iws/schost-1 -p Scalable=False \  
-p Resource_dependencies=schost-1 -p Port_list=80/tcp \  
-p Resource_dependencies_offline_restart=hasstorageplus-1 resource
```

The resource is created in the enabled state.

- 7 **Verify that you have correctly configured the resource dependencies.**

```
# clresource show -v resource | egrep Resource_dependencies_offline_restart
```
- 8 **Set resource-group-1 to the MANAGED state, and bring resource-group-1 online.**

```
# clresourcegroup online -M resource-group-1
```

More Information Affinity Switchovers

The HASStoragePlus resource type contains another extension property, AffinityOn, which is a Boolean that specifies whether HASStoragePlus must perform an affinity switchover for the global devices that are defined in GlobalDevicePaths and FileSystemMountPoints extension properties. For details, see the [SUNW.HASStoragePlus\(5\)](#) man page.

Note – The setting of the AffinityOn flag is ignored for scalable services. Affinity switchovers are not possible with scalable resource groups.

▼ How to Set Up the HASStoragePlus Resource Type for Existing Resources

Before You Begin Read “[Synchronizing the Startups Between Resource Groups and Device Groups](#)” on page 103.

- 1 **Determine whether the resource type is registered.**
 The following command prints a list of registered resource types.

```
# clresourcetype show | egrep Type
```
- 2 **If you need to, register the resource type.**

```
# clresourcetype register SUNW.HASStoragePlus
```
- 3 **Create the HASStoragePlus resource hastorageplus-1.**

```
# clresource create -g resource-group \  
-t SUNW.HASStoragePlus -p GlobalDevicePaths= ... \  
-p FileSystemMountPoints=... -p AffinityOn=True hastorageplus-1
```

 The resource is created in the enabled state.
- 4 **Set up the dependency for each of the existing resources, as required.**

```
# clresource set -p Resource_Dependencies_offline_restart=hastorageplus-1 resource
```
- 5 **Verify that you have correctly configured the resource dependencies.**

```
# clresource show -v resource | egrep Resource_dependencies_offline_restart
```

Configuring an HAStoragePlus Resource for Cluster File Systems

When an HAStoragePlus resource is configured for cluster file systems and brought online, it ensures that these file systems are available. The cluster file system is supported on a root ZFS file system and a nonroot UNIX File System (UFS). The instructions in this section apply to HAStoragePlus resources with UFS. Use HAStoragePlus with local file systems if the data service is I/O intensive. See [“How to Change the Cluster File System to Local File System in an HAStoragePlus Resource”](#) on page 137 for information about how to change the file system of an HAStoragePlus resource.

The cluster file systems can be configured for zone clusters in the HAStoragePlus resources using the loopback mount mechanism. The SUNW.HAStoragePlus resource type makes the cluster file system available to a zone cluster by mounting the file system in the global cluster. The resource type then performs a loopback mount on the zone cluster nodes where the resource group is online.

Note – If you have a failover resource group, the resource group will be online on only one node. If you use a scalable resource group, the `Desired primaries` property defines the number of nodes the resource group will have online.

The cluster file systems configured in the HAStoragePlus resource type for zone clusters should be authorized for use in zone clusters using the `clzonecluster` command. For more information, see the `clzonecluster(1CL)` man page and [“How to Add a Cluster File System to a Zone Cluster”](#) in *Oracle Solaris Cluster Software Installation Guide*.

Sample Entries in /etc/vfstab for Cluster File Systems

The following examples show entries in the `/etc/vfstab` file for global devices that are to be used for cluster file systems.

Note – The entries in the `/etc/vfstab` file for cluster file systems should contain the `globalkeyword` in the mount options.

EXAMPLE 2-32 Entries in `/etc/vfstab` for a Global Device With Solaris Volume Manager

This example shows entries in the `/etc/vfstab` file for a global device that uses Solaris Volume Manager.

```
/dev/md/kappa-1/dsk/d0    /dev/md/kappa-1/rdisk/d0
/global/local-fs/nfs ufs      5  yes      logging,global
```

▼ How to Set Up the HAStoragePlus Resource for Cluster File Systems

- 1 On any node in the cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

- 2 Create a failover or scalable resource group as desired.

Perform the following step to create a failover group.

```
# clresourcegroup create resource-group
```

Perform the following step to create a scalable group.

```
# clresourcegroup create -S [-p Maximum primaries=m] [-p Desired primaries=n] \
[-n node-zone-list] resource-group
```

- 3 Register the HAStoragePlus resource type.

```
# clresourcetype register SUNW.HAStoragePlus
```

- 4 Create the HAStoragePlus resource and define the filesystem mount points.

```
# clresource create -g resource-group -t SUNW.HAStoragePlus \
-p FileSystemMountPoints="mount-point-list" hasp-resource
```

The resource is created in the enabled state.

- 5 Add the data service resources to *resource-group*, and set their dependency to *hasp-resource*.

```
# clresource set -p Resource_dependencies_offline_restart= \
hasp-resource application-resource
```

- 6 Bring online and in a managed state the resource group that contains the HAStoragePlus resource.

```
# clresourcegroup online -M resource-group
```

Example 2–33 Setting up the HAStoragePlus Resource Type with a Cluster File System in a Global Cluster

This example shows how to configure the HAStoragePlus resource with a cluster file system `/global/ufs` in a global cluster for a failover resource group.

```
phys-schost-1# vi /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot  options
#
/dev/md/apachedg/dsk/d0 /dev/md/apachedg/rdisk/d0 /global/ufs ufs 2 yes global, logging
# clresourcegroup create hasp-rg
# clresourcetype register SUNW.HAStoragePlus
# clresource create -g hasp-rg -t SUNW.HAStoragePlus -p \
```

```
FileSystemMountPoints=/global/ufs hasp-rs
# clresourcegroup online -M hasp-rg
```

Example 2–34 Setting up the HASStoragePlus Resource Type with a Cluster File System in a Zone Cluster

This example shows how to configure the HASStoragePlus resource with a cluster file system /global/ufs in a zone cluster for a scalable resource group. The cluster file system is available for the zone cluster nodes on the mount point /zone/ufs. This example configuration makes the global file system /global/ufs mounted in a global cluster and later loopback mounted on two zone-cluster nodes where the resource group is online.

```
phys-schost-1# vi /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot    options
#
/dev/md/apachedg/dsk/d0 /dev/md/apachedg/rdisk/d0 /global/ufs ufs 2 yes global, logging
# clzonecluster configure sczone
clzc:sczone> add fs
clzc:sczone:fs> set dir=/zone/ufs
clzc:sczone:fs> set special=/global/ufs
clzc:sczone:fs> set type=lofs
clzc:sczone:fs> end
clzc:sczone:fs> exit
# clresourcegroup create -Z sczone -p desired primaries=2 -p maximum primaries=2 hasp-rg
# clresourcetype register -Z sczone SUNW.HASStoragePlus
# clresource create -Z sczone -g hasp-rg -t SUNW.HASStoragePlus -p
FileSystemMountPoints=/zone/ufs hasp-rs
# clresourcegroup online -Z sczone -M hasp-rg
```

▼ How to Delete an HASStoragePlus Resource Type for Cluster File Systems

- Disable and delete the HASStoragePlus resource configured for cluster file systems.

```
# clresource delete -F -g resource-group -t SUNW.HASStoragePlus resource
```

Enabling Highly Available Local File Systems

Using a highly available local file system improves the performance of I/O intensive data services. To make a local file system highly available in an Oracle Solaris Cluster environment, use the HASStoragePlus resource type.

You can specify cluster file systems or local file systems. Cluster file systems are accessible from all nodes in a cluster. Local file systems are accessible from a single cluster node. Local file systems that are managed by a SUNW.HASStoragePlus resource are mounted on a single cluster node. These local file systems require the underlying devices to be Oracle Solaris Cluster global devices.

These file system mount points are defined in the format `paths [, ...]`. The default setting for this property is an empty list.

You can use the `SUNW.HASStoragePlus` resource type to make a file system available to zone-cluster nodes. The file systems configured in the `SUNW.HASStoragePlus` resource type for zone clusters should be authorized for use in zone clusters by using the `clzonecluster` command. For more information, see the `clzonecluster(1CL)` man page and “[Adding File Systems to a Zone Cluster](#)” in *Oracle Solaris Cluster Software Installation Guide*.

Note – Local file systems include the UNIX File System (UFS) and Solaris ZFS.

The instructions for each Oracle Solaris Cluster data service that is I/O intensive explain how to configure the data service to operate with the `HASStoragePlus` resource type. For more information, see the individual Oracle Solaris Cluster data service guides.

Note – Do *not* use the `HASStoragePlus` resource type to make a root file system highly available.

Oracle Solaris Cluster provides the following tools for setting up the `HASStoragePlus` resource type to make local file systems highly available:

- The `clsetup` utility.
- Oracle Solaris Cluster maintenance commands.

The `clsetup` utility enables you to add resources to the resource group interactively. Configuring these resources interactively reduces the possibility for configuration errors that might result from command syntax errors or omissions. The `clsetup` utility ensures that all required resources are created and that all required dependencies between resources are set.

Configuration Requirements for Highly Available Local File Systems

Any file system on multihost disks must be accessible from any host that is directly connected to those multihost disks. To meet this requirement, configure the highly available local file system as follows:

- Ensure that the disk partitions of the local file system reside on global devices.
- Set the `AffinityOn` extension property of the `HASStoragePlus` resource that specifies these global devices to `True`.

The `Zpools` extension property of the `HASStoragePlus` resource ignores the `AffinityOn` extension property.

- Create the `HASStoragePlus` resource in a failover resource group.

- Ensure that the failback settings for the device groups and the resource group that contains the HASstoragePlus resource are identical.

Note – The use of a volume manager with the global devices for a highly available local file system is optional.

Format of Device Names for Devices Without a Volume Manager

If you are not using a volume manager, use the appropriate format for the name of the underlying storage device. The format to use depends on the type of storage device as follows:

- For block devices: `/dev/global/dsk/dDsS`
- For raw devices: `/dev/global/rdisk/dDsS`

The replaceable elements in these device names are as follows:

- *D* is an integer that specifies the device ID (DID) instance number.
- *S* is an integer that specifies the slice number.

Sample Entries in `/etc/vfstab` for Highly Available Local File Systems

The following examples show entries in the `/etc/vfstab` file for global devices that are to be used for highly available local file systems.

Note – Solaris ZFS does not use the `/etc/vfstab` file.

EXAMPLE 2-35 Entries in `/etc/vfstab` for a Global Device Without a Volume Manager

This example shows entries in the `/etc/vfstab` file for a global device on a physical disk without a volume manager.

```
/dev/global/dsk/d1s0      /dev/global/rdisk/d1s0
/global/local-fs/nfs ufs   5 no    logging
```

EXAMPLE 2-36 Entries in `/etc/vfstab` for a Global Device With Solaris Volume Manager

This example shows entries in the `/etc/vfstab` file for a global device that uses Solaris Volume Manager.

```
/dev/md/kappa-1/dsk/d0    /dev/md/kappa-1/rdisk/d0
/global/local-fs/nfs ufs   5 no    logging
```

Note – The same file system entries must be added to the zone cluster configuration when you configure the file system for a zone cluster using the `SUNW.HASStoragePlus` resource type.

▼ How to Set Up the HASStoragePlus Resource Type by Using the `clsetup` Utility

The following instructions explain how to set up the `HASStoragePlus` resource type by using the `clsetup` utility. Perform this procedure from any cluster node.

This procedure provides the long forms of the Oracle Solaris Cluster maintenance commands. Most commands also have short forms. Except for the forms of the command names, the commands are identical.

Before You Begin Ensure that the following prerequisites are met:

- Ensure that the required volumes, disk groups, and file systems are created.

1 Assume the root role on any cluster node.

2 Start the `clsetup` utility.

```
# clsetup
```

The `clsetup` main menu is displayed.

3 Type the number for the option for data services and press Return.

The Data Services menu is displayed.

4 Type the number for the option for configuring highly available storage and press Return.

The `clsetup` utility provides the list of prerequisites for performing this task.

5 Verify that the prerequisites are met, and press Return to continue.

The `clsetup` utility provides a list of the cluster nodes that can master the highly available `HASStoragePlus` resource.

6 Select the nodes that can master the highly available `HASStoragePlus` resource.

- To accept the default selection of all listed nodes in an arbitrary order, type `a` and press Return.

- **To select a subset of the listed nodes, type a comma-separated or space-separated list of the numbers that correspond to the nodes and press Return.**

Ensure that the nodes are listed in the order in which the nodes are to appear in the HAS`StoragePlus` resource group's node list. The first node in the list is the primary node of this resource group.

- **To select all nodes in a particular order, type a comma-separated or space-separated ordered list of the numbers that correspond to the nodes and press Return.**

- 7 **To confirm your selection of nodes, type `d` and press Return.**

The `clsetup` utility provides a list of types of shared storage type where data is to be stored.

- 8 **Type the numbers that correspond to type of shared storage that you are using for storing the data and press Return.**

The `clsetup` utility provides a list of the file system mount points that are configured in the cluster. If there are no existing mount points, the `clsetup` utility allows you to define a new mount point.

- 9 **Specify the default mount directory, the raw device path, the `Global Mount` option and the `Check File System Periodically` option and press Return.**

The `clsetup` utility returns you the properties of the mount point that the utility will create.

- 10 **To create the mount point, type `d` and press Return.**

The `clsetup` utility provides the available file system mount points.

Note – You can use the `c` option to define another new mount point.

- 11 **Select the file system mount points.**

- **To accept the default selection of all listed file system mount points in an arbitrary order, type `a` and press Return.**
- **To select a subset of the listed file system mount points, type a comma-separated or space-separated list of the numbers that correspond to the file system mount points and press Return.**

- 12 **To confirm your selection of nodes, type `d` and press Return.**

The `clsetup` utility provides a list of the global disk sets and device groups that are configured in the cluster.

13 Select the global device groups.

- To accept the default selection of all listed device groups in an arbitrary order, type **a** and press Return.
- To select a subset of the listed device groups, type a comma-separated or space-separated list of the numbers that correspond to the device groups and press Return.

14 To confirm your selection of nodes, type **d and press Return.**

The `clsetup` utility provides the names of the Oracle Solaris Cluster objects that the utility will create.

15 If you require a different name for any Oracle Solaris Cluster object, change the name as follows.**a. Type the number for the name that you are changing and press Return.**

The `clsetup` utility provides a screen where you can specify the new name.

b. At the New Value prompt, type the new name and press Return.

The `clsetup` utility returns you to the list of the names of the Oracle Solaris Cluster objects that the utility will create.

16 To confirm your selection of Oracle Solaris Cluster object names, type **d and press Return.**

The `clsetup` utility provides information about the Oracle Solaris Cluster configuration that the utility will create.

17 To create the configuration, type **c and Press Return.**

The `clsetup` utility provides a progress message to indicate that the utility is running commands to create the configuration. When configuration is complete, the `clsetup` utility lists the commands that the utility ran to create the configuration.

18 (Optional) Type **q and press Return repeatedly until you quit the `clsetup` utility.**

If you prefer, you can leave the `clsetup` utility running while you perform other required tasks before using the utility again. If you choose to quit `clsetup`, the utility recognizes your existing resource group when you restart the utility.

19 Verify that the `HASStoragePlus` resource has been created.

Use the `clresource(1CL)` utility for this purpose.

```
# clresource show name_of_rg
```

▼ How to Set Up the HASStoragePlus Resource Type to Make File Systems Highly Available Other Than Solaris ZFS

The following procedure explains how to set up the HASStoragePlus resource type to make file systems other than Solaris ZFS highly available.

- 1 On any node in the global cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Create a failover resource group.

```
# clresourcegroup create resource-group
```
- 3 Register the HASStoragePlus resource type.

```
# clresourcetype register SUNW.HASStoragePlus
```
- 4 Create the HASStoragePlus resource and define the file system mount points.

```
# clresource create -g resource-group \  
-t SUNW.HASStoragePlus -p FileSystemMountPoints=mount-point-list hasp-resource
```
- 5 Bring online and in a managed state the resource group that contains the HASStoragePlus resource.

```
# clresourcegroup online -M resource-group
```

Example 2–37 Setting Up the HASStoragePlus Resource Type to Make a UFS File System Highly Available for the Global Cluster

This example assumes that the file system `/web-1` is configured to the HASStoragePlus resource to make the file system highly available for the global cluster.

```
phys-schost-1# vi /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot   options
#
# /dev/md/apachedg/dsk/d0 /dev/md/apachedg/rdisk/d0 /web-1 ufs 2 no logging
# clresourcegroup create hasp-rg
# clresourcetype register SUNW.HASStoragePlus
# clresource create -g hasp-rg -t SUNW.HASStoragePlus -p FileSystemMountPoints=/global/ufs-1 hasp-rs
# clresourcegroup online -M hasp-rg
```

Example 2–38 Setting Up the HASStoragePlus Resource Type to Make a UFS File System Highly Available for a Zone Cluster

This example assumes that the file system `/web-1` is configured to the HASStoragePlus resource to make the file system highly available for a zone cluster `sczone`. When a local file system is

configured as a highly available local file system for a zone cluster using the `SUNW.HAStoragePlus` resource type, the `HAStoragePlus` resource reads the file system information in the zone cluster configuration.

```
# clzonecluster configure sczone
clzc:sczone> add fs
clzc:sczone:fs> set dir=/web-1
clzc:sczone:fs> set special=/dev/md/apachedg/dsk/d0
clzc:sczone:fs> set raw=/dev/md/apachedg/rdisk/d0
clzc:sczone:fs> set type=ufs
clzc:sczone:fs> add options [logging]
clzc:sczone:fs> end
clzc:sczone:fs> exit
# clresourcegroup create -Z sczone hasp-rg
# clresourcetype register -Z sczone SUNW.HAStoragePlus
# clresource create -Z sczone -g hasp-rg \
-t SUNW.HAStoragePlus -p FileSystemMountPoints=/web-1 hasp-rs
# clresourcegroup online -Z sczone -M hasp-rg
```

▼ How to Set Up the `HAStoragePlus` Resource Type to Make a Local Solaris ZFS File System Highly Available

You perform the following primary tasks to make a local Solaris ZFS highly available:

- Create a ZFS storage pool.
- Create a ZFS file system in that ZFS storage pool.
- Set up the `HAStoragePlus` resource that manages the ZFS storage pool.

This section describes how to complete these tasks.

If you are planning to manually import a ZFS pool that is already managed by the cluster, ensure that the pool is not imported on multiple nodes. Importing a pool on multiple nodes can lead to problems. For more information, see [“Changing a ZFS Pool Configuration That is Managed by an `HAStoragePlus` Resource”](#) on page 133.

1 Create a ZFS storage pool.



Caution – Do not add a configured quorum device to a ZFS storage pool. When a configured quorum device is added to a storage pool, the disk is relabeled as an EFI disk, the quorum configuration information is lost, and the disk no longer provides a quorum vote to the cluster. After a disk is in a storage pool, you can configure that disk as a quorum device. Alternatively, you can unconfigure the disk, add it to the storage pool, then reconfigure the disk as a quorum device.

Observe the following requirements when you create a ZFS storage pool in an Oracle Solaris Cluster configuration:

- Ensure that all of the devices from which you create a ZFS storage pool are accessible from all nodes in the cluster. These nodes must be configured in the node list of the resource group to which the `HASStoragePlus` resource belongs.
- Ensure that the Oracle Solaris device identifier that you specify to the `zpool(1M)` command, for example `/dev/dsk/c0t0d0`, is visible to the `cldevice list -v` command.

Note – The ZFS storage pool can be created using a full disk or a disk slice. It is preferred to create a ZFS storage pool using a full disk by specifying an Oracle Solaris logical device as ZFS file system performs better by enabling the disk write cache. ZFS file system labels the disk with EFI when a full disk is provided.

See “[Creating a Basic ZFS Storage Pool](#)” in *Oracle Solaris 11.1 Administration: ZFS File Systems* for information about how to create a ZFS storage pool.

2 In the ZFS storage pool that you just created, create a ZFS file system.

Observe the following requirements when you create a ZFS file system in the ZFS pool:

- You can create more than one ZFS file system in the same ZFS storage pool.
- `HASStoragePlus` does not support file systems created on ZFS file system volumes.
- Do not place a ZFS file system in the `FilesystemMountPoints` extension property.
- If necessary, change the ZFS `failmode` property setting to either `continue` or `panic`, whichever best fits your requirements.

Note – The ZFS pool `failmode` property is set to `wait` by default. This setting can result in the `HASStoragePlus` resource blocking, which might prevent a failover of the resource group. See the `zpool(1M)` man page to understand the possible values for the `failmode` property and decide which value fits your requirements.

- You can choose to encrypt a ZFS file system when you create it. The `HASStoragePlus` resource automatically mounts all the file systems in the pool during resource online. The encrypted file system that requires interactive entry of a key or a passphrase during mount will experience a problem bringing the resource online. To avoid problems, do not use `keysource=raw | hex | passphrase, prompt|pkcs11:` for the encrypted file systems of the ZFS storage pool managed by a cluster using an `HASStoragePlus` resource. You can use `keysource=raw | hex | passphrase, file://|https://`, where the key or a passphrase location is accessible to the cluster nodes where the `HASStoragePlus` resource is going online.

See “[Creating a ZFS File System Hierarchy](#)” in *Solaris ZFS Administration Guide* for information about how to create a ZFS file system in a ZFS storage pool.

3 On any node in the cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

4 Create a failover resource group.

```
# clresourcegroup create resource-group
```

5 Register the HASStoragePlus resource type.

```
# clresourcetype register SUNW.HASStoragePlus
```

6 Create an HASStoragePlus resource for the local ZFS file system.

```
# clresource create -g resource-group -t SUNW.HASStoragePlus \
-p Zpools=zpool -p ZpoolsSearchDir=/dev/did/dsk \
resource
```

The default location to search for devices of ZFS storage pools is /dev/dsk. It can be overridden by using the ZpoolsSearchDir extension property.

The resource is created in the enabled state.

7 Bring online and in a managed state the resource group that contains the HASStoragePlus resource.

```
# clresourcegroup online -M resource-group
```

Example 2-39 Setting Up the HASStoragePlus Resource Type to Make a Local ZFS File System Highly Available for a Global Cluster

The following example shows the commands to make a local ZFS file system highly available.

```
phys-schost-1% su
Password:
# cldevice list -v
```

DID	Device	Full Device Path
-----	-----	-----
d1		phys-schost-1:/dev/rdisk/c0t0d0
d2		phys-schost-1:/dev/rdisk/c0t1d0
d3		phys-schost-1:/dev/rdisk/c1t8d0
d3		phys-schost-2:/dev/rdisk/c1t8d0
d4		phys-schost-1:/dev/rdisk/c1t9d0
d4		phys-schost-2:/dev/rdisk/c1t9d0
d5		phys-schost-1:/dev/rdisk/c1t10d0
d5		phys-schost-2:/dev/rdisk/c1t10d0
d6		phys-schost-1:/dev/rdisk/c1t11d0
d6		phys-schost-2:/dev/rdisk/c1t11d0
d7		phys-schost-2:/dev/rdisk/c0t0d0
d8		phys-schost-2:/dev/rdisk/c0t1d0

you can create a ZFS storage pool using a disk slice by specifying a Solaris device identifier:

```
# zpool create HAZpool c1t8d0s2
```

or you can create a ZFS storage pool using disk slice by specifying a logical device identifier

```
# zpool create HAZpool /dev/did/dsk/d3s2
```

```
# zfs create HAZpool/export
```

```
# zfs create HAZpool/export/home
```

```
# clresourcegroup create hasp-rg
```

```
# clresourcetype register SUNW.HASStoragePlus
```

```
# clresource create -g hasp-rg -t SUNW.HASStoragePlus -p Zpools=HAzpool hasp-rs
# clresourcegroup online -M hasp-rg
```

Example 2–40 Setting Up the HASStoragePlus Resource Type to Make a Local ZFS File System Highly Available for a Zone Cluster

The following example shows the steps to make a local ZFS file system highly available in a zone cluster *sczone*.

```
phys-schost-1# cldevice list -v
# zpool create HAzpool c1t8d0
# zfs create HAzpool/export
# zfs create HAzpool/export/home
# clzonecluster configure sczone
clzc:sczone> add dataset
clzc:sczone:fs> set name=HAzpool
clzc:sczone:fs> end
clzc:sczone:fs> exit
# clresourcegroup create -Z sczone hasp-rg
# clresource type register -Z sczone SUNW.HASStoragePlus
# clresource create -Z sczone -g hasp-rg -t SUNW.HASStoragePlus \
-p Zpools=HAzpool hasp-rs
# clresourcegroup online -Z -sczone -M hasp-rg
```

▼ How to Delete an HASStoragePlus Resource That Makes a Local Solaris ZFS Highly Available

- Disable and delete the HASStoragePlus resource that makes a local Solaris ZFS highly available.

```
# clresource delete -F -g resource-group -t SUNW.HASStoragePlus resource
```

Sharing a Highly Available Local File System Across Zone Clusters

You can use the SUNW.HASStoragePlus resource type to share a highly available local file system directory managed by a global cluster resource to a zone cluster. This method consolidates the storage and shares a highly available local file system with different applications running on different zone clusters. For information on adding a file system to a zone cluster, see [“Adding File Systems to a Zone Cluster”](#) in *Oracle Solaris Cluster Software Installation Guide*.

This section explains the requirements and procedures for sharing a highly available local file system directory across zone clusters.

Configuration Requirements for Sharing a Highly Available Local File System Directory to a Zone Cluster

The directory of a highly available local file system managed by a global cluster resource can be shared to a zone cluster. To share a highly available local file system directory, the configuration must meet the following requirements:

- Create an `HASStoragePlus` resource in a failover resource group in a global cluster with the file system where the directory to be shared belongs.
- The directory of the highly available local file system that you want to share must be configured to a zone cluster as an `lofs` file system.
- Create an `HASStoragePlus` resource in a failover resource group in a zone cluster with the `lofs` file system.
- The zone cluster resource must have an offline restart dependency on the global cluster resource.
- The zone cluster resource's resource group must have a strong positive affinity or strong positive affinity with failover delegation on the global cluster resource's resource group.

Note – The applications sharing a highly available local file system will experience an availability impact due to collocation of the applications. An application failure on a node and its intent to fail over might have a cascading effect on other applications and the applications would be forced to fail over to another node. Mitigate the problem by reducing the number of applications that are sharing the file system. If the file system that is being shared is UFS, you can choose to configure the cluster file system to a zone cluster. See [“How to Set Up the `HASStoragePlus` Resource for Cluster File Systems” on page 111](#).

▼ How to Set Up the `HASStoragePlus` Resource Type to Share a Highly Available Local File System Directory to a Zone Cluster

The following procedure explains how to set up the `HASStoragePlus` resource type to share a highly available local file system (for example, UFS) or a ZFS pool directory to a zone cluster called *zoneclustername*.

- 1 **On any node in the global cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**

Perform the steps from a node in the global cluster, because the dependencies and affinities from a zone cluster to a global cluster can only be set by an authorized cluster node administrator.

2 Create a failover resource group in the global cluster.

```
# clresourcegroup create gc-hasp-resource-group
```

3 Register the HASStoragePlus resource type in the global cluster.

```
# clresourcetype register SUNW.HASStoragePlus
```

4 Create an HASStoragePlus resource in a failover resource group of the global cluster with the highly available local file system that contains the directory that you want to share to a zone cluster.

```
# clresource create -g gc-hasp-resource-group -t HASStoragePlus \
-p FilesystemMountPoints=mount-point \
-p Zpools=pool gc-hasp-resource
```

5 Bring the global cluster failover resource group online.

```
# clresourcegroup online -M gc-hasp-resource-group
```

6 Configure the directory of the highly available local file system that is being shared to the zone cluster as an Lofs file system.

```
# clzonecluster configure zoneclustername
clzc:zoneclustername> add fs
clzc:zoneclustername:fs> set dir = shared-dir-mount-point-in-zc
clzc:zoneclustername:fs> set special = shared-directory
clzc:zoneclustername:fs> set type = lofs
clzc:zoneclustername:fs> end
clzc:zoneclustername> exit
#
```

7 Create a failover resource group in the zone cluster that has a strong positive affinity or strong positive affinity with failover delegation on the failover resource group of the global cluster.

```
# clresourcegroup create -Z zoneclustername \
-p RG_affinities=++global:gc-hasp-resource-group \
zc-hasp-resource-group
OR
# clresourcegroup create -Z zoneclustername \
-p RG_affinities=++global:gc-hasp-resource-group zc-hasp-resource-group
```

8 Register the HASStoragePlus resource type in the zone cluster.

```
# clresourcetype register -Z zoneclustername SUNW.HASStoragePlus
```

9 Create an HASStoragePlus resource in a failover resource group of the zone cluster. Configure the zone cluster with the Lofs file system for a shared directory with a dependency on the global cluster resource that you want to share to the zone cluster.

```
# clresource create -Z zoneclustername -t SUNW.HASStoragePlus -g zc-hasp-resource-group \
-p FilesystemMountPoints=shared-dir-mount-point-in-zc \
-p Resource_dependencies_offline_restart=global:gc-hasp-resource zc-hasp-resource
```

10 Bring the zone cluster failover resource group online.

```
# clresourcegroup online -Z zoneclustername -M zc-hasp-resource-group
```

Example 2–41 Setting Up the HAStoragePlus Resource Type to Share a UFS Highly Available Local File System Directory to a Zone Cluster

The following example shows how to share the `/local/fs/home` directory of a UFS highly available local file system (`/local/fs`) to a zone cluster called `sczone`.

```
# clresourcegroup create gc-hasp-rg
# clresourcetype register -Z sczone SUNW.HAStoragePlus
# vi /etc/vfstab /dev/md/dg1/dsk/d0 /dev/md/dg1/rdisk/d0 /local/fs ufs 2 no logging
# clresource create -g gc-hasp-rg -t SUNW.HAStoragePlus \
-p FilesystemMountPoints=/local/fs gc-hasp-rs
# clresourcegroup online -M gc-hasp-rg
```

The steps above ensure that the `gc-hasp-rs` resource running in the global cluster manages the highly available local file system `/local/fs`.

```
# clzonecluster configure sczone
clzc:sczone> add fs
clzc:sczone:fs> set dir = /share/local/fs/home
clzc:sczone:fs> set special = /local/fs/home
clzc:sczone:fs> set type = lofs
clzc:sczone:fs> end
clzc:sczone> exit
```

The configuration above makes the highly available local file system's directory `/local/fs/home` available in the zone cluster `sczone` at mount point `/share/local/fs/home`.

```
# clresourcegroup create -Z sczone \
-p RG_affinities==+global:gc-hasp-rg zc-hasp-rg
# clresourcetype register -Z sczone SUNW.HAStoragePlus
# clresource create -Z sczone -t HAStoragePlus -g zc-hasp-rg \
-p FilesystemMountPoints=/share/local/fs/home \
-p Resource_dependencies_offline_restart=global:gc-hasp-rs zc-hasp-rs
# clresourcegroup online -Z sczone -M zc-hasp-rg
```

The steps above create a zone cluster resource that manages the shared directory as an `lofs` file system.

Example 2–42 Setting Up the HAStoragePlus Resource Type to Share a ZFS Pool Directory to a Zone Cluster

The following example shows how to share the ZFS pool "tank" directory `/tank/home` to a zone cluster called `sczone`.

```
# clresourcegroup create gc-hasp-rg
# clresourcetype register SUNW.HAStoragePlus
# clresource create -g gc-hasp-rg -t SUNW.HAStoragePlus \
-p Zpools=tank gc-hasp-rs
# clresourcegroup online -M gc-hasp-rg
```

The steps above ensure that the ZFS highly available local file system is managed by `gc-hasp-rs` running in the global cluster.

```
# clzonecluster configure sczone
clzc:sczone> add fs
clzc:sczone:fs> set dir = /share/tank/home
clzc:sczone:fs> set special = /tank/home
clzc:sczone:fs> set type = lofs
clzc:sczone:fs>end
clzc:sczone> exit
#
```

The configuration above makes the ZFS pool "tank" directory `/tank/home` available in the zone cluster `sczone` at mountpoint `/share/tank/home`.

```
# clresourcegroup create -Z sczone \
-p RG_affinities==+global:gc-hasp-rg zc-hasp-rg
# clresourcetype register -Z sczone SUNW.HASStoragePlus
# clresource create -Z sczone -t HASStoragePlus -g zc-hasp-rg \
-p FilesystemMountPoints=/share/tank/home \
-p Resource_dependencies_offline_restart=global:gc-hasp-rs zc-hasp-rs
# clresourcegroup online -Z sczone -M zc-hasp-rg
```

The steps above create a zone cluster resource that manages the shared directory as an `lofs` file system.

Modifying Online the Resource for a Highly Available Local File System

You might need a highly available local file system to remain available while you are modifying the resource that represents the file system. For example, you might need the file system to remain available because storage is being provisioned dynamically. In this situation, modify the resource that represents the highly available local file system while the resource is online.

In the Oracle Solaris Cluster environment, a highly available local file system is represented by an `HASStoragePlus` resource. Oracle Solaris Cluster enables you to modify an online `HASStoragePlus` resource as follows:

- Adding file systems to the `HASStoragePlus` resource
- Removing file systems from the `HASStoragePlus` resource

Oracle Solaris Cluster software does not enable you to rename a file system while the file system is online.

Note – When you remove the file systems configured in the `HASStoragePlus` resources for a zone cluster, you also need to remove the file system configuration from the zone cluster. For information about removing a file system from a zone cluster, see [“How to Remove a File System From a Zone Cluster”](#) in *Oracle Solaris Cluster System Administration Guide*.

▼ How to Add File Systems Other Than Solaris ZFS to an Online HAStoragePlus Resource

When you add a local or cluster file system to an HAStoragePlus resource, the HAStoragePlus resource automatically mounts the file system.

- 1 **On one node of the cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**
- 2 **In the `/etc/vfstab` file on each node of the cluster, add an entry for the mount point of each file system that you are adding.**

For each entry, set the mount at boot field and the mount options field as follows:

- For local file systems
 - Set the mount at boot field to `no`.
 - Remove the `global` flag.
- For cluster file systems
 - If the file system is a cluster file system, set the mount options field to contain the `global` option.

- 3 **Retrieve the list of mount points for the file systems that the HAStoragePlus resource already manages.**

```
# scha_resource_get -O extension -R hasp-resource -G hasp-rg FileSystemMountPoints
-R hasp-resource           Specifies the HAStoragePlus resource to which you are adding
                           file systems
-G hasp-rg                 Specifies the resource group that contains the HAStoragePlus
                           resource
```

- 4 **Modify the `FileSystemMountPoints` extension property of the HAStoragePlus resource to contain the following mount points:**

- The mount points of the file systems that the HAStoragePlus resource already manages
- The mount points of the file systems that you are adding to the HAStoragePlus resource

```
# clresource set -p FileSystemMountPoints="mount-point-list" hasp-resource
-p FileSystemMountPoints="mount-point-list"
  Specifies a comma-separated list of mount points of the file systems that the HAStoragePlus
  resource already manages and the mount points of the file systems that you are adding. The
  format of each entry in the list is LocalZonePath:GlobalZonePath. In this format, the global
  path is optional. If the global path is not specified, the global path is the same as the local
  path.
```

hasp-resource

Specifies the HASStoragePlus resource to which you are adding file systems.

5 Confirm that you have a match between the mount point list of the HASStoragePlus resource and the list that you specified in Step 4.

```
# scha_resource_get -O extension -R hasp-resource -G hasp-rg \
  FileSystemMountPoints
```

-R *hasp-resource* Specifies the HASStoragePlus resource to which you are adding file systems.

-G *hasp-rg* Specifies the resource group that contains the HASStoragePlus resource.

6 Confirm that the HASStoragePlus resource is online and not faulted.

If the HASStoragePlus resource is online and faulted, validation of the resource succeeded, but an attempt by HASStoragePlus to mount a file system failed.

```
# clresource status hasp-resource
```

Example 2–43 Adding a File System to an Online HASStoragePlus Resource

This example shows how to add a file system to an online HASStoragePlus resource.

- The HASStoragePlus resource is named *rshasp* and is contained in the resource group *rghasp*.
- The HASStoragePlus resource named *rshasp* already manages the file system whose mount point is */global/global-fs/fs*.
- The mount point of the file system that is to be added is */global/local-fs/fs*.

The example assumes that the */etc/vfstab* file on each cluster node already contains an entry for the file system that is to be added.

```
# scha_resource_get -O extension -R rshasp -G rghasp FileSystemMountPoints
STRINGARRAY
/global/global-fs/fs
# clresource set \
-p FileSystemMountPoints="/global/global-fs/fs,/global/local-fs/fs"
# scha_resource_get -O extension -R rshasp -G rghasp FileSystemMountPoints rshasp
STRINGARRAY
/global/global-fs/fs
/global/local-fs/fs
# clresource status rshasp
```

=== Cluster Resources ===

Resource Name	Node Name	Status	Message
rshasp	node46	Offline	Offline
	node47	Online	Online

▼ How to Remove File Systems Other Than Solaris ZFS From an Online HAStoragePlus Resource

When you remove a file system from an HAStoragePlus resource, the HAStoragePlus resource treats a local file system differently from a cluster file system.

- The HAStoragePlus resource automatically unmounts a local file system.
- The HAStoragePlus resource does not unmount the cluster file system.



Caution – Before removing a file system from an online HAStoragePlus resource, ensure that no applications are using the file system. When you remove a file system from an online HAStoragePlus resource, the file system might be forcibly unmounted. If a file system that an application is using is forcibly unmounted, the application might fail or hang.

- 1 **On one node of the cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**

- 2 **Retrieve the list of mount points for the file systems that the HAStoragePlus resource already manages.**

```
# scha_resource_get -O extension -R hasp-resource -G hasp-rg FileSystemMountPoints
-R hasp-resource    Specifies the HAStoragePlus resource from which you are removing file
                    systems.
-G hasp-rg          Specifies the resource group that contains the HAStoragePlus resource.
```

- 3 **Modify the `FileSystemMountPoints` extension property of the HAStoragePlus resource to contain *only* the mount points of the file systems that are to remain in the HAStoragePlus resource.**

```
# clresource set -p FileSystemMountPoints="mount-point-list" hasp-resource
-p FileSystemMountPoints="mount-point-list"
  Specifies a comma-separated list of mount points of the file systems that are to remain in the
  HAStoragePlus resource. This list must not include the mount points of the file systems that
  you are removing.

hasp-resource
  Specifies the HAStoragePlus resource from which you are removing file systems.
```

- 4 **Confirm that you have a match between the mount point list of the HAStoragePlus resource and the list that you specified in [Step 3](#).**

```
# scha_resource_get -O extension -R hasp-resource -G hasp-rg \
FileSystemMountPoints
```

- R *hasp-resource* Specifies the HASStoragePlus resource from which you are removing file systems.
- G *hasp-rg* Specifies the resource group that contains the HASStoragePlus resource.

5 Confirm that the HASStoragePlus resource is online and not faulted.

If the HASStoragePlus resource is online and faulted, validation of the resource succeeded, but an attempt by HASStoragePlus to unmount a file system failed.

```
# clresource status hasp-resource
```

6 (Optional) From the /etc/vfstab file on each node of the cluster, remove the entry for the mount point of each file system that you are removing.

Example 2-44 Removing a File System From an Online HASStoragePlus Resource

This example shows how to remove a file system from an online HASStoragePlus resource.

- The HASStoragePlus resource is named *rshasp* and is contained in the resource group *rghasp*.
- The HASStoragePlus resource named *rshasp* already manages the file systems whose mount points are as follows:
 - /global/global-fs/fs
 - /global/local-fs/fs
- The mount point of the file system that is to be removed is /global/local-fs/fs.

```
# scha_resource_get -O extension -R rshasp -G rghasp FileSystemMountPoints
STRINGARRAY
/global/global-fs/fs
/global/local-fs/fs
# clresource set -p FileSystemMountPoints="/global/global-fs/fs"
# scha_resource_get -O extension -R rshasp -G rghasp FileSystemMountPoints rshasp
STRINGARRAY
/global/global-fs/fs
# clresource status rshasp
```

=== Cluster Resources ===

Resource Name	Node Name	Status	Message
rshasp	node46	Offline	Offline
	node47	Online	Online

▼ How to Add a Solaris ZFS Storage Pool to an Online HAStoragePlus Resource

When you add a Solaris ZFS storage pool to an online HAStoragePlus resource, the HAStoragePlus resource does the following:

- Imports the ZFS storage pool.
- Mounts all file systems in the ZFS storage pool.



Caution – If you are planning to manually import a pool that is already managed by the cluster, ensure that the pool is not imported on multiple nodes. Importing a pool on multiple nodes can lead to problems.

If you want to make configuration changes to a ZFS pool that is managed by cluster with an HAStoragePlus resource, see [“Changing a ZFS Pool Configuration That is Managed by an HAStoragePlus Resource”](#) on page 133.

- 1 **On any node in the cluster, assume the root role that provides `solaris.cluster.modify` RBAC authorization.**

- 2 **Determine the ZFS storage pools that the HAStoragePlus resource already manages.**

```
# clresource show -g hasp-resource-group -p Zpools hasp-resource
```

-g hasp-resource-group Specifies the resource group that contains the HAStoragePlus resource.

hasp-resource Specifies the HAStoragePlus resource to which you are adding the ZFS storage pool.

- 3 **Add the new ZFS storage pool to the existing list of ZFS storage pools that the HAStoragePlus resource already manages.**

```
# clresource set -p Zpools="zpools-list" hasp-resource
```

*-p Zpools="*zpools-list*"* Specifies a comma-separated list of existing ZFS storage pool names that the HAStoragePlus resource already manages and the new ZFS storage pool name that you want to add.

hasp-resource Specifies the HAStoragePlus resource to which you are adding the ZFS storage pool.

- 4 **Compare the new list of ZFS storage pools that the HAStoragePlus resource manages with the list that you generated in [Step 2](#).**

```
# clresource show -g hasp-resource-group -p Zpools hasp-resource
```

`-g hasp-resource-group` Specifies the resource group that contains the HASStoragePlus resource.

`hasp-resource` Specifies the HASStoragePlus resource to which you added the ZFS storage pool.

5 Confirm that the HASStoragePlus resource is online and not faulted.

If the HASStoragePlus resource is online but faulted, validation of the resource succeeded. However, an attempt by the HASStoragePlus resource to import and mount the ZFS file system failed. In this case, you need to repeat the preceding set of steps.

```
# clresourcegroup status hasp-resource
```

▼ How to Remove a Solaris ZFS Storage Pool From an Online HASStoragePlus Resource

When you remove a Solaris ZFS storage pool from an online HASStoragePlus resource, the HASStoragePlus resource does the following:

- Unmounts the file systems in the ZFS storage pool.
- Exports the ZFS storage pool from the node.

1 On any node in the cluster, assume the root role that provides solaris.cluster.modify RBAC authorization.

2 Determine the ZFS storage pools that the HASStoragePlus resource already manages.

```
# clresource show -g hasp-resource-group -p Zpools hasp-resource
```

`-g hasp-resource-group` Specifies the resource group that contains the HASStoragePlus resource.

`hasp-resource` Specifies the HASStoragePlus resource from which you are removing the ZFS storage pool.

3 Remove the ZFS storage pool from the list of ZFS storage pools that the HASStoragePlus resource currently manages.

```
# clresource set -p Zpools="zpools-list" hasp-resource
```

`-p Zpools="zpools-list"` Specifies a comma-separated list of ZFS storage pool names that the HASStoragePlus resource currently manages, minus the ZFS storage pool name that you want to remove.

`hasp-resource` Specifies the HASStoragePlus resource from which you are removing the ZFS storage pool.

- 4 Compare the new list of ZFS storage pools that the `HASStoragePlus` resource now manages with the list that you generated in [Step 2](#).

```
# clresource show -g hasp-resource-group -p Zpools hasp-resource
```

`-g hasp-resource-group` Specifies the resource group that contains the `HASStoragePlus` resource.

`hasp-resource` Specifies the `HASStoragePlus` resource from which you removed the ZFS storage pool.

- 5 Confirm that the `HASStoragePlus` resource is online and not faulted.

If the `HASStoragePlus` resource is online but faulted, validation of the resource succeeded. However, an attempt by the `HASStoragePlus` resource to unmount and export the ZFS file system failed. In this case, you need to repeat the preceding set of steps.

```
# clresource status -t SUNW.HASStoragePlus +
```

Changing a ZFS Pool Configuration That is Managed by an `HASStoragePlus` Resource

To change the ZFS pool configuration that is managed by `HASStoragePlus` resource, you must ensure that the pool is never imported on multiple nodes. Performing imports on multiple nodes can have severe consequences and could cause ZFS pool corruption.

The following procedures help you avoid multiple imports when performing pool configuration changes.

- [“How to Change a ZFS Pool Configuration That is Managed by an `HASStoragePlus` Resource in an Offline State” on page 133](#)
- [“How to Change a ZFS Pool Configuration That is Managed by an Online `HASStoragePlus` Resource” on page 134](#)

▼ How to Change a ZFS Pool Configuration That is Managed by an `HASStoragePlus` Resource in an Offline State

- 1 Ensure that the ZFS pool that requires configuration changes is not imported on any node.

```
# zpool list zfs-pool-name
```

Run this command on all cluster nodes that have a physical connection to the ZFS pool.

- 2 **Import the pool on the alternate root without using the force option on a cluster node that has a physical connection to the ZFS pool.**

```
# zpool import -R zfs-pool-name
```

If the import succeeds, proceed to [Step 3](#). If the import fails, the cluster node that previously accessed the pool might have shut down without exporting the pool. Follow the substeps below to ensure that the cluster node is not using the ZFS pool and then import the pool forcefully:

- a. **Check if the import failed due to an error message similar to the one below. If it did, proceed to [Step b](#) and [Step c](#):**

Cannot import 'zfs-pool-name': pool may be in use from other system, it was last accessed by hostname (hostid: hostid) on accessed-date.

- b. **Verify that the pool is not in use on the machine that last accessed it.**

```
hostname# zpool list zfs-pool-name
```

- c. **If the ZFS pool is not in use on that node, import the pool forcefully.**

```
# zpool import -f zfs-pool-name
```

- 3 **Perform the ZFS pool configuration changes.**
- 4 **Export the ZFS pool and check that the pool is not in use.**

```
# zpool export zfs-pool-name
# zpool list zfs-pool-name
```

▼ How to Change a ZFS Pool Configuration That is Managed by an Online HAStoragePlus Resource

- 1 **Find the cluster node where the ZFS pool is imported.**

It will be the node where the HAStoragePlus resource is online.

```
# clresource show hasp-rs-managing-pool
```

```
=== Cluster Resources ===
Resource Name      Node Name          Status      Message
-----
hasp-rs-managing-pool  phys-node-1      Offline    Offline
                        phys-node-2      Online     Online
```

```
phys-node-2# zpool list zfs-pool-name
```

- 2 **Perform the ZFS pool configuration changes.**

▼ How to Recover From a Fault After Modifying the FileSystemMountPoints Property of an HASStoragePlus Resource

If a fault occurs during a modification of the `FileSystemMountPoints` extension property, the status of the `HASStoragePlus` resource is `online` and `faulted`. After the fault is corrected, the status of the `HASStoragePlus` resource is `online`.

1 Determine the fault that caused the attempted modification to fail.

```
# clresource status hasp-resource
```

The status message of the faulty `HASStoragePlus` resource indicates the fault. Possible faults are as follows:

- The device on which the file system should reside does not exist.
- An attempt by the `fsck` command to repair a file system failed.
- The mount point of a file system that you attempted to add does not exist.
- A file system that you attempted to add cannot be mounted.
- A file system that you attempted to remove cannot be unmounted.

2 Correct the fault that caused the attempted modification to fail.

3 Repeat the step to modify the `FileSystemMountPoints` extension property of the `HASStoragePlus` resource.

```
# clresource set -p FileSystemMountPoints="mount-point-list" hasp-resource
```

```
-p FileSystemMountPoints="mount-point-list"
```

Specifies a comma-separated list of mount points that you specified in the unsuccessful attempt to modify the highly available local file system

```
hasp-resource
```

Specifies the `HASStoragePlus` resource that you are modifying

4 Confirm that the `HASStoragePlus` resource is `online` and not `faulted`.

```
# clresource status
```

Example 2-45 Status of a Faulty `HASStoragePlus` Resource

This example shows the status of a faulty `HASStoragePlus` resource. This resource is `faulty` because an attempt by the `fsck` command to repair a file system failed.

```
# clresource status
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	Status	Status Message
rshasp	node46	Offline	Offline
	node47	Online	Online Faulted - Failed to fsck: /mnt.

▼ How to Recover From a Fault After Modifying the Zpools Property of an HASStoragePlus Resource

If a fault occurs during a modification of the Zpools extension property, the status of the HASStoragePlus resource is online and faulted. After the fault is corrected, the status of the HASStoragePlus resource is online.

1 Determine the fault that caused the attempted modification to fail.

```
# clresource status hasp-resource
```

The status message of the faulty HASStoragePlus resource indicates the fault. Possible faults are as follows:

- The ZFS pool *zpool* failed to import.
- The ZFS pool *zpool* failed to export.

Note – If you import a corrupt ZFS pool, the best option is to choose `Continue` to display an error message. Other choices are `Wait` (which hangs until success occurs or the node panics) or `Panic` (which panics the node).

2 Correct the fault that caused the attempted modification to fail.

3 Repeat the step to modify the Zpools extension property of the HASStoragePlus resource.

```
# clresource set -p Zpools="zpools-list" hasp-resource
```

```
-p Zpools="zpools-list"    Specifies a comma-separated list of ZFS storage pool names that the
                           HASStoragePlus currently manages, minus the ZFS storage pool
                           name that you want to remove.
```

```
hasp-resource            Specifies the HASStoragePlus resource that you are modifying
```

4 Confirm that the HASStoragePlus resource is online and not faulted.

```
# clresource status
```

Example 2-46 Status of a Faulty HAStoragePlus Resource

This example shows the status of a faulty HAStoragePlus resource. This resource is faulty because the ZFS pool *zpool* failed to import.

```
# clresource status hasp-resource

=== Cluster Resources ===

Resource Name      Node Name      Status      Status Message
-----
hasp-resource      node46         OnLine      Faulted - Failed to import:hazpool
                   node47         OffLine     Offline
```

Changing the Cluster File System to a Local File System in an HAStoragePlus Resource

You can change the file system of an HAStoragePlus resource from a cluster file system to a local file system.

▼ How to Change the Cluster File System to Local File System in an HAStoragePlus Resource

- 1 Bring the failover resource group offline.


```
# clresourcegroup offline resource-group
```
- 2 Display the HAStoragePlus resource.


```
# clresource show -g resource-group -t SUNW.HAStoragePlus
```
- 3 Retrieve the list of mount points for each resource.


```
# clresource show -p FilesystemMountPoints hastorageplus-resource
```
- 4 Unmount the cluster file system.


```
# umount mount-points
```
- 5 Modify the `/etc/vfstab` entry of the mount points on all the nodes configured in the node list of the resource group.
 - Remove the `global` keyword from the mount options.
 - Modify the `mount at boot` option from `yes` to `no`.

Repeat the steps for all the cluster file systems of all the HASStoragePlus resources configured in the resource group.

6 Bring the resource group online.

```
# clresourcegroup online -eM resource-group
```

Upgrading the HASStoragePlus Resource Type

The HASStoragePlus resource type enables you to modify highly available local file systems online. Upgrade the HASStoragePlus resource type if all conditions in the following list apply:

- You are upgrading from an earlier version of Oracle Solaris Cluster.
- You need to use the new features of the HASStoragePlus resource type.

For general instructions that explain how to upgrade a resource type, see [“Upgrading a Resource Type” on page 33](#). The information that you need to complete the upgrade of the HASStoragePlus resource type is provided in the subsections that follow.

Information for Registering the New Resource Type Version

To determine the version of the resource type that is registered, use a command from the following list:

- The `cluster show` command lists the name and version of the cluster's resource types.
- The `clresourcetype list -v` command lists the node list of each resource type.

The RTR file for this resource type is `/usr/cluster/lib/rgm/rtreg/SUNW.HASStoragePlus`.

Information for Migrating Existing Instances of the Resource Type

Use the following information to migrate instances of the HASStoragePlus resource type:

- You can perform the migration to version 9 when the resource fault monitor is disabled. Enable the fault monitor after the migration is complete.
- You can perform the migration to any version other than 9 at any time.
- Set the `Type_version` property to the version of the features you intend to use.

- Ensure that the application has the `resource_dependencies_offline_restart` property set to the `HASStoragePlus` resource that it depends on. This is required for version 9 of the `HASStoragePlus` resource type, but it is optional for other versions of the resource type. The application must have either the `resource_dependencies` property or the `resource_dependencies_offline_restart` property set to the `HASStoragePlus` resource that it depends on if the `HASStoragePlus` version is older than 9.

Distributing Online Resource Groups Among Cluster Nodes

For maximum availability or optimum performance, some combinations of services require a specific distribution of online resource groups among cluster nodes. Distributing online resource groups involves creating affinities between resource groups for the following purposes:

- Enforcing the required distribution when the resource groups are first brought online
- Preserving the required distribution after an attempt to fail over or switch over a resource group

This section provides the following examples of how to use resource group affinities to distribute online resource groups among cluster nodes:

- Enforcing collocation of a resource group with another resource group
- Specifying a preferred collocation of a resource group with another resource group
- Balancing the load of a set of resource groups
- Specifying that a critical service has precedence
- Delegating the failover or switchover of a resource group
- Combining affinities between resource groups to specify more complex behavior

Resource Group Affinities

An affinity between resource groups restricts on which nodes the resource groups may be brought online simultaneously. In each affinity, a source resource group declares an affinity for a target resource group or several target resource groups. To create an affinity between resource groups, set the `RG_affinities` resource group property of the source as follows:

-p `RG_affinities=affinity-list`

affinity-list Specifies a comma-separated list of affinities between the source resource group and a target resource group or several target resource groups. You may specify a single affinity or more than one affinity in the list.

Specify each affinity in the list as follows:

operator target-rg

Note – Do not include a space between *operator* and *target-rg*.

operator Specifies the type of affinity that you are creating. For more information, see [Table 2–3](#).

target-rg Specifies the resource group that is the target of the affinity that you are creating.

TABLE 2–3 Types of Affinities Between Resource Groups

Operator	Affinity Type	Effect
+	Weak positive	If possible, the source is brought online on a node or on nodes where the target is online or starting. However, the source and the target are allowed to be online on different nodes.
++	Strong positive	The source is brought online only on a node or on nodes where the target is online or starting. The source and the target are <i>not</i> allowed to be online on different nodes.
-	Weak negative	If possible, the source is brought online on a node or on nodes where the target is <i>not</i> online or starting. However, the source and the target are allowed to be online on the same node.
--	Strong negative	The source is brought online only on a node or on nodes where the target is not online. The source and the target are <i>not</i> allowed to be online on the same node.
+++	Strong positive with failover delegation	Same as strong positive, except that an attempt by the source to fail over is delegated to the target. For more information, see “Delegating the Failover or Switchover of a Resource Group” on page 145.

Weak affinities take precedence over `NodeList` preference ordering.

The current state of other resource groups might prevent a strong affinity from being satisfied on any node. In this situation, the resource group that is the source of the affinity remains offline. If other resource groups' states change to enable the strong affinities to be satisfied, the resource group that is the source of the affinity comes back online.

Note – Use caution when declaring a strong affinity on a source resource group for more than one target resource group. If all declared strong affinities cannot be satisfied, the source resource group remains offline.

Enforcing Collocation of a Resource Group With Another Resource Group

A service that is represented by one resource group might depend so strongly on a service in a second resource group that both services must run on the same node. For example, an application that is comprised of multiple interdependent service daemons might require that all daemons run on the same node.

In this situation, force the resource group of the dependent service to be collocated with the resource group of the other service. To enforce collocation of a resource group with another resource group, declare on the resource group a strong positive affinity for the other resource group.

```
# clresourcegroup set|create -p RG_affinities=++target-rg source-rg
```

source-rg

Specifies the resource group that is the source of the strong positive affinity. This resource group is the resource group *on* which you are declaring a strong positive affinity for another resource group.

```
-p RG_affinities=++target-rg
```

Specifies the resource group that is the target of the strong positive affinity. This resource group is the resource group *for* which you are declaring a strong positive affinity.

A resource group follows the resource group for which it has a strong positive affinity. If the target resource group is relocated to a different node, the source resource group automatically switches to the same node as the target. However, a resource group that declares a strong positive affinity is prevented from failing over to a node on which the target of the affinity is not already running.

Note – Only failovers that are initiated by a resource monitor are prevented. If a node on which the source resource group and target resource group are running fails, both resource groups fail over to the same surviving node.

For example, a resource group `rg1` declares a strong positive affinity for resource group `rg2`. If `rg2` fails over to another node, `rg1` also fails over to that node. This failover occurs even if all the resources in `rg1` are operational. However, if a resource in `rg1` attempts to fail over `rg1` to a node where `rg2` is not running, this attempt is blocked.

The source of a strong positive affinity might be offline on all nodes when you bring online the target of the strong positive affinity. In this situation, the source of the strong positive affinity is automatically brought online on the same node as the target.

For example, a resource group `rg1` declares a strong positive affinity for resource group `rg2`. Both resource groups are initially offline on all nodes. If an administrator brings online `rg2` on a node, `rg1` is automatically brought online on the same node.

You can use the `clresourcegroup suspend` command to prevent a resource group from being brought online automatically due to strong affinities or cluster reconfiguration.

If you require a resource group that declares a strong positive affinity to be allowed to fail over, you must delegate the failover. For more information, see [“Delegating the Failover or Switchover of a Resource Group” on page 145](#).

EXAMPLE 2-47 Enforcing Collocation of a Resource Group With Another Resource Group

This example shows the command for modifying resource group `rg1` to declare a strong positive affinity for resource group `rg2`. As a result of this affinity relationship, `rg1` is brought online only on nodes where `rg2` is running. This example assumes that both resource groups exist.

```
# clresourcegroup set -p RG_affinities=++rg2 rg1
```

Specifying a Preferred Collocation of a Resource Group With Another Resource Group

A service that is represented by one resource group might use a service in a second resource group. As a result, these services run most efficiently if they run on the same node. For example, an application that uses a database runs most efficiently if the application and the database run on the same node. However, the services can run on different nodes because the reduction in efficiency is less disruptive than additional failovers of resource groups.

In this situation, specify that both resource groups should be collocated if possible. To specify preferred collocation of a resource group with another resource group, declare on the resource group a weak positive affinity for the other resource group.

```
# clresourcegroup set|create -p RG_affinities=+target-rg source-rg
```

source-rg

Specifies the resource group that is the source of the weak positive affinity. This resource group is the resource group *on* which you are declaring a weak positive affinity for another resource group.

```
-p RG_affinities=+target-rg
```

Specifies the resource group that is the target of the weak positive affinity. This resource group is the resource group *for* which you are declaring a weak positive affinity.

By declaring a weak positive affinity on one resource group for another resource group, you increase the probability of both resource groups running on the same node. The source of a weak positive affinity is first brought online on a node where the target of the weak positive

affinity is already running. However, the source of a weak positive affinity does not fail over if a resource monitor causes the target of the affinity to fail over. Similarly, the source of a weak positive affinity does not fail over if the target of the affinity is switched over. In both situations, the source remains online on the node where the source is already running.

Note – If a node on which the source resource group and target resource group are running fails, both resource groups are restarted on the same surviving node.

EXAMPLE 2-48 Specifying a Preferred Collocation of a Resource Group With Another Resource Group

This example shows the command for modifying resource group `rg1` to declare a weak positive affinity for resource group `rg2`. As a result of this affinity relationship, `rg1` and `rg2` are first brought online on the same node. But if a resource in `rg2` causes `rg2` to fail over, `rg1` remains online on the node where the resource groups were first brought online. This example assumes that both resource groups exist.

```
# clresourcegroup set -p RG_affinities=+rg2 rg1
```

Distributing a Set of Resource Groups Evenly Among Cluster Nodes

Each resource group in a set of resource groups might impose the same load on the cluster. In this situation, by distributing the resource groups evenly among cluster nodes, you can balance the load on the cluster.

To distribute a set of resource groups evenly among cluster nodes, declare on each resource group a weak negative affinity for the other resource groups in the set.

```
# clresourcegroup set|create -p RG_affinities=neg-affinity-list source-rg
source-rg
```

Specifies the resource group that is the source of the weak negative affinity. This resource group is the resource group *on* which you are declaring a weak negative affinity for other resource groups.

```
-p RG_affinities=neg-affinity-list
```

Specifies a comma-separated list of weak negative affinities between the source resource group and the resource groups that are the target of the weak negative affinity. The target resource groups are the resource groups *for* which you are declaring a weak negative affinity.

By declaring a weak negative affinity on one resource group for other resource groups, you ensure that a resource group is always brought online on the most lightly loaded node in the cluster. The fewest other resource groups are running on that node. Therefore, the smallest number of weak negative affinities are violated.

EXAMPLE 2-49 Distributing a Set of Resource Groups Evenly Among Cluster Nodes

This example shows the commands for modifying resource groups `rg1`, `rg2`, `rg3`, and `rg4` to ensure that these resource groups are evenly distributed among the available nodes in the cluster. This example assumes that resource groups `rg1`, `rg2`, `rg3`, and `rg4` exist.

```
# clresourcegroup set -p RG_affinities=-rg2,-rg3,-rg4 rg1
# clresourcegroup set -p RG_affinities=-rg1,-rg3,-rg4 rg2
# clresourcegroup set -p RG_affinities=-rg1,-rg2,-rg4 rg3
# clresourcegroup set -p RG_affinities=-rg1,-rg2,-rg3 rg4
```

Specifying That a Critical Service Has Precedence

A cluster might be configured to run a combination of mission-critical services and noncritical services. For example, a database that supports a critical customer service might run in the same cluster as noncritical research tasks.

To ensure that the noncritical services do not affect the performance of the critical service, specify that the critical service has precedence. By specifying that the critical service has precedence, you prevent noncritical services from running on the same node as the critical service.

When all nodes are operational, the critical service runs on a different node from the noncritical services. However, a failure of the critical service might cause the service to fail over to a node where the noncritical services are running. In this situation, the noncritical services are taken offline immediately to ensure that the computing resources of the node are fully dedicated to the mission-critical service.

To specify that a critical service has precedence, declare on the resource group of each noncritical service a strong negative affinity for the resource group that contains the critical service.

```
# clresourcegroup set|create -p RG_affinities=-critical-rg noncritical-rg
noncritical-rg
```

Specifies the resource group that contains a noncritical service. This resource group is the resource group *on* which you are declaring a strong negative affinity for another resource group.

```
-p RG_affinities=-critical-rg
```

Specifies the resource group that contains the critical service. This resource group is the resource group *for* which you are declaring a strong negative affinity.

A resource group moves away from a resource group for which it has a strong negative affinity.

The source of a strong negative affinity might be offline on all nodes when you take offline the target of the strong negative affinity. In this situation, the source of the strong negative affinity is

automatically brought online. In general, the resource group is brought online on the most preferred node, based on the order of the nodes in the node list and the declared affinities.

For example, a resource group `rg1` declares a strong negative affinity for resource group `rg2`. Resource group `rg1` is initially offline on all nodes, while resource group `rg2` is online on a node. If an administrator takes offline `rg2`, `rg1` is automatically brought online.

You can use the `clresourcegroup suspend` command to prevent the source of a strong negative affinity from being brought online automatically due to strong affinities or cluster reconfiguration.

EXAMPLE 2-50 Specifying That a Critical Service Has Precedence

This example shows the commands for modifying the noncritical resource groups `ncrg1` and `ncrg2` to ensure that the critical resource group `mcdbrg` has precedence over these resource groups. This example assumes that resource groups `mcdbrg`, `ncrg1`, and `ncrg2` exist.

```
# clresourcegroup set -p RG_affinities=--mcdbrg ncrng1 ncrng2
```

Delegating the Failover or Switchover of a Resource Group

The source resource group of a strong positive affinity cannot fail over or be switched over to a node where the target of the affinity is not running. If you require the source resource group of a strong positive affinity to be allowed to fail over or be switched over, you must delegate the failover to the target resource group. When the target of the affinity fails over, the source of the affinity is forced to fail over with the target.

Note – You might need to switch over the source resource group of a strong positive affinity that is specified by the `++` operator. In this situation, switch over the target of the affinity and the source of the affinity at the same time.

To delegate failover or switchover of a resource group to another resource group, declare on the resource group a strong positive affinity with failover delegation for the other resource group.

```
# clresourcegroup set|create source-rg -p RG_affinities=+++target-rg
```

source-rg

Specifies the resource group that is delegating failover or switchover. This resource group is the resource group *on* which you are declaring a strong positive affinity with failover delegation for another resource group.

`-p RG_affinities=+++target-rg`

Specifies the resource group to which *source-rg* delegates failover or switchover. This resource group is the resource group *for* which you are declaring a strong positive affinity with failover delegation.

A resource group may declare a strong positive affinity with failover delegation for at most one resource group. However, a given resource group may be the target of strong positive affinities with failover delegation that are declared by any number of other resource groups.

A strong positive affinity with failover delegation is not fully symmetric. The target can come online while the source remains offline. However, if the target is offline, the source cannot come online.

If the target declares a strong positive affinity with failover delegation for a third resource group, failover or switchover is further delegated to the third resource group. The third resource group performs the failover or switchover, forcing the other resource groups to fail over or be switched over also.

EXAMPLE 2-51 Delegating the Failover or Switchover of a Resource Group

This example shows the command for modifying resource group `rg1` to declare a strong positive affinity with failover delegation for resource group `rg2`. As a result of this affinity relationship, `rg1` delegates failover or switchover to `rg2`. This example assumes that both resource groups exist.

```
# clresourcegroup set -p RG_affinities=+++rg2 rg1
```

Combining Affinities Between Resource Groups

You can create more complex behaviors by combining multiple affinities. For example, the state of an application might be recorded by a related replica server. The node selection requirements for this example are as follows:

- The replica server must run on a different node from the application.
- If the application fails over from its current node, the application should fail over to the node where the replica server is running.
- If the application fails over to the node where the replica server is running, the replica server must fail over to a different node. If no other node is available, the replica server must go offline.

You can satisfy these requirements by configuring resource groups for the application and the replica server as follows:

- The resource group that contains the application declares a weak positive affinity for the resource group that contains the replica server.

- The resource group that contains the replica server declares a strong negative affinity for the resource group that contains the application.

EXAMPLE 2-52 Combining Affinities Between Resource Groups

This example shows the commands for combining affinities between the following resource groups.

- Resource group `app-rg` represents an application whose state is tracked by a replica server.
- Resource group `rep-rg` represents the replica server.

In this example, the resource groups declare affinities as follows:

- Resource group `app-rg` declares a weak positive affinity for resource group `rep-rg`.
- Resource group `rep-rg` declares a strong negative affinity for resource group `app-rg`.

This example assumes that both resource groups exist.

```
# clresourcegroup set -p RG_affinities==rep-rg app-rg
# clresourcegroup set -p RG_affinities--app-rg rep-rg
```

Zone Cluster Resource Group Affinities

The cluster administrator can specify affinities between a resource group in a zone cluster and another resource group in a zone cluster or a resource group on the global cluster.

You can use the following command to specify the affinity between resource groups in zone clusters.

```
# clresourcegroup set -p RG_affinities=affinity-typetarget-zc:target-rg source-zc:source-rg
```

The resource group affinity types in a zone cluster can be one of the following:

- + (weak positive)
- ++ (strong positive)
- +++ (strong positive with failover delegation)
- - (weak negative)
- -- (strong negative)

EXAMPLE 2-53 Specifying a Strong Positive Affinity Between Resource Groups in Zone Clusters

This example shows the command for specifying a strong positive affinity between resource groups in zone clusters.

The resource group `RG1` in a zone cluster `ZC1` declares a strong positive affinity for a resource group `RG2` in a zone cluster `ZC2`.

If you need to specify a strong positive affinity between a resource group `RG1` in a zone cluster `ZC1` and a resource group `RG2` in another zone cluster `ZC2`, use the following command:

EXAMPLE 2-53 Specifying a Strong Positive Affinity Between Resource Groups in Zone Clusters
(Continued)

```
# clresourcegroup set -p RG_affinities=++ZC2:RG2 ZC1:RG1
```

EXAMPLE 2-54 Specifying a Strong Negative Affinity Between a Resource Group in a Zone Cluster and a Resource Group in the Global Cluster

This example shows the command for specifying a strong negative affinity between resource groups in zone clusters. If you need to specify a strong negative affinity between a resource group RG1 in a zone cluster ZC1 and a resource group RG2 in the global cluster, use the following command:

```
# clresourcegroup set -p RG_affinities=-global:RG2 ZC1:RG1
```

Configuring the Distribution of Resource Group Load Across Nodes

You can enable the automatic distribution of resource group load across nodes by setting load limits. You assign load factors to resource groups, and the load factors correspond to the defined load limits of the nodes.

The default behavior is to distribute resource group load evenly across all the available nodes. Each resource group is started on a node from its node list. The Resource Group Manager (RGM) chooses a node that best satisfies the configured load distribution policy. As resource groups are assigned to nodes by the RGM, the resource groups' load factors on each node are summed up to provide a total load. The total load is then compared against that node's load limits.

You can configure load limits in a global cluster or a zone cluster.

The factors you set to control load distribution on each node include load limits, resource group priority, and preemption mode. In the global cluster, you can set the `Concentrate_load` property to choose the preferred load distribution policy: to concentrate resource group load onto as few nodes as possible without exceeding load limits or to spread the load out as evenly as possible across all available nodes. The default behavior is to spread out the resource group load. Each resource group is still limited to running only on nodes in its node list, regardless of load factor and load limit settings.

Note – You can use the command line or the `clsetup` utility to configure load distribution for resource groups. The following procedure illustrates how to configure load distribution for resource groups using the `clsetup` utility. For instructions on using the command line to perform these procedures, see “Configuring Load Limits” in *Oracle Solaris Cluster System Administration Guide*.

This section contains the following procedures:

- “How to Configure Load Limits for a Node” on page 149
- “How to Set Priority for a Resource Group” on page 150
- “How to Set Load Factors for a Resource Group” on page 151
- “How to Set Preemption Mode for a Resource Group” on page 152
- “How to Concentrate Load Onto Fewer Nodes in the Cluster” on page 153

▼ How to Configure Load Limits for a Node

Each cluster node can have its own set of load limits. You assign load factors to resource groups, and the load factors correspond to the defined load limits of the nodes. You can set soft load limits (which can be exceeded) or hard load limits (which cannot be exceeded).

1 Assume the root role on one active node of the cluster.

2 Start the `clsetup` utility.

```
phys-schost# clsetup
```

The `clsetup` menu is displayed.

3 Choose the menu item, Other Cluster Tasks.

The Other Cluster Tasks Menu is displayed.

4 Choose the menu item, Manage Resource Group Load Distribution.

The Manage Resource Group Load Distribution Menu is displayed.

5 Choose the menu item, Manage Load Limits.

The Manage load limits Menu is displayed.

6 Type `yes` and press the Return key to continue.

7 Type the option number for the operation you want to perform and press the Return key.

You can create a load limit, modify a load limit, or delete a load limit.

- 8 If you chose to create a load limit, select the option number for the node where you want to set the load limit.**

If you want to set a load limit on a second node, select the option number for the second node and press the Return key. After you have selected all the nodes where you want to configure load limits, type `q` and press the Return key.

- 9 Type `yes` and press the Return key to confirm the nodes you selected in [Step 8](#).**

- 10 Type the name of the load limit and press the Return key.**

For example, type `mem_load` as the name of a load limit.

- 11 Type `yes` or `no` to specify a soft limit value and press the Return key.**

If you typed `yes`, type the soft limit value and press Enter.

- 12 Type `yes` or `no` to specify a hard limit value and press the Return key.**

If you typed `yes`, type the hard limit value and press Enter.

- 13 Type `yes` and press the Return key to proceed with creating the load limit.**

- 14 Type `yes` to proceed with the update and press the Return key.**

The message `Command completed successfully` is displayed, along with the soft and hard load limits for the nodes you selected. Press the Return key to continue.

- 15 You can modify or delete a load limit by following the prompts in the `clsetup` utility.**

Return to the previous menu by typing `q` and pressing the Return key.

▼ **How to Set Priority for a Resource Group**

You can configure a resource group to have a higher priority so that it is less likely to be displaced from a specific node. If load limits are exceeded, lower-priority resource groups might be forced offline.

- 1 Assume the root role on one active node of the cluster.**

- 2 Start the `clsetup` utility.**

```
phys-schost# clsetup
```

The `clsetup` menu is displayed.

- 3 Choose the menu item, `Other Cluster Tasks`.**

The `Other Cluster Tasks` Menu is displayed.

- 4 Choose the menu item, Manage Resource Group Load Distribution.**
The Manage Resource Group Load Distribution Menu is displayed.
- 5 Choose the menu item, Set Priority Per Resource Group.**
The Set the Priority of a Resource Group Menu is displayed.
- 6 Type yes and Press the Return key.**
- 7 Type the option for the resource group and press the Return key.**
The existing Priority value is displayed. The default Priority value is 500.
- 8 Type the new Priority value and press the Return key.**
- 9 Type yes to confirm your entry and press the Return key.**
- 10 Press the Return key to return to the previous menu.**
The Manage Resource Group Load Distribution Menu is displayed.

▼ How to Set Load Factors for a Resource Group

A load factor is a value that you assign to the load on a load limit. Load factors are assigned to a resource group, and those load factors correspond to the defined load limits of the nodes.

- 1 Assume the root role on one active node of the cluster.**
- 2 Start the `clsetup` utility.**

```
phys-schost# clsetup
```


The `clsetup` menu is displayed.
- 3 Choose the menu item, Other Cluster Tasks.**
The Other Cluster Tasks Menu is displayed.
- 4 Choose the menu item, Manage Resource Group Load Distribution.**
The Manage Resource Group Load Distribution Menu is displayed.
- 5 Choose the menu item, Set Load Factors Per Resource Group.**
The Set the load factors of a Resource Group Menu is displayed.
- 6 Type yes and press the Return key.**
- 7 Type the option number for the resource group and press the Return key.**

8 Type the desired load factor.

For example, you can set a load factor called `mem_load` on the resource group you selected by typing `mem_load@50`. Press `Ctrl-D` when you are done.

9 Press the Return key to proceed with the update.**10 Press the Return key to return to the previous menu.**

The Manage Resource Group Load Distribution Menu is displayed.

▼ How to Set Preemption Mode for a Resource Group

The `preemption_mode` property determines if a resource group will be preempted from a node by a higher-priority resource group because of node overload. The property indicates the cost of moving a resource group from one node to another.

1 Assume the root role on one active node of the cluster.**2 Start the `clsetup` utility.**

```
phys-schost# clsetup
```

The `clsetup` menu is displayed.

3 Choose the menu item, Other Cluster Tasks.

The Other Cluster Tasks Menu is displayed.

4 Choose the menu item, Manage Resource Group Load Distribution.

The Manage Resource Group Load Distribution Menu is displayed.

5 Choose the menu item, Set Preemption Mode per Resource Group.

The Set the Preemption Mode of a Resource Group Menu is displayed.

6 Type yes and press the Return key to continue.**7 Type the option number for the resource group and press the Return key.**

If the resource group has a preemption mode set, it is displayed, similar to the following:

The preemption mode property of "rg11" is currently set to the following: `preemption mode: Has_Cost`

8 Type the option number for the preemption mode you want and press the Return key.

The three choices are `Has_cost`, `No_cost`, or `Never`.

9 Type yes to proceed with the update and press the Return key.

- 10 **Press the Return key to return to the previous menu.**

The Manage Resource Group Load Distribution Menu is displayed.

▼ **How to Concentrate Load Onto Fewer Nodes in the Cluster**

Setting the `Concentrate_load` property to `false` causes the cluster to spread resource group loads evenly across all available nodes in the resource groups' node lists. By default, the `Concentrate_load` property is set to `FALSE`.

If you set this property to `TRUE`, the cluster attempts to concentrate resource group load on the fewest possible nodes without exceeding any configured hard or soft load limits.

Note – When specifying `Concentrate_load=TRUE`, if a resource group RG2 declares a `++` or `+++` affinity for a resource group RG1, avoid setting any nonzero load factors for RG2. Instead, set larger load factors for RG1 to account for the additional load that would be imposed by RG2 coming online on the same node as RG1. This will allow the `Concentrate_load` feature to work as intended. Alternately, you can set load factors on RG2 but avoid setting any hard load limits for those load factors—set only soft limits. This will allow RG2 to come online even if the soft load limit is exceeded.

You can only set the `Concentrate_load` property in a global cluster; you cannot set this property in a zone cluster. In a zone cluster, the default setting is always `FALSE`.

- 1 **Assume the root role on one active node of the cluster.**

- 2 **Start the `clsetup` utility.**

```
phys-schost# clsetup
```

The `clsetup` menu is displayed.

- 3 **Choose the menu item, Other cluster tasks.**

The Other Cluster Tasks Menu is displayed.

- 4 **Choose the menu item, Set the `concentrate_load` Property of the Cluster.**

The Set the Concentrate Load Property of the Cluster Menu is displayed.

- 5 **Type `yes` and press the Return key.**

The current value of `TRUE` or `FALSE` is displayed.

- 6 **Type `yes` to change the value and press the Return key.**

- 7 Type **yes** to proceed with the update and press the Return key.
- 8 Press the Return key to return to the previous menu.

The Other Cluster Tasks Menu is displayed.

Enabling Oracle Solaris SMF Services to Run With Oracle Solaris Cluster

The Service Management Facility (SMF) enables you to automatically start and restart SMF services, during a node boot or service failure. SMF facilitates some degree of high availability to the SMF services on a single host. This feature is similar to the Oracle Solaris Cluster Resource Group Manager (RGM), which facilitates high availability and scalability for cluster applications. SMF services and RGM features are complementary to each other.

Oracle Solaris Cluster includes three SMF proxy resource types that can be used to enable SMF services to run with Oracle Solaris Cluster in a failover, multi-master, or scalable configuration. The following are the proxy resource types:

- `SUNW.Proxy_SMF_failover`
- `SUNW.Proxy_SMF_multimaster`
- `SUNW.Proxy_SMF_scalable`

The SMF proxy resource types enables you to encapsulate a set of interrelated SMF services into a single resource, *SMF proxy resource* to be managed by Oracle Solaris Cluster. In this feature, SMF manages the availability of SMF services on a single node. Oracle Solaris Cluster provides cluster-wide high availability and scalability of the SMF services.

You can use the SMF proxy resource types to integrate your own SMF controlled services into Oracle Solaris Cluster so that these services have cluster-wide service availability without you rewriting callback methods or service manifest. After you integrate the SMF service into the SMF proxy resource, the SMF service is no longer managed by the default restarter. The restarter that is delegated by Oracle Solaris Cluster manages the SMF service.

SMF proxy resources are identical to other resources, with no restriction on their usage. For example, an SMF proxy resource can be grouped with other resources into a resource group. SMF proxy resources can be created and managed the same way as other resources. An SMF proxy resource differs from other resources in one way. When you create a resource of any of the SMF proxy resource types, you need to specify the extension property `Proxied_service_instances`. You must include information about the SMF services to be proxied by the SMF resource. The extension property's value is the path to a file that contains all the proxied SMF services. Each line in the file is dedicated to one SMF service and specifies `svc_fmri`, path of the corresponding service manifest file.

For example, if the resource has to manage two services, `restarter_svc_test_1:default` and `restarter_svc_test_2:default`, the file should include the following two lines:

```
<svc:/system/cluster/restarter_svc_test_1:default>,</var/svc/manifest/system/cluster/restarter_svc_test_1.xml>
```

```
<svc:/system/cluster/restarter_svc_test_2:default>,</var/svc/manifest/system/cluster/restarter_svc_test_2.xml>
```

The services that are encapsulated under an SMF proxy resource can reside in the global cluster. All the services under the same proxy resource must be in the same zone.



Caution – Do not use SMF `svcadm` for disabling or enabling SMF services that are encapsulated in a proxy resource. Do not change the properties of the SMF services (in the SMF repository) that are encapsulated in a proxy resource.

- “Encapsulating an SMF Service Into a Failover Proxy Resource Configuration” on page 155
- “Encapsulating an SMF Service Into a Multi-Master Proxy Resource Configuration” on page 157
- “Encapsulating an SMF Service Into a Scalable Proxy Resource Configuration” on page 160

▼ Encapsulating an SMF Service Into a Failover Proxy Resource Configuration

For information about failover configuration, see “Creating a Resource Group” on page 42

Note – Perform this procedure from any cluster node.

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.

- 2 Register the proxy SMF failover resource type.

```
# clresourcetype register -f \
/opt/SUNWscsmf/etc/SUNW.Proxy_SMF_failover SUNW.Proxy_SMF_failover
```

- 3 Verify that the proxy resource type has been registered.

```
# clresourcetype show
```

- 4 Create the SMF failover resource group.

```
# clresourcegroup create [-n node-zone-list] resource-group
```

`-n nodelist` Specifies a comma-separated, ordered list of nodes that can master this resource group.

This list is optional. If you omit this list, the resource group is configured on all cluster nodes.

resource-group Specifies your choice of the name of the resource group to add. This name must begin with an ASCII character.

5 Verify that the SMF resource group has been created.

```
# clresourcegroup status resource-group
```

6 Add an SMF failover application resource to the resource group.

```
# clresource create -g resource-group -t SUNW.Proxy_SMF_failover \
-p Port_list=portnumber/protocol \
-x Proxied_service_instances=/tmp/dns_svcs.txt
```

-g resource-group Specifies the name of the SMF failover resource group that you previously created.

-p Port_list=portnumber/protocol Specifies the port number the instance will use to listen for activity. The protocol can be either tcp or udp.

-p Proxied_service_instances Specifies the path to the file you created that specifies the mapping of SMF services and their corresponding manifests for the SMF services to be proxied. In the example above, /tmp/dns_svcs.txt is the path to the text file.

The resource is created in the enabled state.

7 Verify that the SMF failover application resource has been added and validated.

```
# clresource show resource
```

8 Bring the failover resource group online.

```
# clresourcegroup online -M resource-group
```

Note – If you use the `clresource status` command to view the state of the SMF proxy resource type, the status is displayed as `onLine` but not `monitored`. This is not an error message. The SMF proxy resource is enabled and running and this status message is displayed because there is no monitoring support provided for the resources of SMF proxy resource type.

Example 2–55 Registering an SMF Proxy Failover Resource Type

The following example registers the `SUNW.Proxy_SMF_failover` resource type.

```
# clresourcetype register SUNW.Proxy_SMF_failover
# clresourcetype show SUNW.Proxy_SMF_failover

Resource Type:          SUNW.Proxy_SMF_failover
RT_description:         Resource type for proxying failover SMF services
RT_version:             2.0
API_version:            7
```

```

RT_basedir:           /opt/SUNWscsmf/bin
Single_instance:     False
Proxy:               False
Init_nodes:          All potential masters
Installed_nodes:     <All>
Failover:            True
Pkglist:             <NULL>
RT_system:           False
Global_zone:         False

```

Example 2-56 Adding an SMF Proxy Failover Application Resource to a Resource Group

This example shows the addition of a proxy resource type, `SUNW.Proxy_SMF_failover` to a resource group `resource-group-1`.

```

# clresource create -g resource-group-1 -t SUNW.Proxy_SMF_failover \
-x proxied_service_instances=/var/tmp/svslst.txt resource-1
# clresource show resource-1

=== Resources ===

Resource:           resource-1
Type:               SUNW.Proxy_SMF_failover
Type_version:      2.0
Group:              resource-group-1
R_description:
Resource_project_name: default
Enabled{phys-schost-1}: True
Monitored{phys-schost-1}: True

```

▼ Encapsulating an SMF Service Into a Multi-Master Proxy Resource Configuration

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Register the SMF proxy multi-master resource type.

```

# clresourcetype register -f \
/opt/SUNWscsmf/etc/SUNW.Proxy_SMF_multimaster SUNW.Proxy_SMF_multimaster

```

- 3 Create the SMF multi-master resource group.

```

# clresourcegroup create -S [-p Maximum primaries=m] [-p Desired primaries=n] \
[-n node-zone-list] resource-group

```

-S	Specifies that the resource group is to be multi-mastered. If the -p Maximum_primaries and -p Desired_primaries options are omitted, both properties are set to the number of nodes in the resource group's node list.
-p Maximum_primaries= <i>m</i>	Specifies the maximum number of active primaries for this resource group.
-p Desired_primaries= <i>n</i>	Specifies the number of active primaries on which the resource group should attempt to start.
-n <i>nodelist</i>	Specifies a comma-separated, ordered list of nodes in which this resource group is to be available. This list is optional. If you omit this list, the resource group is configured on all cluster nodes.
<i>resource-group</i>	Specifies your choice of the name of the scalable resource group to add. This name must begin with an ASCII character.

4 Verify that the SMF proxy multi-master resource group has been created.

```
# clresourcegroup show resource-group
```

5 Add an SMF proxy multi-master resource to the resource group.

```
# clresource create -g resource-group -t SUNW.Proxy_SMF_multimaster \  
-p Port_list=portnumber/protocol \  
-x Proxied_service_instances=/tmp/dns_svcs.txt
```

-g <i>resource-group</i>	Specifies the name of the SMF multi-master resource group that you previously created.
-p Port_list= <i>portnumber/protocol</i>	Specifies the port number the instance will use to listen for activity. The protocol can be either tcp or udp.
-p Proxied_service_instances	Specifies the path to the file you created that specifies the mapping of SMF services and their corresponding manifests for the SMF services to be proxied. In the example above, <i>/tmp/dns_svcs.txt</i> is the path to the text file.

The resource is created in the enabled state.

6 Verify that the SMF proxy multi-master application resource has been added and validated.

```
# clresource show resource
```

7 Bring the multi-master resource group online.

```
# clresourcegroup online -M resource-group
```

Note – If you use the `clresource status` command to view the state of the SMF proxy resource type, the status is displayed as `online` but not `monitored`. This is not an error message. The SMF proxy resource is enabled and running and this status message is displayed because there is no monitoring support provided for the resources of SMF proxy resource type.

Example 2–57 Registering an SMF Proxy Multi-Master Resource Type

The following example registers the `SUNW.Proxy_SMF_multimaster` resource type.

```
# clresourcetype register SUNW.Proxy_SMF_multimaster
# clresourcetype show SUNW.Proxy_SMF_multimaster

Resource Type:          SUNW.Proxy_SMF_multimaster
RT_description:         Resource type for proxying multimastered SMF services
RT_version:             2.0
API_version:            7
RT_basedir:             /opt/SUNWscsmf/bin
Single_instance:        False
Proxy:                  False
Init_nodes:             All potential masters
Installed_nodes:        <All>
Failover:               True
Pkglist:                <NULL>
RT_system:              False
Global_zone:            False
```

Example 2–58 Creating and Adding an SMF Proxy Multi-Master Application Resource to a Resource Group

This example shows the creation and addition of a multi-master proxy resource type `SUN.Proxy_SMF_multimaster` to a resource group `resource-group-1`.

```
# clresourcegroup create -S \
-p Maximum primaries=2 \
-p Desired primaries=2 \
-n phys-schost-1, phys-schost-2 resource-group-1
# clresourcegroup show resource-group-1

=== Resource Groups and Resources ===

Resource Group:          resource-group-1
RG_description:          <NULL>
RG_mode:                 multimastered
RG_state:                Unmanaged
RG_project_name:         default
RG_affinities:           <NULL>
Auto_start_on_new_cluster: True
Failback:                False
Nodelist:                phys-schost-1 phys-schost-2
Maximum primaries:      2
```

```

Desired_primaries:                2
Implicit_network_dependencies:    True
Global_resources_used:           <All>
Pingpong_interval:               3600
Pathprefix:                      <NULL>
RG_System:                       False
Suspend_automatic_recovery:      False

# clresource create -g resource-group-1 -t SUNW.Proxy_SMF_multimaster \
-x proxied_service_instances=/var/tmp/svslst.txt resource-1
# clresource show resource-1

=== Resources ===

Resource:                        resource-1
Type:                            SUNW.Proxy_SMF_multimaster
Type_version:                    2.0
Group:                           resource-group-1
R_description:
Resource_project_name:           default
Enabled{phys-schost-1}:          True
Monitored{phys-schost-1}:       True

```

▼ Encapsulating an SMF Service Into a Scalable Proxy Resource Configuration

For information about scalable configuration, see [“How to Create a Scalable Resource Group” on page 43](#).

Note – Perform this procedure from any cluster node.

- 1 On a cluster member, assume the root role that provides `solaris.cluster.modify` RBAC authorization.
- 2 Register the SMF proxy scalable resource type.

```
# clresourcetype register -f \
/opt/SUNWscsmf/etc/SUNW.Proxy_SMF_scalable SUNW.Proxy_SMF_scalable
```
- 3 Create the SMF failover resource group that holds the shared address that the scalable resource group is to use. See [“How to Create a Failover Resource Group” on page 42](#) to create the failover resource group.
- 4 Add the shared address resource to the failover resource group.

See [“How to Add a Shared Address Resource to a Resource Group Using the Command-Line Interface” on page 55](#).

5 Create the SMF proxy scalable resource group.

```
# clresourcegroup create -S [-p Maximum primaries=m] [-p Desired primaries=n] \
[-n node-zone-list] resource-group
```

-S Specifies that the resource group is to be multi-mastered. If the `-p Maximum primaries` and `-p Desired primaries` options are omitted, both properties are set to the number of nodes in the resource group's node list.

-p Maximum primaries=*m* Specifies the maximum number of active primaries for this resource group.

-p Desired primaries=*n* Specifies the number of active primaries on which the resource group should attempt to start.

-n *nodelist* Specifies a comma-separated, ordered list of nodes in which this resource group is to be available.

This list is optional. If you omit this list, the resource group is created on all nodes in the cluster.

resource-group Specifies your choice of the name of the scalable resource group to add. This name must begin with an ASCII character.

6 Verify that the scalable resource group has been created.

```
# clresourcegroup show resource-group
```

7 Add an SMF proxy scalable resource to the scalable resource group you created in [Step 5](#).

```
# clresource create-g resource-group -t SUNW.Proxy_SMF_scalable \
-p Resource_dependencies=network-resource[,network-resource...] \
-p Scalable=True \
-p Port_list=portnumber/protocol \
-x Proxied_service_instances=/tmp/dns_svcs.txt
```

-p Resource_dependencies=*network-resource[,network-resource...]* Specifies the name of the scalable network resource you created in [Step 3](#) on which this resource depends.

-g *resource-group* Specifies the name of the SMF proxy scalable resource group that you previously created.

-p Scalable=True Specifies that this resource uses the network load balancing features of Oracle Solaris Cluster software. For more information, see [“How to Add a Scalable Application Resource to a Resource Group” on page 59](#).

The resource is created in the enabled state.

8 Verify that the SMF proxy scalable application resource has been added and validated.

```
# clresource show resource
```

9 Bring the SMF proxy scalable resource group online.

```
# clresourcegroup online -M resource-group
```

Note – If you use the `clresource status` command to view the state of the SMF proxy resource type, the status is displayed as `online` but not `monitored`. This is not an error message. The SMF proxy resource is enabled and running and this status message is displayed because there is no monitoring support provided for the resources of SMF proxy resource type.

Example 2–59 Registering an SMF Proxy Scalable Resource Type

The following example registers the `SUNW.Proxy_SMF_scalable` resource type.

```
# clresourcetype register SUNW.Proxy_SMF_scalable
# clresourcetype show SUNW.Proxy_SMF_scalable

Resource Type:          SUNW.Proxy_SMF_scalable
RT_description:        Resource type for proxying scalable SMF services
RT_version:            2.0
API_version:           7
RT_basedir:            /opt/SUNWscsmf/bin
Single_instance:      False
Proxy:                 False
Init_nodes:            All potential masters
Installed_nodes:       <ALL>
Failover:              True
Pkglist:               <NULL>
RT_system:             False
Global_zone:           False
```

Example 2–60 Creating and Adding an SMF Proxy Scalable Application Resource to a Resource Group

This example shows the creation and addition of a scalable proxy resource type `SUNW.Proxy_SMF_scalable` to a resource group `resource-group-1`.

```
# clresourcegroup create -S \
-p Maximum primaries=2 \
-p Desired primaries=2 \
-p RG_dependencies=resource-group-2 \
-n phys-schost-1, phys-schost-2 resource-group-1
# clresourcegroup show resource-group-1

=== Resource Groups and Resources ===

Resource Group:          resource-group-1
RG_description:          <NULL>
RG_mode:                 Scalable
RG_state:                Unmanaged
RG_project_name:         default
RG_affinities:           <NULL>
Auto_start_on_new_cluster: True
```

```

Failback:                               False
Nodelist:                               phys-schost-1 phys-schost-2
Maximum primaries:                       2
Desired primaries:                       2
RG_dependencies:                         resource-group2
Implicit_network_dependencies:            True
Global_resources_used:                   <All>
Pingpong_interval:                       3600
Pathprefix:                              <NULL>
RG_System:                               False
Suspend_automatic_recovery:              False

# clresource create -g resource-group-1 -t SUNW.Proxy_SMF_scalable \
-p resource_dependencies=net-res -p port_list=1080/tcp \
-x proxied_service_instances=/var/tmp/svslst.txt resource-1
# clresource show resource-1

=== Resources ===

Resource:                               resource-1
Type:                                    SUNW.Proxy_SMF_scalable
Type_version:                           2.0
Group:                                    resource-group-1
R_description:                           resource-1
Resource_project_name:                   default
Enabled{phys-schost-1}:                  True
Monitored{phys-schost-1}:                True

```

You can choose the `resource_dependencies` and port number to use.

Tuning Fault Monitors for Oracle Solaris Cluster Data Services

Each data service that is supplied with the Oracle Solaris Cluster product has a built-in fault monitor. The fault monitor performs the following functions:

- Detecting the unexpected termination of processes for the data service server
- Checking the health of the data service

The fault monitor is contained in the resource that represents the application for which the data service was written. You create this resource when you register and configure the data service. For more information, see the documentation for the data service.

System properties and extension properties of this resource control the behavior of the fault monitor. The default values of these properties determine the preset behavior of the fault monitor. The preset behavior should be suitable for most Oracle Solaris Cluster installations. Therefore, you should tune a fault monitor *only* if you need to modify this preset behavior.

Tuning a fault monitor involves the following tasks:

- Setting the interval between fault monitor probes
- Setting the timeout for fault monitor probes

- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

Perform these tasks when you register and configure the data service. For more information, see the documentation for the data service.

Note – A resource's fault monitor is started when you bring online the resource group that contains the resource. You do not need to start the fault monitor explicitly.

Setting the Interval Between Fault Monitor Probes

To determine whether a resource is operating correctly, the fault monitor probes this resource periodically. The interval between fault monitor probes affects the availability of the resource and the performance of your system as follows:

- The interval between fault monitor probes affects the length of time that is required to detect a fault and respond to the fault. Therefore, if you decrease the interval between fault monitor probes, the time that is required to detect a fault and respond to the fault is also decreased. This decrease enhances the availability of the resource.
- Each fault monitor probe consumes system resources such as processor cycles and memory. Therefore, if you decrease the interval between fault monitor probes, the performance of the system is degraded.

The optimum interval between fault monitor probes also depends on the time that is required to respond to a fault in the resource. This time depends on how the complexity of the resource affects the time that is required for operations such as restarting the resource.

To set the interval between fault monitor probes, set the `Thorough_probe_interval` system property of the resource to the interval in seconds that you require.

Setting the Timeout for Fault Monitor Probes

The timeout for fault monitor probes specifies the length of time that a fault monitor waits for a response from a resource to a probe. If the fault monitor does not receive a response within this timeout, the fault monitor treats the resource as faulty. The time that a resource requires to respond to a fault monitor probe depends on the operations that the fault monitor performs to probe the resource. For information about operations that a data service's fault monitor performs to probe a resource, see the documentation for the data service.

The time that is required for a resource to respond also depends on factors that are unrelated to the fault monitor or the application, for example:

- System configuration

- Cluster configuration
- System load
- Amount of network traffic

To set the timeout for fault monitor probes, set the `Probe_timeout` extension property of the resource to the timeout in seconds that you require.

Defining the Criteria for Persistent Faults

To minimize the disruption that transient faults in a resource cause, a fault monitor restarts the resource in response to such faults. For persistent faults, more disruptive action than restarting the resource is required:

- For a failover resource, the fault monitor fails over the resource to another node.
- For a scalable resource, the fault monitor takes the resource offline.

A fault monitor treats a fault as persistent if the number of complete failures of a resource exceeds a specified threshold within a specified retry interval. Defining the criteria for persistent faults enables you to set the threshold and the retry interval to accommodate the performance characteristics of your cluster and your availability requirements.

Complete Failures and Partial Failures of a Resource

A fault monitor treats some faults as a *complete failure* of a resource. A complete failure typically causes a complete loss of service. The following failures are examples of a complete failure:

- Unexpected termination of the process for a data service server
- Inability of a fault monitor to connect to a data service server

A complete failure causes the fault monitor to increase by 1 the count of complete failures in the retry interval.

A fault monitor treats other faults as a *partial failure* of a resource. A partial failure is less serious than a complete failure, and typically causes a degradation of service, but not a complete loss of service. An example of a partial failure is an incomplete response from a data service server before a fault monitor probe is timed out.

A partial failure causes the fault monitor to increase by a fractional amount the count of complete failures in the retry interval. Partial failures are still accumulated over the retry interval.

The following characteristics of partial failures depend on the data service:

- The types of faults that the fault monitor treats as partial failure
- The fractional amount that each partial failure adds to the count of complete failures

For information about faults that a data service's fault monitor detects, see the documentation for the data service.

Dependencies of the Threshold and the Retry Interval on Other Properties

The maximum length of time that is required for a single restart of a faulty resource is the sum of the values of the following properties:

- `Thorough_probe_interval` system property
- `Probe_timeout` extension property

To ensure that you allow enough time for the threshold to be reached within the retry interval, use the following expression to calculate values for the retry interval and the threshold:

$$\text{retry_interval} \geq 2 \times \text{threshold} \times (\text{thorough_probe_interval} + \text{probe_timeout})$$

The factor of 2 accounts for partial probe failures that do not immediately cause the resource to be failed over or taken offline.

System Properties for Setting the Threshold and the Retry Interval

To set the threshold and the retry interval, set the following system properties of the resource:

- To set the threshold, set the `Retry_count` system property to the maximum allowed number of complete failures.
- To set the retry interval, set the `Retry_interval` system property to the interval in seconds that you require.

Specifying the Failover Behavior of a Resource

The failover behavior of a resource determines how the RGM responds to the following faults:

- Failure of the resource to start
- Failure of the resource to stop
- Failure of the resource's fault monitor to stop

To specify the failover behavior of a resource, set the `Failover_mode` system property of the resource. For information about the possible values of this property, see the description of the `Failover_mode` system property in the `r_properties(5)` man page.

Index

A

- adding
 - file systems to HAStoragePlus resource, 127–128
 - nodes to resource groups
 - overview, 95
 - failover, 96–98
 - scalable, 96
 - resources to a resource group
 - overview, 48–61
 - failover application, 57–59
 - logical hostname, 49–51, 51–53
 - scalable application, 59–61
 - shared address, 53–55, 55–57
- affinities, resource groups, 139–141
- application binaries, determining location, 14–15
- attributes, resource property, 24
- automatic load distribution
 - concentrate load, 153–154
 - load factors, 151–152
 - load limits, 149–150
 - overview, 148–154
 - preemption, 152–153
 - priority, 150–151
- automatic recovery actions, 66–68
- Availability Suite, managed entity monitoring, 104

B

- balancing, load on cluster nodes, 143–144
- bringing online, resource groups, 62–63
- bypassing, name services, 85–86

C

- caution notice, removal of file systems, 129
- changing
 - resource group properties, 80–81
 - resource properties, 81–82
 - resource type properties, 79–80
- CheckNameService extension property, 85
- clearing
 - Start_failed resource state, 88–90, 90–91, 91–92
 - STOP_FAILED error flag, 86–87
- clsetup utility, 21
- clsetup utility
 - logical hostname
 - adding to a resource group, 49–51
 - shared address
 - adding to a resource group, 53–55
- cluster properties, 23
 - Concentrate_load, 23
- collocation
 - enforcing for online resource groups, 141–142
 - preferring for online resource groups, 142–143
- combining, resource group affinities, 146–147
- Command-Line Interface
 - logical hostname
 - adding to a resource group, 51–53
 - shared address
 - adding to a resource group, 55–57
- complete failures, 165–166
- Concentrate_load, resource property, 23
- concentrate load, setting for load distribution, 153–154
- configuration
 - guidelines, 14–16

configuration (*Continued*)

- overview, 19–21
- planning cluster file system, 15
- configuring and administering, Oracle Solaris Cluster data services, 31
- considerations, 19
- creating
 - failover application resources, 57–59
 - logical hostname resources, 49–51, 51–53
 - resource groups
 - failover, 42–43
 - scalable, 43–45
 - scalable application resources, 59–61
 - shared address resources, 53–55
 - using CLI, 55–57

D

- data services
 - considerations, 19
 - planning, 13–26
 - special requirements, 14
- defining, persistent faults, 165–166
- delegating, resource group failover or switchover, 145–146
- description values, rules, 26
- device groups
 - relationship with resource groups, 17
 - synchronizing startups with resource groups, 103–109
- directive, `#$upgrade`, 25
- disabled resources, unexpected behavior, 76
- disabling
 - resource fault monitors, 69
 - resources, 76–78, 91–92
 - SMF instances, 16
- displaying, resource type, resource group, and resource configuration, 78
- distributing, online resource groups, 139–148
- downgrading, resource types, 40–41

E

- editing
 - HASStoragePlus resources, 138–139
 - logical hostname resources, 93–94
 - shared address resources, 93–94
- enabling
 - Oracle Solaris SMF services, 154–163
 - resource fault monitors, 69–70
 - resources, 64–65, 91–92
- enumeration literal names, rules, 24
- error flags, `STOP_FAILED`, 86–87
- error messages
 - failure to modify file system, 135, 136
- `/etc/vfstab` file
 - adding entries to, 127
 - removing entries from, 130
- extension properties
 - `Probe_timeout`
 - effect on restart time, 166
 - tuning, 165

F

- failover, delegating for resource groups, 145–146
- failover application resources, adding to resource group, 57–59
- failover file systems, *See* highly available local file system, 118
- `Failover_mode` system property, 166
- failovers, preserving distribution of online resource groups, 139–148
- fault monitors
 - disabling, 69
 - enabling, 69–70
 - faults detected by, 166
 - probe interval, 164
 - probe timeout, 164–165
 - resource, 68–70
 - response to faults, 166
 - tuning, 163–166
- faults
 - modification of file systems, 135–136, 136–137
 - persistent, 165–166
 - responses to, 166

file systems
 caution notice, 129
 failure to modify, 135–136
 removing from `HASStoragePlus` resource, 129–130
 unmounting, 129

files

`/etc/vfstab`
 adding entries to, 127
 removing entries from, 130
 RTR, 138

format, resource type names, 25

freeing node resources, affinities, 144–145

H

`HASStoragePlus`, 122–126

managed entity monitoring, 104

`HASStoragePlus` resource

cluster file systems

change from local file system, 137–138

configuring, 110–112

`HASStoragePlus` resource type

caution notice, 129

criteria for using, 18–19

failure to modify instances of, 135–136, 136–137

modifying instances of, 126–137

overview, 17–19

resource type versions, 138

upgrading, 138–139

help, 12

highly available local file system across zone clusters,

sharing, 122–126

highly available local file systems, 15

adding file systems to, 127–128

caution notice, 129

enabling, 112

failure to modify, 135–136, 136–137

modifying, 126–137

removing file systems from, 129–130

I

installation, overview, 19–21

Internet Protocol (IP) addresses, restrictions, 19

intervals, fault monitor probes, 164

`IOOption` property, 105

L

legal names

Resource Group Manager (RGM), 24–26

load balancing, 143–144

load distribution of resource groups

concentrate load, 153–154

load factors, 151–152

load limits, 149–150

overview, 148–154

preemption mode, 152–153

priority, 150–151

load factors, setting for load distribution, 151–152

load limits, setting for load distribution, 149–150

local file systems

adding to `HASStoragePlus` resource, 127–128

failure to modify, 136–137

highly available

enabling, 112

modifying, 126–137

list of supported, 113

logical hostname resources

adding to resource group

using CLI, 51–53

using `clsetup` utility, 49–51

modifying, 85–86

loopback mount, using `HASStoragePlus`, 110–112

M

maximum values, restarts, 165

migrating

`HASStoragePlus` resources, 138–139

logical hostname resources, 93–94

shared address resources, 93–94

migrating to a new resource type version, 35–39

mission-critical services, 144–145

modifying resources, 85–86

N

- name services, bypassing, 85–86
- networks, restrictions, 19
- `NodeList` resource group property, and affinities, 140
- nodes
 - adding to resource groups
 - overview, 95
 - failover, 96–98
 - scalable, 96
 - distributing resource groups on, 139–148
 - load balancing, 143–144
 - off-loading noncritical services, 144–145
 - removing from resource groups
 - overview, 98
 - failover, 100–101
 - failover with shared addresses, 101–102
 - scalable, 99–100
- noncritical services, off-loading, 144–145
- `nsswitch.conf`, verifying file contents, 15

O

- off-loading noncritical resource groups,
 - affinities, 144–145
- Oracle Service Management Facility (SMF)
 - enabling, 154–163
 - encapsulating into a failover proxy
 - resource, 155–157
 - encapsulating into a multi-master resource type, 157–160
 - encapsulating into a scalable proxy resource, 160
- Oracle Service Management Facility (SMF) services,
 - enabling, 16
- Oracle Solaris Cluster Administrative commands, 21

P

- partial failures, 165–166
- performance
 - effect of probe interval on, 164
 - optimizing for mission-critical services, 144–145
- persistent faults, defining, 165–166
- `ping` command, responses from disabled resources, 76

planning

- cluster file system, 15
- data services, 13–26
- preemption mode, setting for load distribution, 152–153
- preregistered resource types
 - reregistering after inadvertent deletion, 94–95
 - upgrading, 93–94
- priority, setting for load distribution, 150–151
- `Probe_timeout` extension property
 - effect on restart time, 166
 - tuning, 165
- properties
 - See also* extension properties
 - cluster, 23
 - resource, 23
 - resource group, 23
 - `Type_version`, 94, 138
- property attributes, resource, 24
- property names, rules, 24
- property values, rules, 26

Q

- quiescing resource groups, 65–66

R

- recovering
 - from failure to modify file system, 135–136, 136–137
- recovery actions
 - resuming automatic, 66–68
 - suspending automatic, 66–68
- registering
 - `HASStoragePlus` resource type
 - during upgrade, 138
 - preregistered resource types, 94–95
 - resource types, 32–33
 - `SUNW.LogicalHostname` resource type
 - after inadvertent deletion, 94–95
 - during upgrade, 93

registering (*Continued*)

- SUNW.SharedAddress resource type
 - after inadvertent deletion, 94–95
 - during upgrade, 93

removing

- file systems from HAStoragePlus resource, 129–130
- nodes from resource groups
 - overview, 98
 - failover, 100–101
 - failover with shared addresses, 101–102
 - scalable, 99–100
- resource group, 72–73
- resource types, 70–71
- resources, 73

requirements, data services, 14

reregistering preregistered resource types, 94–95

resource dependency, setting, 82–85

resource fault monitors, 68–70

resource group affinities, zone cluster, 147–148

Resource Group Manager (RGM)

- legal names, 24–26
- values, 26

resource group names, rules, 24

resource group properties, 23

resource groups

- adding nodes to, 95
- adding resources, 48–61
 - failover application, 57–59
 - logical hostname, 49–51, 51–53
 - scalable application, 59–61
 - shared address, 53–55, 55–57

affinities, 139–141

automatic load distribution across nodes, 148–154

bringing online, 62–63

changing properties, 80–81

creating

- failover, 42–43
- scalable, 43–45

delegating failover or switchover, 145–146

displaying configuration information, 78

distributing among nodes, 139–148

enforced collocation, 141–142

enforced separation, 144–145

even distribution, 143–144

resource groups (*Continued*)

failover

- adding nodes, 96–98
- removing nodes from, 100–101

failover with shared addresses

- removing nodes from, 101–102
- moving to UNMANAGED state, 76–78

preferred collocation, 142–143

preferred separation, 143–144

quiescing, 65–66

relationship with device groups, 17

removing, 72–73

removing nodes, 98

restarting, 90–91

resuming automatic recovery actions, 66–68

scalable

- adding nodes, 96
- removing nodes from, 99–100

suspending automatic recovery actions, 66–68

switching over, 88–90

switching the current primary, 74–75

switching to preferred primaries, 63–64

synchronizing startups with device groups, 103–109

resource names, rules, 24

resource properties, 23

resource property attributes, 24

resource type, ScalMountPoint, 45

resource type names, rules, 25

resource type registration (RTR) file, 138

resource types

- changing properties, 79–80
- displaying configuration information, 78
- downgrading, 40–41

HAStoragePlus

- existing resources, 109
- migrating instances of, 138–139
- new resources, 107–109

LogicalHostname

- migrating instances of, 93–94

migrating to a new resource type version, 35–39

preregistered

- reregistering after inadvertent deletion, 94–95
- upgrading, 93–94

registering, 32–33

resource types (*Continued*)

- removing, 70–71
- SharedAddress
 - migrating instances of, 93–94
 - unregistering, 70–71
 - upgrading, 34–35
- resources
 - adding to a resource group, 48–61
 - changing properties, 81–82
 - clearing STOP_FAILED error flag, 86–87
 - disabling, 76–78, 91–92
 - disabling fault monitors, 69
 - displaying configuration information, 78
 - enabling, 64–65, 91–92
 - enabling fault monitors, 69–70
 - failover application
 - adding to a resource group, 57–59
 - logical hostname
 - adding to a resource group, 49–51, 51–53
 - modifying, 85–86
 - removing, 73
 - removing resource types, 70–71
 - scalable application
 - adding to a resource group, 59–61
 - shared address
 - adding to a resource group, 53–55, 55–57
 - isolating from hosts when disabled, 76
 - modifying, 85–86
- responses, to faults, 166
- restarting, resource groups, 90–91
- restarts, maximum allowed, 165
- restrictions, 19
- Retry_count system property, 166
- retry interval, 165
- Retry_interval system property, 166
- RG_affinities resource group property, 139–141
- rules
 - description values, 26
 - enumeration literal names, 24
 - property names, 24
 - property values, 26
 - resource group names, 24
 - resource names, 24

S

- scalable application resources, adding to resource group, 59–61
- ScalMountPoint resource type, creating, 45
- setting, resource dependency, 82–85
- setting up
 - HASStoragePlus resource type, 112
 - existing resources, 109
 - new resources, 107–109
- shared address resources
 - adding to resource group, 55–57
 - using clsetup utility, 53–55
 - isolating from hosts when disabled, 76
 - modifying, 85–86
- sharing a highly available local file system, zone clusters, 122–126
- SMF
 - See also* Oracle Service Management Facility (SMF)
 - enabling, 154–163
 - encapsulating into a failover proxy resource, 155–157
 - encapsulating into a multi-master proxy resource, 157–160
 - encapsulating into a scalable proxy resource, 160
 - Oracle Service Management Facility (SMF), 16
- special requirements, identifying, 14
- Start_failed resource state
 - clearing, 88–90, 90–91, 91–92
- STOP_FAILED error flag, 86–87
- strong negative affinity
 - definition, 140
 - usage example, 144–145
- strong positive affinity
 - definition, 140
 - usage example, 141–142
- strong positive affinity with failover delegation
 - definition, 140
 - usage example, 145–146
- SUNW.LogicalHostname resource type
 - reregistering after inadvertent deletion, 94–95
 - resource type versions, 93
 - upgrading, 93–94
- SUNW.SharedAddress resource type
 - reregistering after inadvertent deletion, 94–95

SUNW.SharedAddress resource type (*Continued*)

- resource type versions, 93
- upgrading, 93–94
- switching, resource groups, 88–90
- switching the current primary, resource groups, 74–75
- switching to preferred primaries, resource groups, 63–64
- switchover, delegating for resource groups, 145–146
- synchronizing startups, resource groups and device groups, 103–109
- syntax
 - description values, 26
 - enumeration literal names, 24
 - property names, 24
 - property values, 26
 - resource group names, 24
 - resource names, 24
 - resource type names, 25
- system properties
 - See also* extension properties
 - See also* properties
 - effect on fault monitors, 163
 - Failover_mode, 166
 - Retry_count, 166
 - Retry_interval, 166
 - Thorough_probe_interval
 - effect on restart time, 166
 - tuning, 164

T

- technical support, 12
- Thorough_probe_interval system property
 - effect on restart time, 166
 - tuning, 164
- timeouts
 - fault monitor
 - guidelines for setting, 164–165
- tools
 - clsetup utility, 21
 - Oracle Solaris Cluster Administrative commands, 21
- troubleshooting
 - modification of file systems, 135–136, 136–137

- tuning, fault monitors, 163–166
- Type_version property, 94, 138

U

- unmounting, file systems, 129
- unregistering, resource types, 70–71
- upgrade directive, 25
- upgrading
 - HASStoragePlus resource type, 138–139
 - preregistered resource types, 93–94
 - resource types, 34–35

V

- values, Resource Group Manager (RGM), 26
- verifying
 - addition of file systems to HASStoragePlus resource, 128
 - nsswitch.conf file contents, 15
 - removal of file systems from HASStoragePlus resource, 129
- versions
 - HASStoragePlus resource type, 138
 - SUNW.LogicalHostname resource type, 93
 - SUNW.SharedAddress resource type, 93
- vfstab file
 - adding entries to, 127
 - adding removing entries from, 130
- volume managers, highly available local file systems, 114

W

- weak negative affinity
 - definition, 140
 - usage example, 143–144
- weak positive affinity
 - definition, 140
 - usage example, 142–143

Z

zone cluster, resource group affinities, 147–148

Zpool's property, recovering from a fault, 136–137