

# **Administering the Disaster Recovery Framework for Oracle® Solaris Cluster 4.4**

**ORACLE®**

**Part No: E69319**  
June 2019



## Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4

### Part No: E69319

Copyright © 2004, 2019, 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, then the following notice is applicable:

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

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

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

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

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

### Access to Oracle Support

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

**Référence: E69319**

Copyright © 2004, 2019, Oracle et/ou ses affiliés. Tous droits réservés.

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 stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, 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 livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel 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 un risque de dommages corporels. Si vous utilisez ce logiciel ou ce 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 des applications dangereuses.

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 de 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, sauf mention contraire stipulée dans un contrat entre vous et Oracle. 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, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

**Accès aux services de support Oracle**

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

# Contents

---

<b>Using This Documentation</b> .....	17
<b>1 Introduction to Administering the Disaster Recovery Framework</b> .....	19
Disaster Recovery Framework Administration .....	19
Disaster Recovery Framework Administration Tools .....	20
Oracle Solaris Cluster Manager Browser Interface .....	20
Command-Line Interface .....	21
<b>2 Overview of Administering a Disaster Recovery Framework Configuration</b> .....	23
Overview of Oracle Solaris Cluster Administration Concepts .....	23
Configuring Resources and Resource Groups .....	23
Configuring Logical Hostnames .....	24
Managing Device Groups .....	25
Disaster Recovery Framework Configuration and Administration Tasks .....	25
Example Disaster Recovery Framework Cluster Configuration .....	26
<b>3 Administering the Disaster Recovery Framework</b> .....	29
Disaster Recovery Framework Resource Groups .....	29
Enabling the Disaster Recovery Framework .....	30
▼ How to Enable the Disaster Recovery Framework .....	30
Disabling the Disaster Recovery Framework .....	33
▼ How to Disable the Disaster Recovery Framework .....	33
Checking the Status of the Disaster Recovery Framework .....	36
Booting a Cluster .....	36
<b>4 Administering Rights Profiles</b> .....	39
Modifying a User's Rights .....	39

<b>5 Administering Cluster Partnerships</b> .....	41
Modifying Partnership Properties .....	41
▼ How to Modify Partnership Properties .....	41
Adding a New Cluster Node .....	43
▼ How to Add a New Node to a Cluster in a Partnership .....	43
Renaming a Cluster Node .....	44
Renaming a Cluster That Is in a Partnership .....	44
▼ How to Rename a Cluster That Is in a Partnership .....	44
Removing Trust Between Partner Clusters .....	49
▼ How to Remove Trust Between Two Clusters .....	49
Leaving or Deleting a Partnership .....	50
▼ How to Leave or Delete a Partnership .....	51
Resynchronizing a Partnership .....	52
▼ How to Resynchronize a Partnership .....	52
<b>6 Administering Heartbeats</b> .....	55
Introduction to Heartbeats .....	55
Creating a Heartbeat .....	56
▼ How to Create a Heartbeat .....	56
Creating a Heartbeat Plug-in .....	57
▼ How to Create Heartbeat Plug-in .....	58
Modifying a Heartbeat Plug-in Property .....	58
▼ How to Modify the Properties of a Heartbeat Plug-in .....	59
Deleting Heartbeats and Heartbeat Plug-ins .....	60
▼ How to Delete a Heartbeat .....	60
▼ How to Delete a Plug-in From a Heartbeat .....	61
Displaying Heartbeat Configuration Information .....	61
▼ How to Display Heartbeat Configuration Information .....	62
Tuning the Heartbeat Properties .....	62
▼ How to Modify the Heartbeat Properties .....	63
Creating a Heartbeat That Uses a Custom Heartbeat Plug-in .....	64
Creating a Custom Heartbeat Plug-in .....	65
▼ How to Add a Custom Heartbeat Plug-in to an Existing Default Heartbeat .....	66
▼ How to Create a Custom Heartbeat Plug-in and Add It to a Custom Heartbeat .....	67
Configuring Heartbeat-Loss Notification .....	69

Introduction to Configuring Heartbeat-Loss Notification .....	69
Configuring the Heartbeat-Loss Notification Properties .....	70
Creating an Action Shell Script for Heartbeat-Loss .....	71
<b>7 Administering Protection Groups .....</b>	<b>73</b>
Activating and Deactivating a Protection Group .....	73
Guidelines for Activating and Deactivating a Protection Group .....	73
▼ How to Activate a Protection Group .....	74
▼ How to Deactivate a Protection Group .....	77
Resynchronizing a Protection Group .....	80
▼ How to Resynchronize a Protection Group .....	80
Modifying Protection Groups and Data Replication Components .....	81
▼ How to Modify a Protection Group .....	81
▼ How to Modify a Data Replication Component .....	82
Deleting Protection Groups and Data Replication Components .....	83
▼ How to Delete a Protection Group .....	84
▼ How to Remove a Data Replication Component From a Protection Group .....	85
<b>8 Administering Sites .....</b>	<b>89</b>
Adding a Cluster to a Site .....	89
▼ How to Add a Cluster to a Site .....	89
▼ How to Accept an Invitation to Join a Site .....	91
Changing the Site Role of a Site Cluster .....	93
▼ How to Change a Site Membership Role .....	93
▼ How to Forcibly Change a Site Member to a Site Controller .....	95
Synchronizing Site Configuration Information .....	97
▼ How to Synchronize Site Configuration Information .....	97
Removing a Cluster From a Site .....	98
▼ How to Remove a Cluster From a Site .....	98
▼ How to Remove an Unreachable Cluster From a Site a Cluster .....	100
Deleting a Site .....	101
▼ How to Delete a Site .....	101
<b>9 Administering Multigroups .....</b>	<b>105</b>
Modifying Protection Groups in a Multigroup .....	105

▼ How to Add a Protection Group to a Multigroup .....	105
▼ How to Change the Protection Groups in a Multigroup Dependency Chain .....	107
▼ How to Remove a Protection Group From a Multigroup .....	108
Starting and Stopping Multigroups .....	110
▼ How to Start All Protection Groups in a Multigroup .....	110
▼ How to Stop All Protection Groups in a Multigroup .....	111
Synchronizing Multigroup Configuration Information .....	112
▼ How to Synchronize Multigroup Configuration Information .....	113
Deleting a Multigroup .....	113
▼ How to Delete a Multigroup .....	114
<b>10 Monitoring and Validating the Disaster Recovery Framework .....</b>	<b>115</b>
Monitoring the Runtime Status of the Disaster Recovery Framework .....	115
Viewing the Disaster Recovery Framework Log Messages .....	122
Displaying Configuration Information for Partnerships and Protection Groups .....	124
▼ How to Display Configuration Information About Partnerships .....	124
▼ How to Display Configuration Information About Protection Groups .....	125
<b>11 Migrating Services .....</b>	<b>127</b>
Detecting Cluster Failure .....	127
Detecting Primary Cluster Failure .....	127
Detecting Secondary Cluster Failure .....	128
Migrating Replication Services by Switching Over Protection Groups .....	129
Actions Performed by the Disaster Recovery Framework During a Switchover .....	129
▼ How to Switch Over Replication From the Primary Cluster to the Secondary Cluster .....	131
▼ How to Switch Over a Multigroup .....	132
Forcing a Takeover of a Protection Group .....	133
Actions Performed by the Disaster Recovery Framework During a Takeover .....	134
▼ How to Force Immediate Takeover of a Protection Group by a Secondary Cluster .....	135
▼ How to Take Over a Multigroup .....	136
Recovering a Protection Group to a Cluster .....	138



---

<b>12 Customizing Switchover and Takeover Actions</b> .....	139
Creating a Role-Change Action Script .....	139
Configuring a Protection Group to Run a Script at Switchover or Takeover .....	142
▼ How to Configure a Protection Group to Run a Script at Switchover or Takeover .....	142
<b>13 Script-Based Plug-Ins</b> .....	145
Introduction to Disaster Recovery Framework Script-Based Plug-Ins .....	145
Advantages and Disadvantages of Using Script-Based Plug-Ins .....	146
Script-Based Plug-In Architecture .....	146
Restrictions of Script-Based Plug-Ins .....	148
Creating Script-Based Plug-Ins .....	148
Property Descriptions for Script-Based Plug-Ins .....	149
Protection Group Properties - Overview .....	149
Replicated Component Properties - Overview .....	150
Protection Group Property Descriptions .....	151
Internals for Script-Based Plug-Ins .....	160
Plug-In Script Functional Requirements .....	160
Script-Based Plug-In Replication Resource Groups and Resources .....	162
Protection Group Status Mapped from Replication Resource Status .....	163
How the Disaster Recovery Framework Handles Password Properties .....	163
<b>A Standard Disaster Recovery Framework Properties</b> .....	165
Heartbeat Properties .....	165
Heartbeat Plug-in Properties .....	166
Partnership Properties .....	167
Protection Group Properties .....	168
Site Properties .....	169
Multigroup Properties .....	171
<b>B Legal Names and Values of Disaster Recovery Framework Entities</b> .....	173
Legal Names for Disaster Recovery Framework Entities .....	173
Legal Values for Disaster Recovery Framework Entities .....	174
<b>C Disaster Recovery Administration Example</b> .....	175

<b>D Takeover Postconditions</b> .....	177
Results of a Takeover When the Partner Cluster Can Be Reached .....	177
Results of a Takeover When the Partner Cluster Cannot Be Reached .....	178
<b>E Troubleshooting the Disaster Recovery Framework</b> .....	181
Troubleshooting Monitoring and Logging .....	181
Configuring the Logger File to Avoid Too Many Traces .....	181
Configuring the Logger File to Avoid Detailed Messages From the gcr Agent .....	182
Configuring the Logger File to Avoid jmx Remote Traces .....	182
Troubleshooting Migration Problems .....	182
Resolving Problems With Application Resource Group Failover When Communication Lost With the Storage Device .....	183
Troubleshooting Cluster Start and Restart .....	183
Validating Protection Groups in an Error State .....	183
Administering Stopped Protection Groups After a Cluster Restart .....	184
Restarting the Common Agent Container .....	184
<b>F Error Return Codes for Script-Based Plug-Ins</b> .....	185
Error Codes That Are Returned in Script-Based Plug-Ins .....	185
<b>Index</b> .....	189

## Figures

---

<b>FIGURE 1</b>	Example Cluster Configuration .....	27
<b>FIGURE 2</b>	Script-Based Plug-Ins Framework .....	147
<b>FIGURE 3</b>	Script-Based Plug-In Replication Resource Group .....	148



## Tables

---

<b>TABLE 1</b>	Disaster Recovery Framework CLI .....	21
<b>TABLE 2</b>	IP Addresses Required by the Disaster Recovery Framework .....	24
<b>TABLE 3</b>	Disaster Recovery Framework Administration Tasks .....	25
<b>TABLE 4</b>	Status Value Descriptions .....	118
<b>TABLE 5</b>	Protection Group Global Policies .....	149
<b>TABLE 6</b>	Optional Replicated Component Properties .....	151
<b>TABLE 7</b>	Disaster Recovery Framework Heartbeat Properties .....	165
<b>TABLE 8</b>	Disaster Recovery Framework Heartbeat Plug-in Properties .....	166
<b>TABLE 9</b>	Disaster Recovery Framework Partnership Properties .....	167
<b>TABLE 10</b>	Disaster Recovery Framework Protection Group Properties .....	168
<b>TABLE 11</b>	Disaster Recovery Framework Site Properties .....	170
<b>TABLE 12</b>	Disaster Recovery Framework Multigroup Properties .....	171
<b>TABLE 13</b>	Takeover Results of Running the <code>geopg takeover</code> Command on the Secondary Cluster .....	177
<b>TABLE 14</b>	Takeover Results of Running the <code>geopg takeover</code> Command on the Primary Cluster .....	178
<b>TABLE 15</b>	Takeover Results of Running the <code>geopg takeover</code> Command on the Secondary Cluster When the Primary Cluster Cannot Be Reached .....	179
<b>TABLE 16</b>	Takeover Results of Running the <code>geopg takeover</code> Command on the Primary Cluster When the Secondary Cluster Cannot Be Reached .....	179



## Examples

---

<b>EXAMPLE 1</b>	Enabling the Disaster Recovery Framework in a Cluster .....	32
<b>EXAMPLE 2</b>	Disabling a Cluster .....	34
<b>EXAMPLE 3</b>	Displaying Whether the Disaster Recovery Framework Has Been Enabled .....	36
<b>EXAMPLE 4</b>	Modifying the Properties of a Partnership .....	42
<b>EXAMPLE 5</b>	Renaming a Cluster in a Partnership .....	47
<b>EXAMPLE 6</b>	Leaving a Partnership .....	51
<b>EXAMPLE 7</b>	Deleting a Partnership .....	52
<b>EXAMPLE 8</b>	Resynchronizing a Partnership .....	53
<b>EXAMPLE 9</b>	Creating a Heartbeat .....	57
<b>EXAMPLE 10</b>	Creating a Heartbeat Plug-in .....	58
<b>EXAMPLE 11</b>	Modifying the Properties of the Heartbeat Plug-in .....	60
<b>EXAMPLE 12</b>	Deleting a Heartbeat .....	60
<b>EXAMPLE 13</b>	Deleting a Plug-in From a Heartbeat .....	61
<b>EXAMPLE 14</b>	Displaying Heartbeat Configuration Information .....	62
<b>EXAMPLE 15</b>	Modifying the Properties of the Default Heartbeat .....	64
<b>EXAMPLE 16</b>	Adding a Custom Heartbeat Plug-in to the Default Heartbeat .....	66
<b>EXAMPLE 17</b>	Adding a Custom Heartbeat Plug-in to a New Custom Heartbeat .....	68
<b>EXAMPLE 18</b>	Configuring Heartbeat-Loss Notification for an Existing Partnership .....	70
<b>EXAMPLE 19</b>	How a Notification Action Script Parses the Command-Line Information Provided by the Disaster Recovery Framework .....	72
<b>EXAMPLE 20</b>	Globally Activating a Protection Group .....	77
<b>EXAMPLE 21</b>	Locally Activating a Protection Group .....	77
<b>EXAMPLE 22</b>	Deactivating a Protection Group on All Clusters .....	79
<b>EXAMPLE 23</b>	Deactivating a Protection Group on the Local Cluster .....	79
<b>EXAMPLE 24</b>	Stopping Remote Replication While Leaving the Protection Group Online .....	79
<b>EXAMPLE 25</b>	Deactivating a Protection Group While Keeping Application Resource Groups Online .....	79

<b>EXAMPLE 26</b>	Modifying the Properties of an Oracle ZFS Storage Appliance Remote Replication Component .....	83
<b>EXAMPLE 27</b>	Deleting a Protection Group .....	85
<b>EXAMPLE 28</b>	Deleting a Protection Group While Keeping Application Resource Groups Online .....	85
<b>EXAMPLE 29</b>	Removing a Remote Replication Component From an Oracle ZFS Storage Appliance Protection Group .....	87
<b>EXAMPLE 30</b>	Displaying Partnership Configuration Information .....	125
<b>EXAMPLE 31</b>	Displaying Configuration Information About a Protection Group .....	125
<b>EXAMPLE 32</b>	Forcing a Switchover From the Primary Cluster to the Secondary Cluster .....	132
<b>EXAMPLE 33</b>	Forcing a Takeover by a Secondary Cluster .....	136
<b>EXAMPLE 34</b>	Switchover Action Script for Updating the DNS .....	141
<b>EXAMPLE 35</b>	Configuring a Protection Group to Run a Command at Cluster Switchover or Takeover .....	143



## Using This Documentation

---

- **Overview** – Describes how to modify, manage, monitor, customize, develop, and troubleshoot the Oracle Solaris Cluster disaster recovery framework.
- **Audience** – Experienced system administrators with extensive knowledge of Oracle software and hardware.
- **Required knowledge** – Knowledge of the Oracle Solaris operating system, of Oracle Solaris Cluster software, and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

This document is not to be used as a planning or presales guide.

## Product Documentation Library

Documentation and resources for this product and related products are available at [http://www.oracle.com/pls/topic/lookup?ctx=product\\_intuitive\\_ID](http://www.oracle.com/pls/topic/lookup?ctx=product_intuitive_ID).

## Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.



# ◆◆◆ CHAPTER 1

## Introduction to Administering the Disaster Recovery Framework

---

Oracle Solaris Cluster disaster recovery framework protects applications from unexpected disruptions by using multiple clusters that are geographically separated. These clusters contain identical copies of the disaster recovery framework, which manage replicated data between the clusters. The disaster recovery framework is a layered extension of the Oracle Solaris Cluster software.

This chapter contains the following sections:

- “Disaster Recovery Framework Administration” on page 19
- “Disaster Recovery Framework Administration Tools” on page 20

### Disaster Recovery Framework Administration

Familiarize yourself with the planning information in the *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4* and the *Disaster Recovery Framework Concepts for Oracle Solaris Cluster 4.4* before beginning administration tasks. This guide contains the standard tasks that are used to administer and maintain the disaster recovery framework configurations.

For general Oracle Solaris Cluster, data service, and hardware administration tasks, refer to the Oracle Solaris Cluster documentation.

You can perform all administration tasks on a cluster that is running the disaster recovery framework without causing any nodes or the cluster to fail. You can install, configure, start, use, stop, and uninstall the disaster recovery framework on an operational cluster.

---

**Note** - You might be required to take nodes or the cluster offline for preparatory actions, such as installing data replication software and performing Oracle Solaris Cluster administrative tasks. Refer to the appropriate product documentation for administration restrictions.

---

## Disaster Recovery Framework Administration Tools

You can perform administrative tasks on a cluster that is running disaster recovery framework by using the Oracle Solaris Cluster Manager browser interface or the command-line interface (CLI).

- [“Oracle Solaris Cluster Manager Browser Interface” on page 20](#)
- [“Command-Line Interface” on page 21](#)

The procedures in this guide describe how to perform administrative tasks by using the CLI.

## Oracle Solaris Cluster Manager Browser Interface

The disaster recovery framework supports Oracle Solaris Cluster Manager, a browser interface tool that you can use to monitor disaster recovery framework status and perform various administrative tasks on your cluster. For specific information about how to use Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#) and the Oracle Solaris Cluster Manager online help.

---

**Note** - To administer Oracle Solaris Cluster software by using the Oracle Solaris Cluster Manager browser interface, ensure that the root passwords are the same on all nodes of both clusters in the partnership.

---

You can use Oracle Solaris Cluster Manager to administer the disaster recovery framework only after the disaster recovery framework has been enabled by using the `geoadm start` command. For information about enabling and disabling the disaster recovery framework, see [Chapter 3, “Administering the Disaster Recovery Framework”](#).

Oracle Solaris Cluster Manager does not support creating custom heartbeats outside of a partnership. If you want to specify a custom heartbeat in a partnership join operation, use the CLI to run the `geops join-partnership` command.

To access Oracle Solaris Cluster Manager, do the following:

1. From a browser that is running on a machine that is outside the cluster, go to the following URL from any Java-enabled and Javascript-enabled browser:

```
https://node:8998/scm
```

2. Log in as the root role or a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.

## Command-Line Interface

Table 1, “Disaster Recovery Framework CLI,” on page 21 lists the commands that you can use to administer the disaster recovery framework. For more information about each command, refer to the *Disaster Recovery Framework Reference for Oracle Solaris Cluster 4.4*.

**TABLE 1** Disaster Recovery Framework CLI

Command	Description
geoadm	Enables or disables the disaster recovery framework on the local cluster and displays the runtime status of the local cluster
geohb	Configures and manages the heartbeat mechanism that is provided with the disaster recovery framework
geomg	Configures and manages multigroups of protection groups
geops	Creates and manages the partnerships between clusters
geopp	Configures and manages protection groups
geosite	Configures and manages cluster sites



## ◆◆◆ CHAPTER 2

# Overview of Administering a Disaster Recovery Framework Configuration

---

This chapter describes what you need to know before you begin administering the disaster recovery framework. Here you also learn about the Oracle Solaris Cluster infrastructure that is required by the disaster recovery framework. You also can find here common Oracle Solaris Cluster concepts and tasks you need to understand before administering the disaster recovery framework. This chapter also provides an example configuration that is used throughout this guide to illustrate the common disaster recovery framework administration tasks.

This chapter contains the following sections:

- [“Overview of Oracle Solaris Cluster Administration Concepts” on page 23](#)
- [“Example Disaster Recovery Framework Cluster Configuration” on page 26](#)

## Overview of Oracle Solaris Cluster Administration Concepts

You must be an experienced Oracle Solaris Cluster administrator to administer the disaster recovery framework.

This section describes the following Oracle Solaris Cluster administration topics that you need to understand before you administer the disaster recovery framework:

- [“Configuring Resources and Resource Groups” on page 23](#)
- [“Configuring Logical Hostnames” on page 24](#)
- [“Managing Device Groups” on page 25](#)

## Configuring Resources and Resource Groups

You use either Oracle Solaris Cluster commands or the Oracle Solaris Cluster Manager browser interface to create failover and scalable resource groups.

For more information about administering resources and resource groups in Oracle Solaris Cluster software, see the [Planning and Administering Data Services for Oracle Solaris Cluster 4.4](#).

## Configuring Logical Hostnames

The logical hostname is a special high-availability (HA) resource. The `geoadm start` command configures the logical hostname that corresponds to the cluster name. The IP address and host maps for the logical hostname must be set up before you run this command. Before assigning hostnames, familiarize yourself with the legal names and values that are described in [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

For more information about using the `geoadm start` command, see [“Enabling the Disaster Recovery Framework” on page 30](#).

The following table lists the Oracle Solaris Cluster and disaster recovery framework components that require IP addresses. Add these IP addresses to the following locations:

- All naming services that are being used
- The local `/etc/inet/hosts` file on each cluster node, after you install the Oracle Solaris OS software

**TABLE 2** IP Addresses Required by the Disaster Recovery Framework

Component	Number of IP Addresses Needed
Oracle Solaris Cluster administrative console	1 per subnet
IP Network Multipathing groups	<ul style="list-style-type: none"> <li>■ Single-adapter groups – 1 primary IP address.</li> <li>■ Multiple-adapter groups – 1 primary IP address plus 1 test IP address for each adapter in the group.</li> </ul>
Cluster nodes	1 per node, per subnet
Domain console network interface	1 per domain
Console-access device	1
Logical addresses	1 per logical host resource, per subnet
Disaster recovery framework hostnames	1 logical IP address per cluster infrastructure. For example, if you have two clusters in your disaster recovery framework configuration, you need two IP addresses.

For more information about configuring the IP address and host maps during the installation of Oracle Solaris Cluster software, refer to [Chapter 2, “Installing Software on Global-Cluster Nodes” in \*Installing and Configuring an Oracle Solaris Cluster 4.4 Environment\*](#).



## Managing Device Groups

A device group is a hardware resource that is managed by the Oracle Solaris Cluster software. A device group is a type of global device that is used by the Oracle Solaris Cluster software to register device resources, such as disks. A device group can include the device resources of disks and Solaris Volume Manager disk sets.

For information about configuring device groups in Oracle Solaris Cluster software, refer to [Chapter 5, “Administering Global Devices, Disk-Path Monitoring, and Cluster File Systems” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

The disaster recovery framework configures Oracle Solaris Cluster device groups to include replication.

For more information about configuring data replication in the disaster recovery framework, see the Oracle Solaris Cluster documentation for the data replication product you use.

## Disaster Recovery Framework Configuration and Administration Tasks

The following table lists administration tasks for your disaster recovery framework configuration.

---

**Note** - For procedures to create partnerships, heartbeats, protection groups, sites, and multigroups, see [Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4](#) and the Oracle Solaris Cluster guide for your data replication product.

---

**TABLE 3** Disaster Recovery Framework Administration Tasks

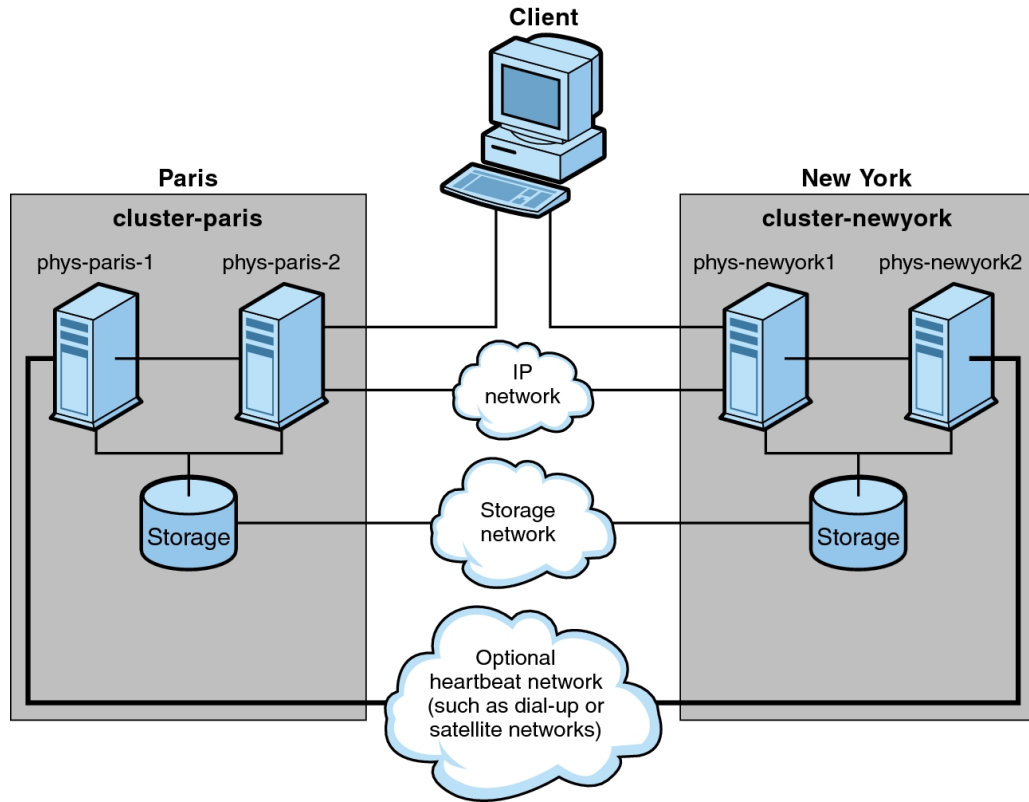
Task	Description and Documentation
Switch over services to the partner cluster.  You can switch over individual protection groups or multigroups.	See the procedures to switch over a protection group from primary to secondary in the Oracle Solaris Cluster guide for your data replication product. You can also use the Oracle Solaris Cluster Manager browser interface.  To switch over multiple protection groups that are configured in a multigroup, see <a href="#">“How to Switch Over a Multigroup” on page 132</a> . You can also use the Oracle Solaris Cluster Manager browser interface.
Take over services from primary to secondary during a disaster.	See the procedures to force immediate takeover of services by a secondary cluster in the Oracle Solaris Cluster guide for your data replication product. You can also use the Oracle Solaris Cluster Manager browser interface.

Task	Description and Documentation
You can take over services for an individual cluster or for a multigroup.	To take over multiple protection groups that are configured in a multigroup, see <a href="#">“How to Take Over a Multigroup” on page 136</a> . You can also use the Oracle Solaris Cluster Manager browser interface.
Recover from a takeover.	<ul style="list-style-type: none"> <li>■ Perform data recovery and error repair outside of the disaster recovery framework. See the Oracle Solaris Cluster documentation for the data replication product you are using.</li> <li>■ Resynchronize the partner clusters. See <a href="#">“Resynchronizing a Protection Group” on page 80</a>. You can also use the Oracle Solaris Cluster Manager browser interface.</li> </ul>
Start an individual protection group or a multigroup.	<p>To start an individual protection group, see <a href="#">“How to Activate a Protection Group” on page 74</a> or use the Oracle Solaris Cluster Manager browser interface.</p> <p>To start a multigroup, see <a href="#">“How to Start All Protection Groups in a Multigroup” on page 110</a> or use the Oracle Solaris Cluster Manager browser interface.</p>
Stop an individual protection group or a multigroup.	<p>To stop an individual protection group, see <a href="#">“How to Deactivate a Protection Group” on page 77</a>. You can also use the Oracle Solaris Cluster Manager browser interface.</p> <p>To stop a multigroup, see <a href="#">“How to Stop All Protection Groups in a Multigroup” on page 111</a>. You can also use the Oracle Solaris Cluster Manager browser interface.</p>
Delete a multigroup	See <a href="#">“Deleting a Multigroup” on page 113</a> . You can also use the Oracle Solaris Cluster Manager browser interface.
Delete a site.	See <a href="#">“Deleting a Site” on page 101</a> . You can also use the Oracle Solaris Cluster Manager browser interface.
Delete a protection group or a data replication component.	See <a href="#">“Deleting Protection Groups and Data Replication Components” on page 83</a> . You can also use the Oracle Solaris Cluster Manager browser interface.
Delete a partnership.	See <a href="#">“Leaving or Deleting a Partnership” on page 50</a> . You can also use the Oracle Solaris Cluster Manager browser interface.
Disable the disaster recovery framework.	See <a href="#">“How to Disable the Disaster Recovery Framework” on page 33</a> .
Uninstall the disaster recovery framework software.	See the <i>Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i> .

## Example Disaster Recovery Framework Cluster Configuration

The following figure describes an disaster recovery framework cluster configuration that is used throughout this guide to illustrate the disaster recovery framework administration tasks. The primary cluster, `cluster-paris`, contains two nodes, `phys-paris-1` and `phys-paris-2`. The secondary cluster, `cluster-newyork`, also contains two nodes, `phys-newyork-1` and `phys-newyork-2`.

FIGURE 1 Example Cluster Configuration





## Administering the Disaster Recovery Framework

---

This chapter contains information about enabling your cluster for participation in a partnership. It also contains information for disabling the disaster recovery framework so that your cluster no longer can participate in partnerships.

This chapter contains the following sections:

- [“Disaster Recovery Framework Resource Groups” on page 29](#)
- [“Enabling the Disaster Recovery Framework” on page 30](#)
- [“Disabling the Disaster Recovery Framework” on page 33](#)
- [“Checking the Status of the Disaster Recovery Framework” on page 36](#)
- [“Booting a Cluster” on page 36](#)

### Disaster Recovery Framework Resource Groups

When you enable the disaster recovery framework, the following Oracle Solaris Cluster resource groups are created:

- `geo-clusterstate` – A scalable resource group that the disaster recovery framework uses to distinguish between node failover and cluster reboot scenarios. The resource group contains the following resources:
  - `geosource geo-init-svc`– Initialization service for the Oracle Solaris Cluster disaster recovery framework.
  - `geo-zc-sysevent` - (For zone clusters only) Runs resource methods in the global zone when the disaster recovery framework is started in a non-global zone. It manages the mechanism which transfers cluster events to subscribers in the zone cluster.
- `geo-infrastructure` – A failover resource group that encapsulates the disaster recovery framework. The resource group contains the following resources:

- `geo-cluster-name` – The logical hostname for the disaster recovery framework. The disaster recovery framework uses the logical hostname of a cluster for intercluster management communication and heartbeat communication. An entry in the naming services must be the same as the name of the cluster and be available on the namespace of each cluster.
- `geo-hbmonitor` – Encapsulates the heartbeat processes for the disaster recovery framework.
- `geo-failovercontrol` – Encapsulates the disaster recovery framework itself. The disaster recovery framework module uses this resource to load into the common agent container.

These resources are for internal purposes only, so you must not change them.

These internal resources are removed when you disable the disaster recovery framework.

You can monitor the status of these resources by using the `clresource status` command. For more information about this command, see the [clresource\(8CL\)](#) man page.

## Enabling the Disaster Recovery Framework

When you enable the disaster recovery framework, the cluster is ready to enter a partnership with another enabled cluster. You can use the CLI commands or Oracle Solaris Cluster Manager to create a cluster partnership.

For more information about setting up and installing the disaster recovery framework software, see the [Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4](#).

### ▼ How to Enable the Disaster Recovery Framework

This procedure enables the disaster recovery framework on the local cluster only. Repeat this procedure on all the clusters of your geographically separated cluster.

**Before You Begin** Ensure that the following conditions are met:

- The cluster is running the Oracle Solaris Operating System and the Oracle Solaris Cluster software.
- The disaster recovery framework software is installed.

- The cluster has been configured for secure cluster communication by using security certificates, that is, nodes within the same cluster must share the same security certificates. This is done during Oracle Solaris Cluster installation.

**1. Log in to a cluster node.**

You must be assigned the Geo Operation rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Ensure that the logical hostname, which is the same as the cluster name, is available and defined.**

```
# cluster list
```

**3. If the cluster name is not the name you want to use, change the cluster name.**

If you must change the name of a cluster that is configured in a partnership, do not perform this step. Instead, follow instructions in [“Renaming a Cluster That Is in a Partnership”](#) on page 44.

Follow cluster naming guidelines as described in [“Planning Required IP Addresses and Hostnames”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*. Cluster names must follow the same requirements as for host names. You cannot rename a cluster that is part of a zone cluster, but you can rename a cluster in a global zone.

```
# cluster rename -c new-cluster-name old-cluster-name
```

For more information, see the `cluster(8CL)` man page.

---

**Note** - After you have enabled the disaster recovery framework, you must not change the cluster name while the framework is enabled.

---

**4. Confirm that the naming service and the local hosts files contain a host entry that matches the cluster name.**

The local host file, `hosts`, is located in the `/etc/inet` directory.

**5. On a node of the cluster, start the disaster recovery framework.**

```
# geoadm start
```

The `geoadm start` command enables the disaster recovery framework on the local cluster only. For more information, see the `geoadm(8)` man page.

**6. Verify that you have enabled the disaster recovery framework and that the disaster recovery framework resource groups are online.**

For a list of the disaster recovery framework resource groups, see [“Disaster Recovery Framework Resource Groups” on page 29](#).

```
# geoadm show
# clresource-group status
# clresource status
```

The output for the `geoadm show` command displays that the disaster recovery framework is active from a particular node in the cluster.

The output for the `clresource-group status` and `clresource status` commands displays that the `geo-failovercontrol`, `geo-hbmonitor`, and `geo-cluster-name` resources and the `geo-infrastructure` resource groups are online on one node of the cluster.

For more information, see the [clresourcegroup\(8CL\)](#) and [clresource\(8CL\)](#) man pages.

**Example 1** Enabling the Disaster Recovery Framework in a Cluster

This example enables the disaster recovery framework on the `cluster-paris` cluster.

1. Start the disaster recovery framework on `cluster-paris`.

```
phys-paris-1# geoadm start
```

2. Ensure that the disaster recovery framework was successfully enabled.

```
phys-paris-1# geoadm show
```

```
--- CLUSTER LEVEL INFORMATION ---
Oracle Solaris Cluster disaster recovery framework is active on cluster-paris from
node phys-paris-1
Command execution successful
phys-paris-1#
```

3. Verify the status of the disaster recovery framework resource groups and resources.

```
phys-paris-1# clresource-group status
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
geo-clusterstate	phys-paris-1	No	Online
	phys-paris-2	No	Online
geo-infrastructure	phys-paris-1	No	Online
	phys-paris-2	No	Offline



```
# clresource status
=== Cluster Resources ===

Resource Name      Node Name      State          Status Message
-----
geo-cluster-name   phys-paris-1   Online         Online - LogicalHostname
online.
                   phys-paris-2   Offline        Offline

geo-hbmonitor      phys-paris-1   Online         Online - Daemon OK
                   phys-paris-2   Offline        Offline

geo-failovercontrol phys-paris-1   Online         Online - Service is online
                   phys-paris-2   Offline        Offline

geo-servicetag     phys-paris-1   Online_not_monitored Online_not_monitored
                   phys-paris-1   Offline        Offline
```

**Next Steps** For information about creating protection groups, see the Oracle Solaris Cluster data replication guide that corresponds to the type of data replication software you are using.

## Disabling the Disaster Recovery Framework

You can disable the disaster recovery framework by using the following procedure.

### ▼ How to Disable the Disaster Recovery Framework

**Before You Begin** Ensure that all protection groups on the local cluster are offline. For more information, see the following procedures:

- To stop an individual protection group, see [“How to Deactivate a Protection Group” on page 77](#).
- To stop a set of protection groups that are configured in a multigroup, see [“How to Stop All Protection Groups in a Multigroup” on page 111](#).



---

**Caution** - If you want to keep the application resource groups online while deactivating a protection group, follow [Example 25, “Deactivating a Protection Group While Keeping Application Resource Groups Online,”](#) on page 79 in “How to Deactivate a Protection Group” on page 77.

---

**1. Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Confirm that all of the protection groups are offline on the local cluster.**

```
phys-paris-1# geoadm status
```

For more information about the `geoadm status` command and its output, see “[Monitoring the Runtime Status of the Disaster Recovery Framework](#)” on page 115.

**3. Disable the disaster recovery framework.**

```
phys-paris-1# geoadm stop
```

This command removes the disaster recovery framework resource groups that were created when you enabled the disaster recovery framework.

For more information about this command, see the [geoadm\(8\)](#) man page.

---

**Note** - Disabling the disaster recovery framework removes only the disaster recovery framework resource groups. Resource groups that have been created to support data replication are not removed unless you remove the protection group that the resource groups are supporting by using the `geopg delete` command.

---

**4. Verify that the software was disabled and that the disaster recovery framework resource groups are no longer displayed.**

```
phys-paris-1# geoadm show
phys-paris-1# clresource-group status
```

For more information, see the [clresourcegroup\(8CL\)](#) man page.

**Example 2** Disabling a Cluster

This example disables the `cluster-paris` cluster.

1. Confirm that all protection groups are offline.

```
phys-paris-1# geoadm status
```

```
Cluster: cluster-paris
```

```
Partnership "paris-newyork-ps" :OK
```

```
  Partner clusters :cluster-newyork
```

```
  Synchronization :OK
```

```
  ICRM Connection :OK
```

```
Heartbeat "paris-to-newyork" monitoring "cluster-newyork":OK
```

```
Heartbeat plug-in "ping_plugin" :Inactive
```

```
Heartbeat plug-in "tcp_udp_plugin":OK
```

```
Protection group "tcpg" :OK
```

```
  Partnership :paris-newyork-ps
```

```
  Synchronization :OK
```

```
Cluster cluster-paris :OK
```

```
  Role :Primary
```

```
  PG activation state :Deactivated
```

```
  Configuration :OK
```

```
  Data replication :OK
```

```
  Resource groups :OK
```

```
Cluster cluster-newyork :OK
```

```
  Role :Secondary
```

```
  PG activation state :Deactivated
```

```
  Configuration :OK
```

```
  Data replication :OK
```

```
  Resource groups :OK
```

2. Disable the disaster recovery framework.

```
phys-paris-1# geoadm stop
```

```
... verifying pre conditions and performing pre remove operations ... done
```

```
...removing product infrastructure ... please wait ...
```

3. Confirm that the disaster recovery framework was successfully disabled.

```
phys-paris-1# geoadm show
```

```
--- CLUSTER LEVEL INFORMATION ---
```

```
Oracle Solaris Cluster disaster recovery framework is not active on cluster-paris
```

```
--- LOCAL NODE INFORMATION ---  
Node phys-paris-1 does not host active product module.
```

```
Command execution successful  
phys-paris-1#
```

4. Verify that disaster recovery framework resource groups and resources have been removed.

```
phys-paris-1# clresource-group status  
phys-paris-1#
```

## Checking the Status of the Disaster Recovery Framework

Use the `geoadm show` command to determine whether the disaster recovery framework is enabled on the local cluster and on which node the disaster recovery framework is active. The disaster recovery framework is considered active on the node on which the `geo-infrastructure` resource group has a state of `Online`.

You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Partnerships. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

### EXAMPLE 3 Displaying Whether the Disaster Recovery Framework Has Been Enabled

This example displays information on the `phys-paris-1` node of the `cluster-paris` cluster.

```
phys-paris-1# geoadm show  
  
--- CLUSTER LEVEL INFORMATION ---  
Oracle Solaris Cluster disaster recovery framework is active on:  
node phys-paris-2, cluster cluster-paris  
  
Command execution successful  
phys-paris-1#
```

## Booting a Cluster

The following events take place when you boot a cluster:

1. After the disaster recovery framework is enabled, the disaster recovery framework starts automatically. Verify that the software started successfully by using the `geoadm show` command.
2. The heartbeat framework checks which partners it can reach.
3. Check the current status of the cluster by using the `geoadm status` command. For more information about this command and its output, see [“Monitoring the Runtime Status of the Disaster Recovery Framework” on page 115](#).



## Administering Rights Profiles

---

This chapter describes how to administer role-based access control (RBAC) rights in relation to the disaster recovery framework. For information about rights for the disaster recovery framework, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### Modifying a User's Rights

---

**Note** - The rights profiles for Oracle Solaris Cluster software, including the disaster recovery framework, are intended to simplify the assignment of management roles and protect against accidental errors. However, a malicious user would be able to abuse rights-based cluster administration privileges to gain wider system privileges. Therefore, user rights should be assigned with care.

---

When you grant authorization to users other than the root role, you must do so on all nodes of both partner clusters. If the cluster is in a site, the update must be done on all nodes of all clusters in the site. Otherwise, operations that have a global scope might fail, due to insufficient user rights on one or more nodes in the partnership or site.

To modify the rights for a user, you must be logged in as the root role or assume a role that is assigned the System Administrator rights profile.

For example, you can assign the Geo Management rights profile to the user admin as follows:

```
# usermod -P "Geo Management" admin
# profiles admin
Geo Management
Basic Solaris User
#
```

For more information about how to modify the rights properties for a user, refer to [Chapter 2, “Oracle Solaris Cluster and User Rights”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.



## Administering Cluster Partnerships

---

This chapter provides the procedures for administering partnerships between two disaster recovery framework enabled clusters.

This chapter contains the following sections:

- “Modifying Partnership Properties” on page 41
- “Adding a New Cluster Node” on page 43
- “Renaming a Cluster Node” on page 44
- “Renaming a Cluster That Is in a Partnership” on page 44
- “Removing Trust Between Partner Clusters” on page 49
- “Leaving or Deleting a Partnership” on page 50
- “Resynchronizing a Partnership” on page 52

### Modifying Partnership Properties

This section provides procedures to modify a property of a partnership.

#### ▼ How to Modify Partnership Properties

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the name of the partnership to go to its page, click the Properties tab, and click Edit to access the listed properties for editing. For Oracle Solaris Cluster Manager log-in instructions, see “[How to Access Oracle Solaris Cluster Manager](#)” in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

1. **Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

## 2. Modify partnership properties.

```
# geops set-prop -p property-setting [-p...] partnership-name
```

*-p property-setting*

Specifies the value of partnership properties with a string of *property=value* pair statements.

Specify a description of the partnership with the `Description` property.

You can configure heartbeat-loss notification with the `Notification_emailaddr`s and `Notification_actioncmd` properties. For more information about configuring heartbeat-loss notification, see [“Configuring Heartbeat-Loss Notification”](#) on page 69.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*partnership-name*

Specifies the name of the partnership.

For information about the names and values that are supported by the disaster recovery framework, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

For more information about the `geops` command, refer to the [geops\(8\)](#) man page.

## 3. Verify that your modification was made correctly.

```
# geops list
```

### Example 4 Modifying the Properties of a Partnership

This example modifies the notification email address for the `cluster-paris` cluster.

```
# geops set-prop -p Notification_emailaddrs=operations@example.com \  
paris-newyork-ps  
# geops list
```

## Adding a New Cluster Node

When you add a new node to a cluster that is in a partnership, you must perform additional tasks on that node to make it an active participant in the disaster recovery framework configuration.

### ▼ How to Add a New Node to a Cluster in a Partnership

Perform all steps from the new node.

**1. Add the new node to the cluster.**

Follow procedures in [Chapter 8, “Administering Cluster Nodes”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

**2. Install disaster recovery framework, data replication, and application software on the new node.**

- To install disaster recovery framework software, see [Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4](#).
- To install data replication and application software, see the appropriate guide for the software that you use.

**3. If the cluster with the new node is the primary for any activated protection groups, remove application resource groups from those protection groups.**

This step is necessary to avoid application downtime.

```
# geopg remove-resource-group resource-group protection-group
```

**4. Deactivate all protection groups that are active on this cluster locally.**

```
# geopg stop -e local protection-group
```

**5. Stop the disaster recovery framework.**

```
# geoadm stop
```

**6. Re-enable the disaster recovery framework.**

This action recreates each disaster recovery framework resource group and adds all nodes in the cluster, including the new node, to the node list.

```
# geoadm start
```

7. **Reactivate the protection groups that you deactivated in [Step 4](#).**

```
# geopg start -e local protection-group
```

8. **Restore any application resource groups that you removed in [Step 3](#).**

```
# geopg add-resource-group resource-group protection-group
```

## Renaming a Cluster Node

You can rename a node in a disaster recovery framework cluster that is in a partnership of an Oracle Solaris Cluster configuration.

If the cluster where you are performing the rename procedure is primary for the protection group, and you want to have the application in the protection group online, you can switch the primary group to a secondary during the rename procedure.

For instructions on renaming a node in a disaster recovery framework cluster, see [“How to Rename a Node” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#). The procedure includes instructions that pertain to a node in a disaster recovery framework configuration.

## Renaming a Cluster That Is in a Partnership

When you rename a cluster that is in a partnership, the partnership becomes invalid. You must fully unconfigure the existing partnership and create a new one that uses the new cluster name.

### ▼ How to Rename a Cluster That Is in a Partnership

This procedure demonstrates how to rename one of the global clusters that is in a partnership. You can rename more than one of the clusters at the same time.

---

**Note** - You cannot use this procedure to rename a zone cluster in a partnership.

---

If the cluster that you rename belongs to more than one partnership, perform each step on all clusters that share a partnership with the cluster to rename, before you proceed to the next step in the procedure.

1. **From one node of the cluster that you are renaming, remove resource groups from each protection group that the cluster belongs to.**

This task avoids production application downtime.

```
# geopg remove-resource-group application-resource-group protection-group
```

2. **From one node of each cluster in a protection group, confirm that application resource groups have been removed.**

```
# geopg list protection-group
```

3. **From one node of the cluster that you are renaming, stop each protection group globally.**

This task stops data replication.

```
# geopg stop protection-group -e global
```

4. **From one node of each cluster in a protection group, delete the protection group.**

```
# geopg delete protection-group
```

5. **From one node of each cluster in a partnership, leave the partnership.**

```
# geops leave-partnership partnership
```

6. **From one node of each cluster, confirm that the protection group and the partnership have been removed.**

```
# geoadm status
```

7. **From one node of each cluster, disable the disaster recovery framework.**

```
# geoadm stop
```

8. **From one node of each cluster, confirm that the disaster recovery framework was disabled.**

Verify that the geo-infrastructure, geo-clusterstate, and data-replication resource groups are deleted.

```
# clrg list  
# geoadm status
```

**9. From one node of the cluster that you are renaming, change the cluster name.**

Follow cluster naming guidelines as described in “[Planning Required IP Addresses and Hostnames](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

```
# cluster rename -c new-cluster-name
```

---

**Note** - The name of the cluster must not include the domain. If a partnership contains clusters that are in different domains, you specify the domain to administrative commands, when necessary, by appending the domain name to the cluster name as *cluster.domain*. Only certain disaster recovery framework administrative commands require this fully qualified name when clusters in a partnership are not in the same domain.

---

**10. Confirm that the cluster name is changed.**

```
# cluster list
```

**11. On each node of both clusters, ensure that hostname entries that match the new cluster name are free and are added to the local `/etc/inet/hosts` files.**

If clusters in the partnership are in different domains, include the domain in the `/etc/hosts` entry for each cluster.

```
# ping new-cluster
    There should be no response
# echo "IP-address new-cluster" >> /etc/inet/hosts
```

**12. From one node of each cluster, start the disaster recovery framework.**

```
# geoadm start
```

If the disaster recovery framework fails to start, and the failure is not due to problems with the new logical host, restart the common agent container on all nodes by using the `cacoadm restart` command, then start the disaster recovery framework.

**13. From one node of each cluster, verify that the disaster recovery framework is successfully started.**

```
# geoadm status
```

**14. From one node of each cluster, add trust between the clusters.**

```
# geops add-trust -c remote-partner-cluster[.domain-name]
```

**15. From one node of each cluster, confirm that trust is added successfully.**

```
# geops verify-trust -c remote-partner-cluster[.domain-name]
```

**16. Create and join a new partnership between the clusters.****a. From the primary cluster, create the partnership.**

```
# geops create -c remote-partner-cluster[.domain-name] partnership
```

**b. From the secondary cluster, join the partnership.**

```
# geops join-partnership remote-partner-cluster[.domain-name] partnership
```

**17. On each cluster, confirm that the new partnership is successfully created and joined.**

```
# geoadm status
```

**18. If you did not reboot the nodes of the cluster that you renamed, restart the heartbeats on each node of the renamed cluster.**

Restarting the heartbeat initiates the heartbeat to read and store the new cluster name.

```
# svcadm disable svc:/system/cluster/gchb_resd:default
# svcadm enable svc:/system/cluster/gchb_resd:default
```

**Example 5** Renaming a Cluster in a Partnership

This example renames the cluster `newyork`, in the `paris-newyork-ps` partnership, to `chicago`. The names of the nodes in this cluster are not changed, so `phys-newyork-1` becomes a node in the newly named `chicago` cluster. The `paris-newyork-ps` partnership is first unconfigured. After the cluster is renamed, a new `paris-chicago-ps` partnership is created with the `chicago` cluster as primary and the `paris` cluster as secondary. The two clusters belong to the same domain, so the domain name is not specified to the commands.

```
phys-newyork-1# geopg remove-resource-group app-rg
```

```
phys-newyork-1# geopg list examplepg
```

```
phys-paris-1# geopg list examplepg
```

```
phys-newyork-1# geopg stop examplepg -e global
```

```
phys-newyork-1# geopg delete examplepg
```

```
phys-paris-1# geopg delete examplepg
```

```
phys-newyork-1# geops leave-partnership paris-newyork-ps
```

```
phys-paris-1# geops leave-partnership paris-newyork-ps
```

```
phys-newyork-1# geoadm stop
```

```
phys-paris-1# geoadm stop

phys-newyork-1# clrg list
phys-newyork-1# geoadm status
phys-paris-1# clrg list
phys-paris-1# geoadm status

phys-newyork-1# cluster rename -c chicago
phys-newyork-1# cluster list

phys-newyork-1# ping chicago
phys-newyork-1# echo "192.168.10.1 chicago" >> /etc/hosts
    repeat on each node of the chicago cluster

phys-paris-1# ping chicago
phys-paris-1# echo "192.168.20.1 chicago" >> /etc/hosts
    repeat on each node of the paris cluster

phys-newyork-1# geoadm start
phys-paris-1# geoadm start

phys-newyork-1# geoadm status
phys-paris-1# geoadm status

phys-newyork-1# geops add-trust -c paris
phys-paris-1# geops add-trust -c chicago

phys-newyork-1# geops verify-trust -c paris
phys-paris-1# geops verify-trust -c chicago

phys-newyork-1# geops create -c paris paris-chicago-ps
phys-paris-1# geops join-partnership chicago paris-chicago-ps

phys-newyork-1# geoadm status
phys-paris-1# geoadm status

phys-newyork-1# /etc/init.d/initgchb_resd stop
phys-newyork-1# /etc/init.d/initgchb_resd start
    repeat on each node of the chicago cluster

phys-paris-1# svcadm disable svc:/system/cluster/gchb_resd:default
phys-paris-1# svcadm enable svc:/system/cluster/gchb_resd:default
    repeat on each node of the paris cluster
```

**Next Steps** Perform the following tasks:

- Create a new protection group and replicate it to partner.
- Add device groups.



- Start globally.
- Add resource groups to the protection group and verify the configuration.

---

**Note** - When you create the new protection group, pay close attention to which cluster is the primary and which is the secondary, to ensure that data replication is started in the desired direction.

---

Follow procedures in the appropriate Oracle Solaris Cluster data-replication guide.

## Removing Trust Between Partner Clusters

This section provides procedures to remove trust between partner clusters.

### ▼ How to Remove Trust Between Two Clusters

Perform this procedure to remove trust from between two clusters.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, highlight the name of the partnership, and click Remove Partner Trust. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**Before You Begin** Ensure that the following conditions are met:

- The cluster on which you want to remove trust is running.
  - The cluster name of the partner cluster is known.
  - The host information of the partner cluster must be defined in the local host file. The local cluster needs to know how to reach the partner cluster by name.
1. **Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).
  2. **If there is a partnership configured between the two clusters, dissolve that partnership.**

Run the following command on both clusters:

```
# geops leave
```

3. **On all nodes of both clusters, remove all keys for the remote cluster from the truststore file on the local node.**

```
# geops remove-trust -c remote-partner-cluster-name
```

Perform this step on all the nodes of the local cluster, and then repeat this step on all nodes of the partner cluster.

```
-c remote-partner-cluster-name
```

Specifies the logical hostname of the cluster from which you want to remove the keys. The name for the remote cluster must be identical to the cluster name you specified when adding trust with the `geops add-trust` command. You do not need to specify the fully qualified name if the remote cluster is reachable by partial name.

When you use this option with the `add-trust` or `remove-trust` subcommand, the option specifies the alias where the public keys on the remote cluster are stored. An alias for certificates on the remote cluster has the following pattern:

```
remote-partner-cluster-name.certificate[0-9]*
```

Keys and only keys that belong to the remote cluster should have their alias match this pattern.

For more information about the `geops` command, refer to the [geops\(8\)](#) man page.

4. **Repeat the preceding steps on a node of the remote partner cluster.**

## Leaving or Deleting a Partnership

This section provides procedures to leave or delete a partnership. Because the `geops leave-partnership` command destroys the local partnership configuration information, when the last member leaves a partnership, the partnership no longer exists.

You can alternatively use the `geops` command to remove a cluster from a partnership and release all the resources that are associated with the partnership.

## ▼ How to Leave or Delete a Partnership

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, highlight the name of the partnership, and click Leave Partnership. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

**Before You Begin** Ensure that the following conditions are met:

- The local cluster is a member of the partnership you want to leave.
- This partnership does not contain any protection groups.

**1. Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Verify that the partnership does not contain any protection groups.**

```
# geopp list
```

If you find that the partnership contains protection groups, you can delete them with the `geopp delete` command. For information about deleting protection groups, see [“Deleting Protection Groups and Data Replication Components”](#) on page 83.

**3. Remove the partnership on a node of the cluster that is a member of the partnership.**

```
# geops leave-partnership partnership-name
```

*partnership-name*

Specifies the name of the partnership

---

**Note** - The `geops leave-partnership` command deletes the heartbeats configured for the partnership, including custom heartbeats.

---

For more information, refer to the [geops\(8\)](#) man page.

**Example 6** Leaving a Partnership

In this example, the `cluster-paris` cluster leaves the `paris-newyork-ps` partnership.

```
phys-paris-1# geops leave-partnership paris-newyork-ps
```

**Example 7** Deleting a Partnership

After the `cluster-paris` cluster leaves the `paris-newyork-ps` partnership, as described in the previous example, the only remaining member of the partnership is the `cluster-newyork` cluster. You can delete the `paris-newyork-ps` partnership by forcing the `cluster-newyork` cluster to leave the partnership.

```
phys-newyork-1# geops leave-partnership paris-newyork-ps
```

**Next Steps** Repeat this procedure on the other cluster in the partnership.

## Resynchronizing a Partnership

This section provides information about resynchronizing partnership information between its partner clusters.

Partner clusters that become disconnected during a disaster situation might force the administrator to perform a takeover for a protection group that the partners share. When both clusters are brought online again, both partner clusters might report as the primary of the protection group. You must resynchronize the configuration information of the local protection group with the configuration information that is retrieved from the partner cluster.

If a cluster that is a member of a partnership fails, when the cluster restarts, it detects whether the partnership parameters have been modified while it was down. You decide which partnership configuration information you want to keep: the information on the cluster that failed or the information on the failover cluster. Then, resynchronize the configuration of the partnership accordingly.

You do not need to resynchronize the configuration information in the following situations if the original secondary cluster goes down and resumes operation later.

Use the `geoadm status` command or Oracle Solaris Cluster Manager to check whether you need to resynchronize a partnership. If the `Configuration` status is `Synchronization Status Error`, you need to synchronize the partnership. If the `Local` status is `Partnership Error`, do not resynchronize the partnership. Instead, wait until a heartbeat exchange occurs.

### ▼ How to Resynchronize a Partnership

Perform this procedure to resynchronize partnership configuration information.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, highlight the name of the partnership, and click Update Partnership. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

**Before You Begin** Ensure that the following conditions are met:

- The local cluster is disaster recovery framework enabled.
- The local cluster was an active member of the partnership before failing.




---

**Caution** - Resynchronizing a partnership overwrites the partnership configuration on the cluster where the command is run with the information from the partner cluster.

---

**1. Log in to a node on the cluster that needs to be synchronized with the information retrieved from the partner cluster.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Resynchronize the partnership.**

```
# geops update partnership-name
```

*partnership-name*

Specifies the name of the partnership

**Example 8** Resynchronizing a Partnership

This example resynchronizes a partnership.

```
# geops update paris-newyork-ps
```



## Administering Heartbeats

---

The disaster recovery framework uses heartbeats over the public network as a way for the individual clusters participating in partnerships to detect cluster failures at partner sites. The heartbeat monitor uses plug-in modules to query the heartbeat status of its partners.

This chapter contains the following sections:

- [“Introduction to Heartbeats” on page 55](#)
- [“Creating a Heartbeat” on page 56](#)
- [“Creating a Heartbeat Plug-in” on page 57](#)
- [“Modifying a Heartbeat Plug-in Property” on page 58](#)
- [“Deleting Heartbeats and Heartbeat Plug-ins” on page 60](#)
- [“Displaying Heartbeat Configuration Information” on page 61](#)
- [“Tuning the Heartbeat Properties” on page 62](#)
- [“Creating a Heartbeat That Uses a Custom Heartbeat Plug-in” on page 64](#)
- [“Configuring Heartbeat-Loss Notification” on page 69](#)

### Introduction to Heartbeats

A heartbeat in the disaster recovery framework is a container for a collection of heartbeat plug-ins. A heartbeat has a name and one property that you can tune, `Query_interval`. The `Query_interval` property specifies the delay between heartbeat status requests.

The heartbeat plug-in facilitates the actual physical monitoring activity. The plug-in is defined by a required query command or query library, an optional requester and responder agent, a type, and a `Plugin_properties` string.

The disaster recovery framework product provides the following default plug-ins:

- `tcp_udp_plugin`- Performs a simple heartbeat check on the cluster logical host IP address. If `tcp_udp_plugin` cannot use UDP port 2084, the plug-in tries to use TCP port 2084.

---

**Note** - The Internet Assigned Numbers Authority (IANA) has officially assigned port number 2084 for use by the disaster recovery framework heartbeats.

---

- `ping_plugin`- Pings the cluster logical hostname on the remote cluster.

A default heartbeat that uses the default heartbeat plug-ins is created every time you run `geops create` or `geops join` without specifying a custom heartbeat. The name of the default heartbeat is `hb_local-cluster-name~remote-partner-cluster-name`. For more information about the `geops` command, refer to the [geops\(8\)](#) man page.

You can create custom heartbeat plug-ins and associate them with existing default heartbeats or with new custom heartbeats.

---

**Note** - Custom heartbeats are provided for special circumstances and require careful configuration. Consult your Oracle specialist for assistance if your system requires the use of custom heartbeats.

If you create a custom heartbeat, you must add at least one plug-in to prevent the partnership from remaining in degraded mode.

---

## Creating a Heartbeat

This section describes procedures for creating heartbeats.

### ▼ How to Create a Heartbeat

Use this procedure to create a new heartbeat. To use the heartbeat with a partnership, you must create the heartbeat before you create a partnership. If you create a partnership before you create the custom heartbeat, the default heartbeat that is used by the partnership will prevent the custom heartbeat from being created.

If you create a custom heartbeat, you must add at least one plug-in to prevent the partnership from remaining in degraded mode.

A custom heartbeat prevents the default heartbeat from being used during partnership creation. If you want to use the default heartbeat for your partnership, you must delete the custom heartbeat before running the `geops create` command.



### 1. Log in to a cluster node.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### 2. Create the heartbeat.

```
# geohb create -r remote-partner-cluster-name \
[-p property-setting [-p...]] heartbeat-name
```

*-r remote-partner-cluster-name*

Specifies the name of the remote, secondary partner cluster.

*-p property-setting*

Specifies a heartbeat property that is assigned a value by using a *name=statement* pair. Multiple properties might be set at one time by using multiple statements.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*heartbeat-name*

Specifies an identifier for the heartbeat.

If you create a custom heartbeat, you must add at least one plug-in to prevent the partnership from remaining in degraded mode.




---

**Caution** - The name of the custom heartbeat on each cluster in the same partnership must be different. Choose a name that identifies the heartbeat uniquely, such as `paris-to-newyork` on the cluster `cluster-paris` and `newyork-to-paris` on cluster `cluster-newyork`.

---

For more information about the `geohb` command, refer to the [geohb\(8\)](#) man page.

#### Example 9 Creating a Heartbeat

This example creates a heartbeat that is named `paris-to-newyork`.

```
# geohb create -r cluster-newyork paris-to-newyork
```

## Creating a Heartbeat Plug-in

This section describes procedures for creating a heartbeat plug-in.

## ▼ How to Create Heartbeat Plug-in

### 1. Log in to a cluster node.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### 2. Add the heartbeat plug-in to an existing heartbeat.

```
# geohb add-plugin heartbeat-name plug-in-name [-p property-setting [-p...]]
```

*heartbeat-name*

Specifies the identifier for heartbeat on the local cluster.

*plug-in-name*

Specifies the name of the heartbeat plug-in.

*-p property-setting*

Specifies a heartbeat plug-in property that is assigned a value by using a *name=statement* pair. Multiple properties might be set at one time by using multiple statements.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

For more information about the geohb command, refer to the [geohb\(8\)](#) man page.

#### Example 10 Creating a Heartbeat Plug-in

This example creates a heartbeat plug-in that is named `command1`.

```
# geohb add-plugin paris-to-newyork command1 -p Query_cmd=/usr/bin/hb/
```

## Modifying a Heartbeat Plug-in Property

This section describes procedures for modifying heartbeat plug-in properties. When you modify a plug-in property, your changes take effect immediately.

## ▼ How to Modify the Properties of a Heartbeat Plug-in

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, click the heartbeat name to go to its page, click the plug-in name to go to its page, click the Properties tab, and click Edit to access the listed properties for editing. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

### 1. Log in to a cluster node.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

### 2. Modify the heartbeat plug-in properties.

```
# geohb modify-plugin -p property-setting [-p...] plug-in-name heartbeat-name
```

*heartbeat-name*

Specifies an identifier for the heartbeat.

*plug-in-name*

Specifies the name of the heartbeat plug-in.

-p *property-setting*

Specifies a heartbeat plug-in property that is assigned a value by using a *name=statement* pair. Multiple properties might be set at one time by using multiple statements.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

---

**Note** - You cannot edit some properties of the default plug-ins.

---

For information about the names and values that are supported by the disaster recovery framework, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

For more information about the geohb command, refer to the [geohb\(8\)](#) man page.

**Example 11** Modifying the Properties of the Heartbeat Plug-in

This example modifies the settings of the default TCP/UDP plug-in, `tcp_udp_plugin`, to use only TCP.

```
# geohb modify-plugin -p Plugin_properties=paris-cluster/TCP/2084 \  
tcp_udp_plugin hb_cluster-paris~cluster-newyork
```

## Deleting Heartbeats and Heartbeat Plug-ins

This section provides the following procedures:

- [“How to Delete a Heartbeat” on page 60](#)
- [“How to Delete a Plug-in From a Heartbeat” on page 61](#)

### ▼ How to Delete a Heartbeat

**1. Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

**2. Delete the heartbeat.**

```
# geohb delete heartbeat-name
```

*heartbeat-name*

Specifies an identifier for the heartbeat settings.

For more information about the `geohb` command, refer to the [geohb\(8\)](#) man page.

**Example 12** Deleting a Heartbeat

This example deletes a heartbeat that is named `paris-to-newyork`.

```
# geohb delete paris-to-newyork
```

## ▼ How to Delete a Plug-in From a Heartbeat

### 1. Log in to a cluster node.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### 2. Remove the plug-in from the heartbeat.

```
# geohb remove-plugin plug-in-name heartbeat-name
```




---

**Caution** - Do not delete the default heartbeat plug-ins `tcp_upd_plugin` and `ping_plugin`.

---

*plug-in-name*

Specifies the name of the custom heartbeat plug-in

*heartbeat-name*

Specifies an identifier for the heartbeat that contains this plug-in

For information about the names and values that are supported by the disaster recovery framework, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

For more information about the `geohb` command, refer to the [geohb\(8\)](#) man page.

#### Example 13 Deleting a Plug-in From a Heartbeat

This example removes the plug-in that is named `command1` from the heartbeat that is named `paris-to-newyork`.

```
# geohb remove-plugin command1 paris-to-newyork
```

## Displaying Heartbeat Configuration Information

This section describes procedures for displaying heartbeat configuration information.

## ▼ How to Display Heartbeat Configuration Information

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, click the heartbeat name to go to its page, click the Properties tab, and click Edit to access the listed properties for editing. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

**1. Log in a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Display the current configuration information for a specific heartbeat or the whole heartbeat subsystem.**

```
# geohb list [heartbeat-name-list]
```

*heartbeat-name-list*

Specifies the names of the heartbeats on the local cluster for which configuration information should be displayed.

If you do not specify a list of heartbeat names, this command displays information about all the configured heartbeats.

For more information about the `geohb` command, refer to the [geohb\(8\)](#) man page.

**Example 14** Displaying Heartbeat Configuration Information

This example displays information about the `paris-to-newyork` heartbeat.

```
# geohb list paris-to-newyork
```

## Tuning the Heartbeat Properties

Default heartbeats are created as part of partnership creation. If you use a custom heartbeat, the custom heartbeat should be created before you create a partnership. You can modify the

properties of the default and custom heartbeats by using the `geohb set-prop` command. For more information about this command, refer to the [geohb\(8\)](#) man page.

---

**Note** - Custom heartbeats are provided for special circumstances and require careful configuration. Consult your Oracle specialist for assistance if your system requires the use of custom heartbeats.

---

If you modify the default value of the `Query_interval` property, ensure that the interval is sufficiently long. An interval that is too short causes a timeout and heartbeat-loss event before the logical hostname resource is available. This failover should result in no more than two unanswered heartbeat requests. Setting a default `Query_interval` value of 120 seconds with the default `heartbeat.retries` parameter of 3 enables the peer cluster to be unresponsive for 6 minutes ( $120 * 3$ ) without having a false failure declared.

The `heartbeat.retries` parameter is specified in the `com.sun.cluster.agent.geocontrol.xml` file.

If you adjust the delay setting of the `Query_interval` property, ensure that the following condition is met:

$$\text{Query\_interval} > \text{worst-case-logical-host-failover-time} / 2$$

You must empirically determine the logical-host failover time for the cluster in question.

The following must be true to avoid false failures:

$$\text{Query\_interval} > \text{worst-case-logical-host-failover-time} / 3$$

You should not change the `heartbeat.retries` value. If you want to change the default value of the `heartbeat.retries` property, contact an Oracle service representative.

## ▼ How to Modify the Heartbeat Properties

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, click the heartbeat name to go to its page, click the Properties tab, and click Edit to access the listed properties for editing. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**1. Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Modify the heartbeat properties.**

```
# geohb set-prop -p property-setting [-p...] heartbeat-name
```

*-p property-setting*

Specifies the default properties of the heartbeat.

A heartbeat property is assigned a value by a *name=statement* pair. Multiple properties can be set at one time by using multiple statements.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*heartbeat-name*

Specifies an identifier for the heartbeat settings.

For information about the names and values that are supported by the disaster recovery framework, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

For more information about the geohb command, refer to the [geohb\(8\)](#) man page.

**Example 15** Modifying the Properties of the Default Heartbeat

This example modifies the settings for the default heartbeat between `cluster-paris` and `cluster-newyork`.

```
# geohb set-prop -p Query_interval=60 hb_cluster-paris~cluster-newyork
```

## Creating a Heartbeat That Uses a Custom Heartbeat Plug-in

You can create a custom heartbeat plug-in and configure an existing default heartbeat or a new custom heartbeat to use this custom heartbeat plug-in.

Custom heartbeats are provided for special circumstances and require careful configuration. Consult your Oracle specialist for assistance if your system requires the use of custom heartbeats.



---

**Note** - If you configure a custom heartbeat, ensure that the name of your custom heartbeat is different from the name of the custom heartbeat on the partner cluster.

---



---

**Caution** - The presence of a custom heartbeat prevents the default heartbeat from being used during partnership creation. If you want to use the default heartbeat for your partnership, you must delete the custom heartbeat before running the `geops create` command.

---

This section contains the following information:

- [“Creating a Custom Heartbeat Plug-in” on page 65](#)
- [“How to Add a Custom Heartbeat Plug-in to an Existing Default Heartbeat” on page 66](#)
- [“How to Create a Custom Heartbeat Plug-in and Add It to a Custom Heartbeat” on page 67](#)

## Creating a Custom Heartbeat Plug-in

When a heartbeat is created, your custom heartbeat plug-in is passed the following arguments by the disaster recovery framework:

### *query-interval*

The value of the `Query-interval` property, which defines the delay in seconds after which a heartbeat status request is declared a failure.

### *mode*

The mode for the plug-in startup, either `Normal` or `Emergency`.

### *plug-in-property-values*

The value of the `Plugin-properties` property that is configured for the heartbeat plug-in, if any.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

Your custom heartbeat plug-in is expected to check the heartbeat on the secondary cluster and return one of the following exit values:

- Zero, if successful – Indicates that the secondary cluster is alive
- Nonzero, on failure – Indicates that the secondary cluster did not respond to the heartbeat check

## ▼ How to Add a Custom Heartbeat Plug-in to an Existing Default Heartbeat

### 1. Log in to a node in the primary cluster.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### 2. Add the custom heartbeat plug-in to the default heartbeat.

```
# geohb add-plugin -p property-setting [-p...] \  
plug-in-name hb_local-cluster-name-remote-partner-cluster-name
```

*-p property-setting*

Specifies the properties of the heartbeat plug-in by using a *name=statement* pair.

Specify the path to your custom heartbeat plug-in by using the *Query\_cmd* property.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*plug-in-name*

Specifies the name of the custom heartbeat plug-in.

*hb\_local-cluster-name-remote-partner-cluster-name*

Specifies the name of the default heartbeat to which you want to add the custom heartbeat plug-in.

### 3. Verify that your changes were made correctly.

```
# geoadm status
```

### 4. Repeat the previous steps on a node of the secondary cluster.

#### Example 16 Adding a Custom Heartbeat Plug-in to the Default Heartbeat

This example adds the custom heartbeat plug-in, *command1*, to the default heartbeat, *hb\_cluster-paris-cluster-newyork*.

```
# geohb add-plugin -p query_cmd=/usr/bin/hb command1 hb_cluster-paris-cluster-newyork  
# geoadm status
```

## ▼ How to Create a Custom Heartbeat Plug-in and Add It to a Custom Heartbeat

### 1. Log in to a cluster node.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### 2. Create the new custom heartbeat.

```
# geohb create -r remote-partner-cluster-name [-p property-setting [-p...]] heartbeat-name
```

```
-r remote-partner-cluster-name
```

Specifies the name of the remote, secondary partner cluster.

```
-p property-setting
```

Specifies the default properties of the heartbeat.

A heartbeat property is assigned a value by a *name=statement* pair.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

```
heartbeat-name
```

Specifies an identifier for the heartbeat settings.




---

**Caution** - The name of the custom heartbeat on each cluster in the same partnership must be different. Choose a name that uniquely identifies the heartbeat, such as *paris-to-newyork* on the cluster *cluster-paris* and *newyork-to-paris* on cluster *cluster-newyork*.

---

For more information about the `geohb` command, refer to the [geohb\(8\)](#) man page.

### 3. Add the custom heartbeat plug-in to the heartbeat.

```
# geohb add-plugin -p property-setting [-p...] plug-in-name heartbeat-name
```

```
-p property-setting
```

Specifies the properties of the heartbeat plug-in by using a *name=statement* pair.

Specify the path to your custom heartbeat plug-in by using the `Query_cmd` property.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*plug-in-name*

Specifies the name of the custom heartbeat plug-in.

*heartbeat-name*

Specifies an identifier for the heartbeat.

**4. Create the partnership that will use the heartbeat that you created in the previous step.**

```
# geops create -c remote-partner-cluster-name -h heartbeat-name \  
[-p property-setting [-p...]] partnership-name
```

*-c remote-partner-cluster-name*

Specifies the name of remote cluster that will participate in the partnership.

This name matches the logical hostname used by the disaster recovery framework on the remote cluster.

*-h heartbeat-name*

Specifies the custom heartbeat to be used in the partnership to monitor the availability of the partner cluster.

*-p property-setting*

Sets the value of partnership properties with a string of *name=value* pair statements.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*partnership-name*

Specifies the name of the partnership.

For more information about using `geops create` command to create a partnership, see [“How to Create a Partnership” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

**5. Verify that your changes were made correctly.**

```
# geoadm status
```

**Example 17** Adding a Custom Heartbeat Plug-in to a New Custom Heartbeat

This example creates the heartbeat `paris-to-newyork`, which uses a custom heartbeat plug-in, and associates the heartbeat with a new partnership.

```
# geohb create -r cluster-newyork paris-to-newyork
# geohb add-plugin -p query_cmd=/usr/bin/hb/ command1 paris-to-newyork
# geops create -c cluster-newyork -h paris-to-newyork paris-newyork-ps
# geoadm status
```

## Configuring Heartbeat-Loss Notification

This section contains the following information:

- [“Introduction to Configuring Heartbeat-Loss Notification” on page 69](#)
- [“Configuring the Heartbeat-Loss Notification Properties” on page 70](#)
- [“Creating an Action Shell Script for Heartbeat-Loss” on page 71](#)

### Introduction to Configuring Heartbeat-Loss Notification

You can configure the disaster recovery framework to send email notification and to run an action script when a heartbeat is lost. You configure heartbeat-loss notification by using the optional `Notification_emailaddr`s and `Notification_actioncmd` properties.

Heartbeat-loss notification occurs if the heartbeat still fails after the interval you configure with the `Query_interval` property of the heartbeat. The heartbeat monitor sends out a heartbeat request to the responder on the logical host every `Query_interval` period. If no response is received within the `Query_interval`, an internal count is incremented. If the recount reaches the number that is specified in the `heartbeat.retries` property, the heartbeat is deemed to have failed.

For example, you can use the default `Query_interval` of 120 seconds and the default `heartbeat.retries` of 3. The heartbeat-lost event will be sent a maximum of 10 minutes after the last heartbeat response from the partner cluster.

```
120sec (delay since last query) + 3*120sec (wait for normal response)
+ 120 sec (wait for retry response)
```

Delays can occur between the generation of the heartbeat-loss event and the triggering of the heartbeat-loss notification.

---

**Note** - A heartbeat-loss event does not necessarily indicate that the remote cluster has crashed.

---

The following sections describe how to configure the heartbeat-loss notification properties and how to create a custom action script that the disaster recovery framework runs after a heartbeat-loss event.

## Configuring the Heartbeat-Loss Notification Properties

You can configure heartbeat-loss notification by using two partnership properties, `Notification_emailaddr`s and `Notification_actioncmd`. You specify these properties by using the `geops` command.

You can specify these properties on the default heartbeat during partnership creation. For more information, see [“How to Create a Partnership” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#). You can also modify these properties by using the procedure that is described in [“How to Modify the Heartbeat Properties” on page 63](#).

If you want to be notified of heartbeat loss by email, set the `Notification_emailaddr`s property. You can specify a list of email addresses, separated by commas. If you want to use email notification, the cluster nodes must be configured as email clients. For more information about configuring mail services, see [Chapter 5, “Administering Mail Services” in \*Managing sendmail Services in Oracle Solaris 11.4\*](#).

If you want to run a command in response to heartbeat loss, set the `Notification_actioncmd` property.

### **EXAMPLE 18** Configuring Heartbeat-Loss Notification for an Existing Partnership

This example specifies a notification email address and a custom notification script for the partnership, `paris-newyork-ps`.

```
phys-paris-1# geops set-prop \  
-p Notification_emailaddr=ops@paris.example.com,ops@newyork.example.com \  
-p Notification_actioncmd=/opt/hb_action.sh paris-newyork-ps
```

## Creating an Action Shell Script for Heartbeat-Loss

You can create an action shell script that runs when the local cluster detects a heartbeat-loss in the partner cluster. The script runs with root permissions. The file must have root ownership and execution permissions, but the script should not have write permissions.

If you have configured the `Notification_actioncmd` property, the action command runs with arguments that provide information about the event in the following command line:

```
# custom-action-command-path -c local-cluster-name -r remote-partner-cluster-name -e 1 \  
-n nodename -t time
```

*custom-action-command-path*

Specifies a path to the action command you have created.

*-c local-cluster-name*

Specifies the name of the local cluster.

*-p remote-partner-cluster-name*

Specifies the name of the remote partner cluster.

*-e1*

Specifies that `HBL0ST=1`, which indicates that a heartbeat-loss event has occurred. The disaster recovery framework only supports heartbeat-loss notification, so `-e 1` is the only value that can be passed to the action shell script.

*-nnodename*

Specifies the name of the cluster node that sent the heartbeat-loss event notification.

*-t timestamp*

Specifies the time of the heartbeat-loss event as the number of milliseconds since January 1, 1970, 00:00:00 GMT.



---

**Caution** - You can use this script to perform an automatic takeover on the secondary cluster. However, such an automated action is risky. If the heartbeat-loss notification is caused by a total loss of all heartbeat connectivity on both the primary and secondary clusters, such an automated action could lead to a situation where two primary clusters exist.

---

**EXAMPLE 19** How a Notification Action Script Parses the Command-Line Information Provided by the Disaster Recovery Framework

This example displays the event information that is provided in the command-line being parsed in a notification action shell script.

```
#!/bin/sh

set -- `getopt abo: $*`
if [ $? != 0 ]
then
echo $USAGE
exit 2

fi
for i in $*
do

case $i in
-p)    PARTNER_CLUSTER=$1; shift;;
-e)    HB_EVENT=$2; shift;;
-c)    LOCAL_CLUSTER=$3; shift;;
-n)    EVENT_NODE=$4; shift;;
esac
done
```



## Administering Protection Groups

---

This chapter contains the following information:

- [“Activating and Deactivating a Protection Group” on page 73](#)
- [“Resynchronizing a Protection Group” on page 80](#)
- [“Modifying Protection Groups and Data Replication Components” on page 81](#)
- [“Deleting Protection Groups and Data Replication Components” on page 83](#)

### Activating and Deactivating a Protection Group

This section provides the following information:

- [“Guidelines for Activating and Deactivating a Protection Group” on page 73](#)
- [“How to Activate a Protection Group” on page 74](#)
- [“How to Deactivate a Protection Group” on page 77](#)

### Guidelines for Activating and Deactivating a Protection Group

When you activate a protection group, the protection group assumes the role that you assigned to it during configuration. When you deactivate a protection group, its application resource groups are also unmanaged.

You can activate or deactivate a protection group in the following ways:

- Globally – Activates or deactivates a protection group on both clusters where the protection group is configured.
- On the primary cluster only – Secondary cluster remains inactive.

When a protection group is activated on the primary cluster, the application resource groups that are configured for the protection group are also started. The disaster recovery framework uses the following Oracle Solaris Cluster commands on the primary cluster to bring the resource groups online:

```
# clresourcegroup online -eM resource-group-list
```

- On the secondary cluster only – Primary cluster remains inactive.

When you activate a protection group, the data replication product that you are using determines the clusters on which data replication can start. The following sections describe additional behaviors when you activate or deactivate a particular data replication product:

- [“Effects of Activating an Oracle Data Guard Protection Group” on page 74](#)

## Effects of Activating an Oracle Data Guard Protection Group

The Oracle Data Guard software allows data replication to start only if you activate a protection group in one of the following ways:

- Locally from the primary cluster.
- Globally from either the primary or the standby cluster.

So, if you attempt to activate an Oracle Data Guard protection group locally from the standby cluster, data replication does not start. However, if you activate the protection group globally from the standby cluster, data replication does start.

## ▼ How to Activate a Protection Group

This procedure activates, or starts, the protection group on the primary and secondary clusters, depending on the scope of the command. When you activate a protection group on the primary cluster, its application resource groups are also brought online.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, highlight the protection group name, and click Start Protection Group. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **Assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see “Disaster Recovery Framework Rights Profiles” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

## 2. Activate the protection group.

When you activate a protection group on the primary cluster, its application resource groups are also brought online.

```
phys-node-n# geopg start -e scope [-n] protection-group-name
```

`-e scope`

Specifies the scope of the command.

If the scope is `local`, then the command operates on the local cluster only. If the scope is `global`, the command operates on both clusters that deploy the protection group.

---

**Note** - The property values `global` and `local` are *not* case sensitive.

---

`-n`

Prevents the start of replication at protection group startup.

If you omit this option, the replication subsystem starts at the same time as the protection group. The following data replication products have additional behaviors:

- For Oracle Data Guard, if you omit this option, the `geopg start` command performs the following operations on each Oracle Data Guard Broker configuration in the protection group:
  - Verifies that the resource group that is named in the `local_oracle_svr_rg_name` property contains a resource of type `SUNW.scalable_rac_server_proxy` for a scalable resource group or a resource of type `SUNW.oracle_server` for a failover resource group.
  - Verifies that the Oracle Data Guard `dgmgrl` command can connect using the values that are given for `sysdba_username`, `sysdba_password`, and `local_db_service_name`. Or if the `sysdba_username` and `sysdba_password`

properties are null, verifies that the `dgmgrl` command can connect using the Oracle wallet connection format, `dgmgrl /@local_db_service_name`.

- Verifies that the role configured for the replication resource is the same as the role of the protection group on the local cluster.
- Verifies that the Oracle Data Guard Broker configuration details match those that are held by the disaster recovery framework. The details to check include which cluster is primary, the configuration name, the database mode (for both the primary and standby clusters), the replication mode, the standby type, that `FAST_START FAILOVER` is disabled, and that `BystandersFollowRoleChange` is equal to `NONE`.
- For MySQL, if you omit this option, the `geopg start` command performs the following actions, if the role of the protection group is secondary on the local cluster:
  - Starts the MySQL slave threads
  - Prevents modification by an unprivileged database user if this option is configured
  - Prepares the `my.cnf` file to start the database with modifications prevented for an unprivileged database user if this option is configured

*protection-group-name*

Specifies the name of the protection group.

The `geopg start` command uses the `clresourcegroup online -eM resourcegrouplist` command to bring resource groups and resources online. For more information about using this command, see the [clresourcegroup\(8CL\)](#) man page.

If the role of the protection group is primary on the local cluster, the `geopg start` command brings the application resource groups in the protection group online on the local cluster. For Oracle Data Guard, this includes the shadow Oracle database server resource groups.

The following data replication products have additional behaviors:

- For MySQL, the `geopg start` command also performs the following operations:
  - Prepares the `my.cnf` file to start the database without the slave threads
  - Brings online the application resource groups in the protection group on the local cluster

If the command fails, the `Configuration` status might be set to `Error`, depending on the cause of the failure. The protection group remains deactivated, but data replication might be started and some resource groups might be brought online. Run the `geoadm status` command or Oracle Solaris Cluster Manager to obtain the status of your system.

If the `Configuration` status is set to `Error`, revalidate the protection group by using the procedures that are described in [“How to Validate a Protection Group” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

**Example 20** Globally Activating a Protection Group

This example globally activates a protection group.

```
phys-paris-1# geopg start -e global sales-pg
```

**Example 21** Locally Activating a Protection Group

This example activates a protection group on a local cluster only. This local cluster might be a primary cluster or a standby cluster, depending on the role of the cluster.

```
phys-paris-1# geopg start -e local sales-pg
```

## ▼ How to Deactivate a Protection Group

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, highlight the protection group name, and click Stop Protection Group. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

You can deactivate, or stop, a protection group in the following ways:

- Globally, meaning you deactivate a protection group on both the primary and the standby cluster where the protection group is configured
- On the primary cluster only
- On the standby cluster only

This procedure deactivates the protection group on all nodes of the primary and secondary clusters, depending on the scope of the command. When you deactivate a protection group, its application resource groups are also unmanaged.

The result of deactivating a protection group on the primary or standby cluster depends on the type of data replication that you are using.

- If you are using Oracle Data Guard software, you can stop the Oracle Data Guard configuration from the primary or the standby cluster when the configuration is enabled because the Oracle Data Guard command-line interface (`dgmgrl`) on both clusters still accepts commands.

1. **Assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

## 2. Deactivate the protection group.

When you deactivate a protection group on the primary cluster, its application resource groups are also taken offline.

```
# geopg stop -e scope [-D] protection-group-name
```

`-e scope`

Specifies the scope of the command.

If the scope is `Local`, then the command operates on the local cluster only. If the scope is `Global`, the command operates on both clusters where the protection group is deployed.

---

**Note** - The property values, such as `global` and `local`, are *not* case sensitive.

---

`-D`

Specifies that only replication should be stopped and the protection group should be online.

If you omit this option, the replication subsystem and the protection group are both stopped. If the role of the protection group on the local cluster is primary, omitting the `-D` option also results in taking the application resource groups offline and putting them in an unmanaged state.

`protection-group-name`

Specifies the name of the protection group.

The following data replication products have additional behaviors:

- For Oracle Data Guard, if the role of the protection group is primary on the local cluster, the `geopg stop` command disables the Oracle Data Guard Broker configuration.
  - If the `geopg stop` command fails, run the `geoadm status` command or Oracle Solaris Cluster Manager to see the status of each data replication component. For example, the

Configuration status might be set to Error depending on the cause of the failure. The protection group might remain activated even though some resource groups might be unmanaged. The protection group might be deactivated with data replication running.

- If the Configuration status is set to Error, revalidate the protection group by using the procedures described in [“How to Validate a Protection Group”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**Example 22** Deactivating a Protection Group on All Clusters

This example deactivates a protection group on all clusters.

```
# geopg stop -e global sales-pg
```

**Example 23** Deactivating a Protection Group on the Local Cluster

This example deactivates a protection group on the local cluster.

```
# geopg stop -e local sales-pg
```

**Example 24** Stopping Remote Replication While Leaving the Protection Group Online

This example stops replication on the local cluster only.

```
# geopg stop -e local -D sales-pg
```

If you decide later to deactivate both the protection group and its underlying replication subsystem, you can rerun the command without the -D option:

```
# geopg stop -e local sales-pg
```

**Example 25** Deactivating a Protection Group While Keeping Application Resource Groups Online

This example keeps online two application resource groups, apprg1 and apprg2, while deactivating their protection group, sales-pg, on both clusters.

1. Remove the application resource groups from the protection group.

```
# geopg remove-resource-group apprg1,apprg2 sales-pg
```

2. Deactivate the protection group.

```
# geopg stop -e global sales-pg
```

**Troubleshooting** If the `geopg stop` command fails, run the `geoadm status` command or Oracle Solaris Cluster Manager to obtain the status of each data replication component. For example, the configuration status might be set to Error depending on the cause of the failure. The protection group might remain activated even though some resource groups might be unmanaged. The protection group might be deactivated with replication running.

If the configuration status is set to Error, revalidate the protection group. See [“Validating a Protection Group” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

## Resynchronizing a Protection Group

You can resynchronize the configuration information of the local protection group with the configuration information retrieved from the partner cluster. You need to resynchronize a protection group when its Synchronization status in the output of the `geoadm status` command or in Oracle Solaris Cluster Manager is Error. For example, you might need to resynchronize protection groups after booting the cluster. For more information, see [“Booting a Cluster” in \*Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

Resynchronizing a protection group updates only entities that are related to the disaster recovery framework. To update Oracle Solaris Cluster resource groups, resource types, and resources, use the `cluster export -t rg,rt,rs` command to generate an XML cluster configuration file, modify the XML file for the expected configuration on the secondary cluster, and run the `clresource create` command with the `-a` option to apply the configuration updates. For more information, see [“How to Configure Oracle Solaris Cluster Software on All Nodes \(XML\)” in \*Installing and Configuring an Oracle Solaris Cluster 4.4 Environment\*](#) and the `cluster(8CL)` and `clresource(8CL)` man pages.

### ▼ How to Resynchronize a Protection Group

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, highlight the protection group name, and click Update Protection Group. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**Before You Begin** The protection group must be deactivated on the cluster where you run the `geopg update` command.



1. **Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

2. **Resynchronize the protection group.**

```
# geopg update protection-group-name
```

*protection-group-name*

Specifies the name of the protection group

## Modifying Protection Groups and Data Replication Components

This section contains the following procedures:

- [“How to Modify a Protection Group” on page 81](#)
- [“How to Modify a Data Replication Component” on page 82](#)

### ▼ How to Modify a Protection Group

Perform this procedure to modify the properties of a protection group.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, click the protection group name to go to its page, click the Properties tab, then click Edit to access the list of editable properties. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**Before You Begin** Ensure that the protection group you want to modify exists locally.

1. **Log in to one of the cluster nodes.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

## 2. **Modify the configuration of the protection group.**

This command modifies the properties of a protection group on all nodes of the local cluster. If the partner cluster contains a protection group of the same name, this command also propagates the new configuration information to the partner cluster.

```
# geopg set-prop -p property-setting [-p . . .] protection-group
```

*-p property-setting*

Specifies the properties of the protection group.

For more information about the protection group properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*protection-group*

Specifies the name of the protection group.

For information about the names and values that are supported by the disaster recovery framework, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

For more information about the `geopg` command, see the [geopg\(8\)](#) man page.

## ▼ **How to Modify a Data Replication Component**

Use this procedure to modify a property of a data replication component.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, click the protection group name to go to its page, click the data replication component name to go to its page, click the Properties tab, and click Edit to access the list of editable properties. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

### 1. **Assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and Oracle ZFS Storage Appliance software.

---

## 2. Modify the data replication component.

The following command modifies the properties of a data replication component in a protection group on the local cluster. The command then propagates the new configuration to the partner cluster if the partner cluster contains a protection group with the same name.

```
# geopg modify-replication-component -p property [-p...] replication-component protection-group
```

*-p property*

Specifies the properties of the data replication component.

*replication-component*

Specifies the name of the data replication component.

*protection-group*

Specifies the name of the protection group that contains the data replication component.

### Example 26 Modifying the Properties of an Oracle ZFS Storage Appliance Remote Replication Component

The following example modifies the `Timeout` property of the remote replication component `trancos` which is part of the appliance protection group `zfssa-p`.

```
# geopg modify-replication-component -p Timeout=215 trancos zfssa-pg
```

## Deleting Protection Groups and Data Replication Components

This section provides the following procedures:

- [“How to Delete a Protection Group” on page 84](#)

- [“How to Remove a Data Replication Component From a Protection Group”](#) on page 85

## ▼ How to Delete a Protection Group

Use the following procedure to delete a protection group from the local cluster.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, highlight the protection group name, and click Delete. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

If you want to delete the protection group everywhere, you must run the `geopg delete` command on each cluster where the protection group exists.

To keep the application resource groups online while deleting the protection group, you must remove the application resource groups from the protection group.

**Before You Begin** Ensure that the following conditions are met:

- The protection group you want to delete exists locally.
- The protection group is offline on all clusters from which you want to delete it.

### 1. Log in to one of the nodes on the primary cluster, `cluster-paris`.

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

### 2. Delete the protection group.

the following command deletes the configuration of the protection group from the local cluster. The command also removes the replication resource group for each device group in the protection group.

```
# geopg delete protection-group-name
```

*protection-group-name*

Specifies the name of the protection group

### 3. To also delete the protection group on the secondary cluster, repeat Step 1 and Step 2 from a node of the secondary cluster.

**Example 27** Deleting a Protection Group

This example deletes a protection group `zfszapg` from both partner clusters. The protection group is offline on both partner clusters.

In this example, `phys-paris-1` is a node of the primary cluster and `phys-newyork-1` is a node of the secondary cluster.

```
# rlogin phys-paris-1 -l root
phys-paris-1# geopg delete zfszapg
# rlogin phys-newyork-1 -l root
phys-newyork-1# geopg delete zfszapg
```

**Example 28** Deleting a Protection Group While Keeping Application Resource Groups Online

This example keeps online two application resource groups, `apprg1` and `apprg2`, while deleting their protection group, `mypg`, from both partner clusters. The following commands remove the application resource groups from the protection group, then delete the protection group.

```
phys-paris-1# geopg remove-resource-group apprg1,apprg2 mypg
phys-paris-1# geopg stop -e global mypg
phys-paris-1# geopg delete mypg
phys-newyork-1# geopg delete mypg
```

**Troubleshooting** If the deletion is unsuccessful, the configuration status is set to Error. Fix the cause of the error and rerun the `geopg delete` command.

For Oracle Data Guard, unlike other data replication modules, the Oracle database-server resource group is not added to the protection group. Instead, a shadow Oracle database-server resource group is added to represent this resource group. You can add and remove the shadow Oracle database-server resource group to and from the protection group at any time without affecting the Oracle Data Guard data replication. Consequently, the application resource groups that are shown in this example would have no data to replicate in an Oracle Data Guard protection group. Application resource groups that might meet this criteria can be scalable web servers, where their data is static or held on some remote storage that is not controlled by this cluster.

## ▼ How to Remove a Data Replication Component From a Protection Group

Use the following procedure to remove a data replication component from a protection group.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, click the protection group name to go to its page, highlight the data replication component name, and click Remove. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**Before You Begin** Ensure that the following conditions are met:

- The protection group is defined on the local cluster.
- The protection group is offline on the local cluster and the partner cluster, if the partner cluster can be reached.
- The data replication component is managed by the protection group.

**1. Assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and Oracle ZFS Storage Appliance software.

---

**2. Remove the data replication component.**

```
# geopg remove-replication-component replication-component protection-group
```

*replication-component*

Specifies the name of the data replication component.

*protection-group*

Specifies the name of the protection group.

**Example 29** Removing a Remote Replication Component From an Oracle ZFS Storage Appliance Protection Group

In the following example, the remote replication component `trancos` is removed from the appliance protection group, `zfssa-pg`.

```
# georg remove-replication-component trancos zfssa-pg
```





## Administering Sites

---

You administer a site from a cluster that is a controller of the site. Changes to the site are propagated by the issuing controller cluster to all other clusters in the site.

This chapter contains the following information about administering a site:

- [“Adding a Cluster to a Site” on page 89](#)
- [“Changing the Site Role of a Site Cluster” on page 93](#)
- [“Synchronizing Site Configuration Information” on page 97](#)
- [“Removing a Cluster From a Site” on page 98](#)
- [“Deleting a Site” on page 101](#)

### Adding a Cluster to a Site

This section contains the following procedures to add a cluster to a site.

- [“How to Add a Cluster to a Site” on page 89](#)
- [“How to Accept an Invitation to Join a Site” on page 91](#)

### ▼ How to Add a Cluster to a Site

Perform this command to add one or more clusters to an existing site. You can add a global cluster or a zone cluster. For procedures to create a site, see [“How to Create a Site” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, and in the Clusters section click Add. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **From a node of a site controller cluster, assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the `root` role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Ensure that all nodes of the site controller cluster are online.**

```
phys-schost-1# cluster status -t node
=== Cluster Nodes ===
```

```
--- Node Status ---
```

Node Name	Status
-----	-----
phys-schost-2	Online
phys-schost-1	Online

If any node is offline, wait until the node is brought back up before you issue the site invitation. The addition of a new cluster to a site will fail if any node in the issuing cluster is not online.

3. **Invite the cluster to join the site.**

- **To add the cluster as a site controller, use the following command.**

```
site-controller# geosite add-member -c cluster site
```

- **To add the cluster as a member, use the following command.**

```
site-controller# geosite add-member -m cluster site
```

```
-c
    Specifies the site controller role.
```

```
-m
    Specifies the site member role.
```

*cluster*

The name of the cluster to add to the site. To add multiple clusters, separate the cluster names with a comma (.). You can use both the `-c` and `-m` options in the same `geosite add-member` command.

*site*

The name of the site that you are adding the cluster to.

#### 4. Verify the invitation.

Command output is similar to the following, where the new cluster is added as a site controller.

```
site-controller## geosite status

Site : site

Controller "issuing-cluster"
Configuration           : OK

Controller "added-cluster"
Configuration           : OK
Synchronization        : Unknown
...
```

**Troubleshooting** If the `geosite add-member` command fails with a timeout, update the site's Timeout value to a larger number and retry the command.

**Next Steps** To complete the addition of the cluster to the site, the invited cluster must join the site. Go to [“How to Accept an Invitation to Join a Site” on page 91](#).

## ▼ How to Accept an Invitation to Join a Site

Perform this procedure to complete the addition of a cluster to a site.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, highlight the site name, and click Join. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**Before You Begin**

- Ensure that an invitation to join a site has been issued for the cluster. See [“How to Add a Cluster to a Site” on page 89](#).

- Ensure that all nodes of the cluster that issued the invitation are online.
- Ensure that the common agent container is started on all nodes of both the cluster that issued the invitation and the invited cluster.

**1. From a node of the site cluster that issued the invitation, ensure that all nodes are online.**

```
phys-schost-1# clnode status
=== Cluster Nodes ===
```

```
--- Node Status ---
```

Node Name	Status
phys-schost-2	Online
phys-schost-1	Online

If any node of the issuing cluster is offline, wait until the node is back online before you accept the site invitation. The acceptance of a site invitation will fail if any node in the issuing cluster is not online.

**2. From a node of the invited cluster, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

**3. Ensure that all nodes of the invited cluster are online.**

```
invited-cluster-node# cluster status -t node
```

**4. Accept the invitation to join the site.**

```
invited-cluster-node# geosite join issuing-cluster site
```

```
issuing-cluster
```

The name of the cluster that issued the `geosite` command to add the invited cluster.

*site*

The name of the site that the invited cluster is joining.

##### 5. Verify that the cluster is a member of the site.

The following command lists all sites that the issuing cluster is a member of.

```
invited-cluster-node# geosite list
site
```

**Troubleshooting** If a `geosite join` operation fails due to a failure to verify trust, ensure that the common agent container is running on all nodes of both the invited cluster and the issuing cluster. If a node is not running the common agent container, issue the `/usr/sbin/cacaoadm start` command on that node.

If a `geosite join` operation fails due to a timeout, re-issue the command with a `joinTimeout` property value larger than the default of 30. For example, `geosite join cluster -p joinTimeout=120 site`.

## Changing the Site Role of a Site Cluster

A cluster that is part of a site can have one of two roles, controller or member:

- A cluster that has the controller role for a site is authorized to perform operations on the other site members and the multigroups they manage. The controller authorization is based on mutual agreement, any cluster can propose itself as a controller, but it can only control those clusters which have then agreed to accept its authority.
- A cluster that is a simple site member has agreed to accept commands from the site controllers, but cannot issue commands with site-wide effect.

This section provides the following procedures to change the role of a site cluster:

- [“How to Change a Site Membership Role” on page 93](#)
- [“How to Forcibly Change a Site Member to a Site Controller” on page 95](#)

### ▼ How to Change a Site Membership Role

Perform this procedure to change the role of a site cluster between member and controller.

---

**Note** - If the site has no reachable site controller, instead perform [“How to Forcibly Change a Site Member to a Site Controller”](#) on page 95.

---

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the cluster name, and click Set Role. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

1. **From a node of a site controller cluster, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information about, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Change the site role of the cluster.**

- **To change a site member to a site controller, use the following command.**

```
site-controller# geosite set-role -c cluster site
```

- **To change a site controller to a site member, use the following command.**

```
site-controller# geosite set-role -m cluster site
```

-c

Specifies the site controller role.

-m

Specifies the site member role.

*cluster*

The name of the cluster to change roles. You can specify more than one cluster from the same site in a single `geosite set -role` command by using the `-c` or `-m` option for each cluster that you want to change roles.

*site*

The name of the site with which the cluster has membership.

## ▼ How to Forcibly Change a Site Member to a Site Controller

Perform this procedure to change the role of a site member to a site controller when no existing site controller is accessible. There must be at least one other accessible member in the site.

The site member submits itself or another site member as a candidate for site controller. Each site member cluster must individually accept the candidate cluster as a new controller. No controller operations that are issued by the candidate cluster are obeyed by a site member cluster that has not yet accepted the candidate cluster as a controller.

If a site controller cluster was unavailable at the time a site member cluster submits itself to be a controller, the controller automatically accepts and synchronizes the site configuration change when it again becomes available.

If the site member is the only cluster in the site, the controller role is immediately assigned to the cluster. If later another cluster is added to the site, the new cluster must then accept the existing cluster's state as a site controller.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the cluster name, click Set Role, and set the force option. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

1. **From a node of the site member cluster that you want to make a site controller, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

**2. Submit the request to become a site controller.**

```
candidate-cluster-node# geosite set-role -c candidate-cluster site
```

```
-c candidate-cluster
```

The name of the cluster that is requesting to be made a site controller. The candidate cluster can be the issuing cluster or another site member cluster.

```
site
```

The name of the site for which the candidate cluster is requesting the site controller role.

**3. From another member cluster in the site, accept the request from the candidate cluster.**

A site member cluster in the site must accept the role change request before the change will take effect for that cluster.

---

**Note** - If the site has no other clusters, omit the remaining steps in this procedure.

---

```
site-member-cluster-node# geosite accept candidate-cluster site
```

```
candidate-cluster
```

The name of the cluster that requested to be made a controller of the specified site.

**4. Verify on the accepting cluster that the candidate cluster is now a site controller.**

```
site member-cluster-node# geosite show -v site
```

```
...
```

```
candidate-cluster controller
```

**5. Repeat [Step 3](#) and [Step 4](#) for each remaining cluster in the site.**

**Troubleshooting** If more than one site member cluster proposes itself as a site controller and different site clusters accept a different new site controller, this creates a conflict for control of the site. To



avoid such conflicts, promote only one cluster as a site controller and accept it on all accessible site clusters before you promote another cluster as a new site controller.

If a conflict for control of the site does occur, identify which cluster should be the site controller, then issue the `geosite update desired-site-controller site` command from each accessible site cluster that is in a synchronization error with that cluster.

If any cluster is down at the time that a site cluster is promoted to site controller, after the site cluster returns to service, ensure that it has no synchronization conflicts and, if necessary, use the `geosite update` command to resolve any conflicts.

## Synchronizing Site Configuration Information

Under normal circumstances, a membership change or a change to a site property is automatically propagated to all clusters in the site.

This section provides procedures to manually update site configuration information on the local cluster with configuration information from the remote site cluster. The local site configuration information is overwritten by information from the remote cluster.

### ▼ How to Synchronize Site Configuration Information

Perform this procedure to update site configuration information on the local cluster.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, highlight the site name, and click Validate Site. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **From a node of a site controller cluster, assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

## 2. Synchronize the site configuration information with the remote site cluster.

Site configuration information from the remote site cluster overwrites the site configuration information on the local cluster.

```
local-site-cluster-node# geosite update remote-site-cluster site
```

```
remote-site-cluster
```

The name of the remote site cluster whose site configuration information to import to the local site cluster. Both clusters must be configured in the same site.

```
site
```

The name of the site.

**See Also** For details about the synchronization statuses, see the Synchronization Status section of the [geoadm\(8\)](#) man page.

## Removing a Cluster From a Site

This section contains the following procedures to remove a cluster from a site:

- [“How to Remove a Cluster From a Site” on page 98](#)
- [“How to Remove an Unreachable Cluster From a Site a Cluster” on page 100](#)

### ▼ How to Remove a Cluster From a Site

Perform this procedure to make a cluster remove itself from a site. If the cluster you want to remove from a site is unreachable, instead follow [“How to Remove an Unreachable Cluster From a Site a Cluster” on page 100](#).

---

**Note** - To remove the last cluster from a site, instead follow [“How to Delete a Site” on page 101](#).

---

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task.

- To remove the local cluster, click Sites, highlight the site name, and click Leave Site.
- To remove another cluster, click Sites, click the site name to go to its page, highlight the cluster name, and click Remove,

For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **From a node of the cluster that you want to remove from a site, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Detach the issuing cluster from the site.**

- **If the cluster is not the last controller in the site, use the following command.**

```
node-of-cluster-to-remove# geosite leave site
```

*site*

The name of the site to remove the cluster from.

- **If the cluster is the last controller in the site, include the `-f` option in the command.**

```
node-of-cluster-to-remove# geosite leave -f site
```

- If the site contains other member clusters, the site continues to exist without a site controller.
- If the site has no other member clusters, removal of the last controller also deletes the site.

---

**Note** - If the last site member being removed is not a site controller, the remove command fails with an error.

---

3. **From a remaining site cluster, verify that the removed cluster is no longer listed as a site member.**

```
# geosite show -v site

*** Site "site" is not configured ***
```

## ▼ How to Remove an Unreachable Cluster From a Site a Cluster

Perform this procedure to remove from a site a cluster that is unreachable. If a cluster you want to remove is reachable, instead follow [“How to Remove a Cluster From a Site” on page 98](#).

---

**Note** - To remove the last cluster from a site, instead follow [“How to Delete a Site” on page 101](#).

---

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the cluster name, and click Remove. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **From a node of a site controller cluster, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

**2. Remove the cluster from the site.**

```
site-controller-cluster-node# geosite remove-member cluster site
```

```
cluster
```

The name of the cluster to remove from the site. You can remove multiple clusters by specifying each cluster name separated by a comma (,).

**3. From a remaining site cluster, verify that the removed cluster is no longer listed as a site member.**

```
# geosite show -v site
```

## Deleting a Site

This section contains procedures to delete a site. A site is automatically deleted when its last cluster is detached from the site. Each cluster must remove itself from the site. Because the last cluster to remove must be a site controller, first detach all clusters except one site controller. If the last cluster you remove is not a site controller, the command fails with an error.

When a site is deleted, all multigroups that the site references are automatically deleted on all clusters where the site was defined. The deletion of a multigroup has no effect on the protection groups that were configured in the deleted multigroup.

### ▼ How to Delete a Site

This procedure deletes a site by removing all of its member clusters. Perform this procedure from one site controller cluster.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, in the Clusters section highlight and click Remove for each remote cluster, then highlight the local cluster and click Remove.

For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**Before You Begin** Ensure that the site has at least one site controller cluster. If the site has no site controller cluster, change the role of a site member cluster to site controller before you perform this procedure. See [“How to Forcibly Change a Site Member to a Site Controller” on page 95](#).

1. **From a node of a site controller cluster, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Remove all other clusters from the site.**

```
# geosite remove-member list-of-members site
```

*list-of-members*

A comma-separated list of each cluster in the site, excluding the local cluster.

*site*

The name of the site to detach the specified clusters from.

3. **Verify that the local cluster is the only remaining member of the site.**

```
# geosite show -v site  
last-site-controller-cluster controller
```

4. **Forcibly detach the local cluster from the site.**

```
# geosite leave -f site
```

The site is automatically deleted when the last site cluster is removed.

**5. Verify that the site no longer exists.**

```
# geosite show -v site
```

```
*** Site "site" is not configured ***
```





## Administering Multigroups

---

This chapter contains the following information about administering multigroups:

- [“Modifying Protection Groups in a Multigroup”](#) on page 105
- [“Starting and Stopping Multigroups”](#) on page 110
- [“Synchronizing Multigroup Configuration Information”](#) on page 112
- [“Deleting a Multigroup”](#) on page 113

For procedures to create a new multigroup, see [“Configuring Sites and Multigroups”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

For procedures to switch over or take over multigroups, see [Chapter 11, “Migrating Services”](#).

### Modifying Protection Groups in a Multigroup

When you make a change to a multigroup, the site controller cluster that issues the change propagates the change to all other clusters in the site where the changed multigroup is active.

This section provides the following procedures to add, change, remove, and synchronize protection groups in a multigroup:

- [“How to Add a Protection Group to a Multigroup”](#) on page 105
- [“How to Change the Protection Groups in a Multigroup Dependency Chain”](#) on page 107
- [“How to Remove a Protection Group From a Multigroup”](#) on page 108
- [“How to Synchronize Multigroup Configuration Information”](#) on page 113

#### ▼ How to Add a Protection Group to a Multigroup

Perform this procedure to add one or more protection groups to a multigroup.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, click the multigroup name to go to its page, and click Add Protection Group. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

**Before You Begin** Ensure that a partner cluster of the protection group you are adding is configured in the site.

1. **From a node of a controller cluster for the site to configure with the new multigroup, assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the `root` role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Add the protection group to the multigroup.**

```
site-controller-cluster-node# geomg add-protection-group protection-group-list multigroup
```

The following describes the syntax choices for *protection-group-list*:

*cluster:protection-group*

Specifies a single protection group: The colon (:) separates the cluster name *cluster* from the name of the protection group that is configured in that cluster.

*cluster:protection-group/cluster:protection-group*

Specifies a protection group that has a dependency on another protection group, called a *dependency chain*: The protection group that is specified before the slash (/) in the dependency chain depends on the protection group that is specified after the slash.

*cluster1:protection-group1,cluster1:protection-group2,cluster2:protection-group1/cluster3:protection-group1*

The comma (,) separates multiple protection group names in the protection-group list.

*(cluster1:protection-group2,cluster2:protection-group1)/cluster3:protection-group1*

Specifies that multiple protection groups, *cluster1:protection-group2* and *cluster2:protection-group1*, all have a dependency on the *cluster3:protection-group1* protection group. In this form of dependency chain, parentheses are only used to enclose the multiple protection groups that have a dependency on another, single protection group. Only one protection group can be specified as the depended-on protection group in the dependency chain.

For more information, see the [geomg\(8\)](#) man page.

**3. Verify the addition of the protection groups.**

*site-controller-cluster-node# geomg show multigroup*

## ▼ How to Change the Protection Groups in a Multigroup Dependency Chain

Perform this procedure to change a set of protection groups in a multigroup that are configured in a dependency to or from each other. Such interdependent protection groups are referred to as a *dependency chain*. For more information about configuring dependency chains, see [“How to Add a Protection Group to a Multigroup” on page 105](#).

You must first delete the entire dependency chain of protection groups from the multigroup. You then add back any protection groups in the deleted dependency chain that you want to continue as part of the multigroup.

This procedure does not affect the protection groups themselves.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, click the multigroup name to go to its page, and use the Remove Protection Group and Add Protection Group buttons. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

**1. From a node of a site controller cluster, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

## 2. Remove the protection-group dependency chain from the multigroup.

Specify each protection-group dependency chain that you want to change.

```
# geomg remove-protection-group protection-group-list multigroup
```

*protection-group-list*

Name of one or more interdependency protection-groups to remove from the multigroup. Specify the full protection-group dependency that contains the protection group that you want to remove.

```
# geomg remove-protection-group dependee-protection-group/depended-on-protection-group multigroup
```

To remove multiple protection groups, with and without a dependency relationship, separate each protection group name with a comma (,).

```
# geomg remove-protection-group protection-group,dependee-protection-group/depended-on-protection-group multigroup
```

*multigroup*

The name of the multigroup from which you are removing the protection groups.

## 3. Add back those protection groups that you want to still be in the multigroup.

```
# geomg add-protection-group protection-group-list multigroup
```

# ▼ How to Remove a Protection Group From a Multigroup

Perform this procedure to remove a protection group from a multigroup. This procedure does not affect the protection group.

---

**Note** - If the protection group to remove is part of set of interdependent protection groups, called a dependency chain, do not perform this procedure. Go instead to [“How to Change the Protection Groups in a Multigroup Dependency Chain”](#) on page 107.

---

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, click the multigroup name to go to its page, highlight the name of the protection group, and click Remove Protection Group. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

1. **From a node of a site controller cluster, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Remove the protection group from the multigroup.**

- **If the protection group to remove is not part of a dependency chain with one or more other protection groups, use the following command:**

```
# geomg remove-protection-group cluster:protection-group multigroup
```

- **If the protection group to remove is part of a dependency chain with one or more other protection groups, use the following command to specify the complete dependency chain of protection groups:**

```
# geomg remove-protection-group protection-group-dependency-chain multigroup
```

```
cluster
```

Name of the cluster where the protection group to remove is configured.

*protection-group-dependency-chain*

Name of the complete dependency chain of protection groups that contains the protection group that you want to remove. For example, `newyork:pg1/newyork:pg2` is a dependency chain in which the `newyork:pg1` protection group depends on the `newyork:pg2` protection group.

To remove multiple protection groups, separate each protection group or dependency chain name with a comma (,).

3. **If a protection group that you removed was part of a dependency chain, add back any protection groups in the dependency chain that you want to continue to be in the multigroup.**

```
# geomg add-protection-group protection-group-list multigroup
```

## Starting and Stopping Multigroups

This section contains the following procedures:

- [“How to Start All Protection Groups in a Multigroup” on page 110](#)
- [“How to Stop All Protection Groups in a Multigroup” on page 111](#)

### ▼ How to Start All Protection Groups in a Multigroup

Perform this procedure to start all protection groups that are configured a multigroup. The protection groups to start can be either those on only the clusters that are specified in the protection-group list of the multigroup or those on both partner clusters where the protection groups are configured.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the multigroup name, and click Start Protection Groups. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **From a node of a site controller cluster, assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

## 2. Start all protection groups in the multigroup.

- To start all protection groups on only the clusters specified in the protection-group list of the multigroup, include the `-e local` option in the command.

```
# geomg start -e local multigroup
```

- To start all protection groups on both partner clusters where the protection groups are configured, include the `-e global` option in the command.

```
# geomg start -e global multigroup
```

See the [geomg\(8\)](#) man page for information about additional options for the `start` subcommand.

## ▼ How to Stop All Protection Groups in a Multigroup

Perform this procedure to stop all protection groups that are configured a multigroup. The protection groups to stop can be either those on only the clusters that are specified in the protection-group list of the multigroup or those on both partner clusters where the protection groups are configured.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the multigroup name, and click Stop Protection Groups. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

1. **From a node of a site controller cluster, assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the `root` role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Stop all protection groups in the multigroup.**
  - **To stop all protection groups on only the clusters specified in the protection-group list of the multigroup, include the `-e local` option in the command.**

```
# geomg stop -e local multigroup
```

- **To stop all protection groups on both partner clusters where the protection groups are configured, include the `-e global` option in the command.**

```
# geomg stop -e global multigroup
```

See the [geomg\(8\)](#) man page for information about additional options for the `stop` subcommand.

## Synchronizing Multigroup Configuration Information

Under normal circumstances, information about a multigroup configuration change is automatically synchronized among all clusters in the multigroup.

This section provides procedures to manually update the locally known configuration information of a multigroup with the configuration information known to another cluster in the site. Synchronization of a multigroup is normally performed automatically after you add or remove a protection group. You would perform this procedure if synchronization failed for one or more clusters of the multigroup site. The configuration information known to the local cluster is overwritten by information from the specified remote cluster in the site.



## ▼ How to Synchronize Multigroup Configuration Information

Perform this procedure to manually synchronize multigroup configuration information within a site.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the multigroup name, and click Validate. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

1. **From a node of a controller cluster of the local site, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Synchronize the multigroup configuration information with the remote cluster in the site.**

```
# geomg update remote-site-cluster multigroup-name
```

For more information about multigroup synchronization states, see the [geoadm\(8\)](#) man page.

## Deleting a Multigroup

This section describes procedures to delete a multigroup. This operation has no effect on the protection groups that are configured in the deleted multigroup.

Alternatively, if you intend to delete a site that references the multigroup you want to delete, instead follow [“How to Delete a Site” on page 101](#). When you delete a site, all multigroups that the site references are automatically deleted as well.

## ▼ How to Delete a Multigroup

Perform this procedure to delete a multigroup from all clusters of a site where the multigroup exists.

---

**Note** - You can also use the Oracle Solaris Cluster Manager browser interface to perform this task. Click Sites, click the site name to go to its page, highlight the multigroup name, and click Delete. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

1. **From a node of a controller cluster of the site where the multigroup exists, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Delete the multigroup.**

```
controller-cluster-node# geomg delete multigroup
```

3. **Verify that the multigroup is deleted.**

```
controller-cluster-node# geomg show multigroup
```

## Monitoring and Validating the Disaster Recovery Framework

---

This chapter describes the files and tools that you can use to monitor and validate the disaster recovery framework.

This chapter contains the following sections:

- [“Monitoring the Runtime Status of the Disaster Recovery Framework” on page 115](#)
- [“Viewing the Disaster Recovery Framework Log Messages” on page 122](#)
- [“Displaying Configuration Information for Partnerships and Protection Groups” on page 124](#)

Also see the applicable disaster recovery framework data replication guide for additional procedures to check the runtime status of that particular type of protection group.

### Monitoring the Runtime Status of the Disaster Recovery Framework

You can display the runtime status of the local disaster recovery framework enabled cluster by using the `geoadm status` command. When you run this command, it displays output that is organized in the following sections:

- **Cluster** – Provides the name of the local cluster
- **Partnership** – Provides information about all partnerships, including the name of the partner cluster, the synchronization state, the local heartbeats, and the local heartbeat plug-in
- **Protection group** – Provides information about the status of protection groups, including information about the local cluster and the remote cluster
- **Site** – Provides information about all sites, including the names of the sites, the name and role of each site cluster, site heartbeat, and site heartbeat plug-in,

- **Multigroup** – Provides information about all multigroups, including the names of the multigroups, the name of associated sites, and synchronization status
- **Pending operations** – Provides status information about any ongoing transaction processes

You must be assigned the Basic Solaris User rights profile to run the `geoadm status` command. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

For example, an administrator runs the `geoadm status` command on `cluster-paris` and the following information is displayed:

```
phys-paris-1# geoadm status

Cluster: cluster-paris

Partnership "paris-newyork-ps": OK
  Partner clusters      : cluster-newyork
  Synchronization      : OK
  ICRM Connection      : OK

Heartbeat "paris-to-newyork" monitoring "cluster-newyork": OK
  Plug-in "ping_plugin" : Inactive
  Plug-in "tcp_udp_plugin" : OK

Protection group "tcpg" : OK
  Partnership          : paris-newyork-ps
  Synchronization     : OK

Cluster cluster-paris : OK
  Role                 : Primary
  Activation state     : Deactivated
  Configuration       : OK
  Data replication    : OK
  Resource groups     : None

Cluster cluster-newyork : OK
  Role                 : Secondary
  Activation state     : Deactivated
  Configuration       : OK
  Data replication    : OK
  Resource groups     : None

Protection group "testpg" : OK
  Partnership          : paris-newyork-ps
  Synchronization     : OK

Cluster cluster-paris : OK
  Role                 : Primary
```

```

    Activation state : Deactivated
    Configuration   : OK
    Data replication : OK
    Resource groups : None

Cluster cluster-newyork : OK
  Role                 : Secondary
  Activation state     : Deactivated
  Configuration       : OK
  Data replication    : OK
  Resource groups     : None

Site : site1

  Controller "cluster-paris"
    Configuration : OK

  Controller "cluster-madrid"
    Configuration : OK
    Synchronization : OK

    Heartbeat "paris-to-madrid~site1-cluster-madrid" monitoring "cluster-
madrid": OK
      Plug-in "tcp_udp_plugin" : OK

  Member "cluster-london"
    Configuration : OK
    Synchronization : OK

    Heartbeat "paris-to-london~site1-cluster-london" monitoring "cluster-
london": OK
      Plug-in "tcp_udp_plugin" : OK

Multigroup "mg2" :
  Site : site1
  Configuration : OK
  Synchronization with cluster cluster-madrid : OK
  Synchronization with cluster cluster-london : OK

Multigroup "mg1" :
  Site : site1
  Configuration : OK
  Synchronization with cluster cluster-madrid : OK
  Synchronization with cluster cluster-london : OK

Pending operations:

Protection group "tcpg" operation: Start

```

Pending multigroup operations:

Multigroup "mg1" operation: Start

The information displayed shows that the protection group, `tcpg`, is started on both the primary cluster, `cluster-paris`, and the secondary cluster, `cluster-newyork`. Data is replicating between the partner clusters and both partners are synchronized.

The following table describes the meaning of the status values.

**TABLE 4** Status Value Descriptions

Field	Value Descriptions
Partnership	<p>OK – The partners are connected.</p> <p>Error – The connection between the partner clusters is lost.</p> <p>Degraded – The partnership has been successfully created but a connection with the partner cluster has not yet been established. This status value occurs when the partnership has been created and the partner cluster has not been configured.</p>
Synchronization	<p>OK – The configuration information is synchronized between partner clusters.</p> <p>Error – The configuration information differs between the partner clusters. You need to resynchronize the partnership for a partnership synchronization error, or resynchronize the protection group, for a protection group synchronization error.</p> <ul style="list-style-type: none"> <li>■ For information about resynchronizing a partnership, see <a href="#">“Resynchronizing a Partnership” on page 52</a>.</li> <li>■ For information about resynchronizing a protection group, see <a href="#">“Resynchronizing a Protection Group” on page 80</a>.</li> </ul> <p>Mismatch – Configuration information has been created separately on the clusters. The configuration information must be replaced by a copy of the configuration information from the partner cluster. You can synchronize the protection group configuration by using the <code>geopg get</code> command.</p> <p>Unknown – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.</p>
ICRM Connection	<p>OK – The Intercluster Resource Management (ICRM) module is running properly.</p> <p>Error – The ICRM module on the local cluster is unable to communicate with the ICRM module on the remote cluster.</p>
Heartbeat	<p>OK – Heartbeat checks are running and the partner cluster responds within the specified timeout and retry periods.</p> <p>Offline – Heartbeat checks are not running.</p> <p>Error – Heartbeat checks are running but the partner is not responding and retries have timed out.</p>

Field	Value Descriptions
	Degraded – Heartbeat checks are running but one of the primary heartbeat plug-ins is degraded or is not running.
Heartbeat plug-in	OK – Responses are being received from the partner.  Inactive – Plug-in is not in use but is a standby for retrying to contact the partner if the other plug-ins obtain no response.  No-Response – Partner cluster is not responding.
Protection group (overall protection group state)	OK – No component of the protection group on either partner is in the Degraded, Error, or Unknown status, and the protection group configuration is the same on both partner clusters.  Degraded – Data replication is either not running or is in a partial error state.  Error – At least one component of the partnership is in an error state on at least one partner, or the protection group configuration is different between the partner clusters.  Unknown – The status for at least one component of the protection group is unknown, or the status of the protection group is not accessible.
Protection group > Synchronization  (state of protection group configuration information between partner clusters)	OK – The configuration is synchronized between partner clusters.  Error – The configuration on the partner clusters is different. You must synchronize the protection group again.  Mismatch – The protection group has been configured on each partner cluster individually. You must remove the configuration from one cluster and copy the configuration of the partner cluster.  Unknown – Information is not accessible because the partners are disconnected.
Protection group > Cluster (state of protection group on each cluster)	None – The data replication or resource group component is not configured in the protection group.  OK – The state of all the protection group components, such as configuration data, data replication, or resource groups, is OK, NONE, or N/A on the cluster.  Degraded – The state of one or more of the protection group components is in the Degraded state on the cluster.  Error – The state of some components of the protection group, such as configuration data, data replication, or resource groups, is in Error.  Unknown – The state of some components of the protection group, such as configuration data, data replication, or resource groups, is unavailable.
Protection group > Cluster > Role	Primary – The cluster is the Primary for this protection group.  Secondary – The cluster is the Secondary for this protection group.  Unknown – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.

Field	Value Descriptions
Protection group > Cluster > Activation state	<p>Activated – The protection group is activated.</p> <p>Deactivated – The protection group is deactivated.</p> <p>Unknown – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.</p>
Protection group > Cluster > Configuration	<p>OK – Protection group configuration has been validated without errors on the cluster.</p> <p>Error – Protection group configuration validation resulted in errors on the cluster. You need to revalidate the protection group. For information about validating a protection group, see <a href="#">“Validating a Protection Group” in <i>Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i></a>.</p> <p>Unknown – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.</p>
Protection group > Cluster > Data replication	<p>None – Data replication is not configured.</p> <p>OK – Data replication is running and data is synchronized with the partner cluster when the protection group is activated. Replication is suspended when the protection group is deactivated. This state represents data replication on this cluster and does not reflect the overall state of data replication. This state is mapped from the corresponding state in the data replication subsystem.</p> <p>Degraded – Data is not replicated and not synchronized with the partner cluster when the protection group is activated. New writes will succeed but not be replicated. This state represents data replication on this cluster and does not reflect the overall state of data replication. This state is mapped from the corresponding state in the data replication subsystem.</p> <p>Error – Data replication from the primary cluster to the secondary cluster is in error if the data replication subsystem reports an error or if data replication is not suspended when the protection group is deactivated. This state represents data replication on this cluster and does not reflect the overall state of data replication. This state is mapped from the corresponding state in the data replication subsystem.</p> <p>Unknown – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.</p>
Protection group > Cluster > Resource groups	<p>None – No resource group is protected by this protection group.</p> <p>OK – If the cluster has the Primary role, all resource groups are online when the protection group is activated or unmanaged when the protection group is deactivated. If the cluster has the Secondary role, all resource groups are unmanaged.</p> <p>Error – If the cluster has the Primary role, not all resource groups are online when the protection group is activated or unmanaged when the protection group is deactivated. If the cluster has the Secondary role, not all resource groups are unmanaged.</p>



Field	Value Descriptions
	Unknown – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.
Site > Controller or Member > Configuration	OK – The site configuration is correct. Error – The site configuration has an error. Unknown – The site configuration is not checked.
Site > Controller or Member > Synchronization	DIFFERENT – Site configuration information on two compared clusters is different but compatible. This is a transitory status. Eventually, the older of the two configurations is automatically synchronized with the newer configuration, at which time the synchronization status becomes OK. If for some reason the status remains at DIFFERENT, use the <code>geosite validate</code> subcommand to synchronize the site configurations.  ERROR – The site configuration on two compared clusters is different and cannot be automatically resolved. Configuration changes from an issuing controller cluster are not accepted by a cluster while it is in the ERROR synchronization status with the issuing controller. If configuration information between two clusters do not resolve automatically and the clusters are in the ERROR status, use the <code>geosite update</code> subcommand to resolve site configuration conflicts between the two clusters.  OK – Site configuration information matches on the compared clusters.  UNKNOWN – Site configuration information cannot be compared because the Geographic Edition software cannot reach a cluster. Configuration changes from an issuing controller cluster are not propagated to a cluster while it is in the UNKNOWN synchronization status. If the cluster is not automatically synchronized, use the <code>geosite validate</code> subcommand to update the cluster with the latest site configuration information.
Site > Controller or Member > Heartbeat	OK – Heartbeat monitoring is enabled, and the partner cluster is responding within timeout and retry periods.  Degraded – Heartbeat checks are running but one of the primary heartbeat plug-ins is degraded or is not running.  Error – Heartbeat monitoring is running but the partner cluster is not responding and retries have timed out.  Offline – Heartbeat monitoring is offline.
Site > Controller or Member > Heartbeat > Plug-in	OK – The partner cluster is responding.  Inactive – The plug-in is not in use. It is a standby plug-in that is used for retrying if other plug-ins do not respond.  No-response – The partner cluster is not responding.
Multigroup > Configuration	OK – The configuration of the multigroup is correct and validated without errors.  Error – The configuration of the multigroup has errors. Run the <code>geomg validate</code> command on the multigroup to identify the error.

Field	Value Descriptions
	Unknown – The configuration is not accessible.
Multigroup > Synchronization	<p>DIFFERENT – Multigroup configuration information on two compared clusters is different but compatible. This is a transitory status. Eventually, the older of the two configurations is automatically synchronized with the newer configuration, at which time the synchronization status becomes OK. If for some reason the status remains at DIFFERENT, use the <code>geomg validate</code> command to synchronize the multigroup configurations.</p> <p>ERROR – The multigroup configuration on two compared clusters is different and cannot be automatically resolved. Configuration changes from an issuing controller cluster are not accepted by a cluster while it is in the ERROR synchronization status with the issuing controller. If configuration information between two clusters do not resolve automatically and the clusters are in the ERROR status, use the <code>geomg update</code> command to resolve multigroup configuration conflicts between the two clusters.</p> <p>OK – Multigroup configuration information matches on the compared clusters.</p> <p>UNKNOWN – Multigroup configuration information cannot be compared because the Geographic Edition software cannot reach a cluster. Configuration changes from an issuing controller cluster are not propagated to a cluster while it is in the UNKNOWN synchronization status. If the cluster is not automatically synchronized after the disaster recovery framework framework is started on that cluster, use the <code>geomg validate</code> command to update the cluster with the latest multigroup configuration information.</p>

## Viewing the Disaster Recovery Framework Log Messages

All the disaster recovery framework components produce messages that are stored in log files.

Information about the loading, running, and stopping disaster recovery framework components in the common agent container is recorded in the following log files. The most recently logged messages are in file 0, then 1, and 2.

- `/var/cacao/instances/default/logs/cacao.0`
- `/var/cacao/instances/default/logs/cacao.1`
- `/var/cacao/instances/default/logs/cacao.2`

System log messages are stored in the `/var/adm/messages` log file.

Each cluster node keeps separate copies of the previous log files. The combined log files on all cluster nodes form a complete snapshot of the currently logged information. The log messages of the disaster recovery framework modules are updated on the node where the

disaster recovery framework is currently active. The data replication control-log messages are updated on the node where the data replication resource is currently Online.

For data replication modules that are based on script-based plug-ins, you can set the DEBUG property to TRUE for more verbose messages to aid in troubleshooting.

cacao log settings help in debugging the disaster recovery framework core framework and the replication modules. The `geocontrol`, `geoutilities` and `notifier` are disaster recovery framework sub modules. `geocontrol` corresponds to core and replication modules, `geoutilities` applies to all utility related API's, and `notifier` applies to event notification processing.

Debugging must be enabled on all the nodes of the cluster and the nodes of the partners of interest.

To turn on debug so that it does **not** persists across cacao restarts, type:

```
# cacaoadm set-filter com.sun.cluster.agent.geoutilities=FINEST
# cacaoadm set-filter com.sun.cluster.agent.geocontrol=FINEST
# cacaoadm set-filter com.sun.cluster.agent.notifier=FINEST
```

To turn on debug so that it persists across cacao restarts, type:

```
# cacaoadm set-filter -p com.sun.cluster.agent.geoutilities=FINEST
# cacaoadm set-filter -p com.sun.cluster.agent.geocontrol=FINEST
# cacaoadm set-filter -p com.sun.cluster.agent.notifier=FINEST
```

When debugging is turned on, the number of messages sent to `/var/cacao/instances/default/logs/cacao.0` greatly increases. If needed, increase the limit allowed in the log file by updating the `log-file-limit` parameter. Updating this parameter requires a cacao stop. Stopping cacao on a node running `geo-failovercontrol` causes this resource to restart. To avoid this, you can update `log-file-limit` on nodes that are not running `geo-failovercontrol`, then switch `geo-infrastructure rg` to a node where `log-file-limit` has been updated, then update the node where `geo-infrastructure` was running.

To update `log-file-limit`, type:

```
# cacaoadm stop
# cacaoadm set-param log-file-limit=<new-value-desired>
# cacaoadm start
```

## Displaying Configuration Information for Partnerships and Protection Groups

You can display the current local cluster partnership configuration, including a list of all partnerships that are defined between the local cluster and remote clusters.

You can also display the current configuration of a specific protection group or of all the protection groups that are defined on a cluster.

This section provides the following procedures:

- [“How to Display Configuration Information About Partnerships” on page 124](#)
- [“How to Display Configuration Information About Protection Groups” on page 125](#)

### ▼ How to Display Configuration Information About Partnerships

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, then click the partnership name. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

#### 1. Log in to a cluster node.

You must be assigned the Basic Solaris User rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

#### 2. Display information about the partnership.

```
# geops list partnership-name
```

*partnership-name*

Specifies the name of the partnership. If you do not specify a partnership, then the `geops list` command displays information on all partnerships.

For information about the names and values that are supported by the disaster recovery framework, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities”](#).

**Example 30** Displaying Partnership Configuration Information

This example displays configuration information about the partnership between local `cluster-paris` and remote `cluster-newyork`.

```
# geops list paris-newyork-ps
```

## ▼ How to Display Configuration Information About Protection Groups

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name to go to its page, and click the protection group name. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager” in \*Administering an Oracle Solaris Cluster 4.4 Configuration\*](#).

---

- 1. Log in to a cluster node.**

You must be assigned the Basic Solaris User rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

- 2. Display information about a protection group.**

```
# geopg list [protection-group]
```

*protection-group*

Specifies the name of a protection group.

If you do not specify a protection group, then the command lists information about all the protection groups that are configured on your system.

**Example 31** Displaying Configuration Information About a Protection Group

This example displays configuration information for `zfssapg`, which is configured on `cluster-paris`.

```
# geopg list zfssapg
```



## Migrating Services

---

This chapter provides information about detecting cluster failure and moving services to an accessible cluster.

- [“Detecting Cluster Failure” on page 127](#)
- [“Migrating Replication Services by Switching Over Protection Groups” on page 129](#)
- [“Forcing a Takeover of a Protection Group” on page 133](#)
- [“Recovering a Protection Group to a Cluster” on page 138](#)

### Detecting Cluster Failure

This section describes the internal processes that occur when failure is detected on a primary or a secondary cluster.

- [“Detecting Primary Cluster Failure” on page 127](#)
- [“Detecting Secondary Cluster Failure” on page 128](#)

### Detecting Primary Cluster Failure

When the primary cluster for a protection group fails, the secondary cluster in the partnership detects the failure. The cluster that fails might be a member of more than one partnership, resulting in multiple failure detections.

The following actions take place when a primary cluster failure occurs. During a failure, the appropriate protection groups are in the Unknown state on the cluster that failed.

- Heartbeat failure is detected by a partner cluster.

- The heartbeat is activated in emergency mode to verify that the heartbeat loss is not transient and that the primary cluster has failed. The heartbeat remains in the `Online` state during this default time-out interval, while the heartbeat mechanism continues to retry the primary cluster.

This query interval is set by using the `Query_interval` heartbeat property. If the heartbeat still fails after the interval you configured, a heartbeat-lost event is generated and logged in the system log. When you use the default interval, the emergency-mode retry behavior might delay heartbeat-loss notification for about nine minutes. Messages are displayed in the Oracle Solaris Cluster Manager browser interface and in the output of the `geoadm status` command.

For more information about logging, see [“Viewing the Disaster Recovery Framework Log Messages” on page 122](#).

- If the partnership is configured for heartbeat-loss notification, then one or both of the following actions occurs:
  - An email is sent to the address specified in the `Notification_emailaddrs` property.
  - The script defined in `Notification_actioncmd` is executed.

For more information about configuring heartbeat-loss notification, see [“Configuring Heartbeat-Loss Notification” on page 69](#).

## Detecting Secondary Cluster Failure

When a secondary cluster for a protection group fails, a cluster in the same partnership detects the failure. The cluster that failed might be a member of more than one partnership, resulting in multiple failure detections.

During failure detection, the following actions take place:

- Heartbeat failure is detected by a partner cluster.
- The heartbeat is activated in emergency mode to verify that the secondary cluster is dead.
- When a failure is confirmed by the disaster recovery framework, the cluster notifies the administrator. The system detects all protection groups for which the cluster that failed was acting as secondary. The state of the appropriate protection groups is marked `Unknown`.



## Migrating Replication Services by Switching Over Protection Groups

Perform a switchover of a protection group when you want to migrate services to the partner cluster in an orderly fashion. You can switch over an individual protection group or switch over a multigroup of multiple protection groups in a single operation.

This section contains the following information:

- [“Actions Performed by the Disaster Recovery Framework During a Switchover” on page 129](#)
- [“How to Switch Over Replication From the Primary Cluster to the Secondary Cluster” on page 131](#)
- [“How to Switch Over a Multigroup” on page 132](#)

### Actions Performed by the Disaster Recovery Framework During a Switchover

A switchover consists of the following actions:

- Application services are unmanaged on the former primary cluster.
- The replication role is reversed and now continues to run from the new primary.
- Application services are brought online on the new primary cluster.

When you run the `geopg switchover` command, the software confirms that the secondary cluster does indeed allow the replication role reversal. The command creates copies of the replicated source projects and exports the shares for use on the target appliance to ensure that no projects and mount point conflicts can occur. The command then destroys such created clones on the target appliance and confirms that the actual reverse replication operation can be performed on the target appliance.

For Oracle Data Guard, before the switchover, the command checks that the remote database is in an `enabled` state in the Oracle Data Guard Broker configuration. The command also confirms that the configuration is healthy by issuing the Oracle Data Guard command-line interface (`dgmgrl`) `show configuration` command to ensure that the command returns a `SUCCESS` state. If the output from this command indicates that Oracle Data Guard Broker is busy performing its own health check, the Oracle Data Guard command-line interface retries the command until it

receives a SUCCESS response or until two minutes have passed. If the command-line interface is unable to get a SUCCESS response, the command fails.

For Oracle ZFS Storage Appliance, the command creates copies of the replicated source projects and exports the shares for use on the target appliance to ensure that no projects and mount point conflicts can occur. The command then destroys such created clones on the target appliance and confirms that the actual reverse replication operation can be performed on the target appliance.

The software performs the following actions on the original primary cluster:

- Takes offline the application resource groups in the protection group and places them in the Unmanaged state.
- Performs a switchover to the secondary cluster for each replication configuration in the protection group.

On the original secondary cluster, the command takes the following actions:

- Runs the script that is defined in the RoleChange\_ActionCmd property
- Brings online all application resource groups in the protection group

If the command completes successfully, the secondary cluster becomes the new primary cluster for the protection group. The original primary cluster becomes the new secondary cluster. The application resource groups in the protection group are brought online on the new primary cluster and replication from the appliance that is connected from the new primary cluster to the new secondary cluster begins.

For Oracle Data Guard, databases that are associated with the Oracle Data Guard Broker configurations of the protection group have their role reversed according to the role of the protection group on the local cluster. For HA for Oracle configurations, the dataguard\_role resource property is also updated with the status of the new primary and standby clusters. The shadow Oracle database server resource group and any other application resource groups in the protection group are online on the new primary cluster. Data replication from the new primary cluster to the new standby cluster begins.

The `geopg switchover` command returns an error if any of the previous operations fails. Run the `geoadm status` command to view the status of each component. For example, the Configuration status of the protection group might be set to `Error`, depending on the cause of the failure. The protection group might be activated or deactivated.

If the Configuration status of the protection group is set to `Error`, revalidate the protection group by using the procedures that are described in [“Validating a Protection Group” in \*Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

If the configuration of the protection group is not the same on each partner cluster, you need to resynchronize the configuration by using the procedures that are described in [“Resynchronizing](#)

a Protection Group” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

## ▼ How to Switch Over Replication From the Primary Cluster to the Secondary Cluster

Perform this procedure to switch over a protection group to the secondary cluster.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface, if you do not need to use the `-f` force option. Click Partnerships, click the partnership name to go to its page, highlight the protection group name, and click Switchover. For Oracle Solaris Cluster Manager log-in instructions, see “[How to Access Oracle Solaris Cluster Manager](#)” in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

**Note** - To switch over a set of protection groups that are configured as a multigroup, instead follow procedures in “[How to Switch Over a Multigroup](#)” on page 132.

---

**Before You Begin** Before you switch over a protection group from the primary cluster to the secondary cluster, ensure that the following conditions are met:

- For Oracle ZFS Storage Appliance, data replication is active between the primary cluster and the secondary cluster. That is, the replication is enabled from the source to the target appliances.
- For Oracle ZFS Storage Appliance, the disaster recovery framework replication resource for this appliance replication shows the `Online` state.
- For Oracle Data Guard, the Oracle Data Guard Broker `show configuration` command must show a `SUCCESS` state. This state is reflected in the state of the disaster recovery framework replication resource for this Oracle Data Guard Broker configuration, which should show the `online` state.
- The disaster recovery framework is running on both clusters.
- The secondary cluster is a member of a partnership.
- Both cluster partners can be reached.
- The overall state of the protection group is set to `OK`.

1. **Assume the `root` role or assume a role that is assigned the Geo Management rights profile.**

For more information, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and the data replication software.

---

## 2. Initiate the switchover.

The application resource groups that are a part of the protection group are stopped and started during the switchover.

```
phys-paris-1# geopg switchover [-f] -m new-primary-cluster protection-group-name
```

`-f`

Forces the command to perform the operation without asking you for confirmation.

`-m new-primary-cluster`

Specifies the name of the cluster that is to be the new primary cluster for the protection group.

`protection-group-name`

Specifies the name of the protection group.

### Example 32 Forcing a Switchover From the Primary Cluster to the Secondary Cluster

The following example performs a switchover to the secondary cluster `cluster-newyork`.

```
phys-paris-1# geopg switchover -f -m cluster-newyork example-protection-group
```

**Next Steps** To fail back a protection group to the original cluster, see [“Recovering a Protection Group to a Cluster” on page 138](#).

## ▼ How to Switch Over a Multigroup

Perform this procedure to switch operation of all protection groups in a multigroup to partner clusters in the specified site.

---

**Note** - To switch over an individual protection group, instead follow procedures in [“How to Switch Over Replication From the Primary Cluster to the Secondary Cluster”](#) on page 131.

---

**Before You Begin** Ensure that the multigroup is started.

1. **From a node of a controller cluster in the target site, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Switch the multigroup to the partner clusters in the specified site.**

```
# geomg switchover -s site multigroup
```

```
-s site
```

The name of the site that the multigroup is to switch to.

```
multigroup
```

The name of the multigroup to switch over.

See the [geomg\(8\)](#) man page for information about additional options for the switchover subcommand.

**Next Steps** To fail back a protection group to the original cluster, see [“Recovering a Protection Group to a Cluster”](#) on page 138.

## Forcing a Takeover of a Protection Group

Perform a takeover when applications need to be brought online on the secondary cluster, regardless of whether the data is completely consistent between the primary and the secondary.

You can take over an individual protection group or take over multiple protection groups that are configured as a multigroup in a single operation.

The information in this section assumes that the protection group has been started.

This section contains the following information:

- [“Actions Performed by the Disaster Recovery Framework During a Takeover” on page 134](#)
- [“How to Force Immediate Takeover of a Protection Group by a Secondary Cluster” on page 135](#)
- [“How to Take Over a Multigroup” on page 136](#)

## Actions Performed by the Disaster Recovery Framework During a Takeover

A takeover consists of the following actions:

- If the former primary cluster can be reached and the protection group is not locked for notification handling or some other reason, the protection group is deactivated.
- The data of the former primary cluster `cluster-paris` is taken over by the new primary cluster.

---

**Note** - The data on the former primary cluster `cluster-parisC` might not be consistent with the data on the original primary cluster. Data replication from the new primary cluster to the former primary cluster is stopped.

---

- Application services are brought online on the new primary cluster.
- The protection group is activated without replication enabled.

For more details about takeover and the effects of the `geopg takeover` command, see [Appendix C, “Disaster Recovery Administration Example,” in \*Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

For details about the possible conditions of the primary and secondary cluster before and after a takeover, see [Appendix D, “Takeover Postconditions,” in \*Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

## ▼ How to Force Immediate Takeover of a Protection Group by a Secondary Cluster

Perform this procedure to take over an individual protection group by the secondary cluster.

---

**Note** - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface, if you do not need to use the `-f` force option. Click Partnerships, click the partnership name to go to its page, highlight the protection group name, and click Takeover. For Oracle Solaris Cluster Manager log-in instructions, see [“How to Access Oracle Solaris Cluster Manager”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

---

---

**Note** - To take over a set of protection groups that are configured as a multigroup, instead follow procedures in [“How to Take Over a Multigroup”](#) on page 136.

---

**Before You Begin** Before you force the secondary cluster to assume the activity of the primary cluster, ensure that the following conditions are met:

- The disaster recovery framework is up and running on the secondary cluster.
- The secondary cluster is a member of a partnership.
- The Configuration status of the protection group is OK on the secondary cluster.
- If you are using EMC Symmetrix Remote Data Facility, ensure that no LUNs in the data replication device group are in the SyncInProgress state. Otherwise, the application might fail to start on the new primary cluster due to data inconsistencies.

Perform this procedure from a node in the secondary cluster.

1. **Assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and the data replication software.

---

## 2. Initiate the takeover.

```
phys-newyork-1# geopg takeover [-f] protection-group-name
```

-f

Forces the command to perform the operation without your confirmation, even if the primary cluster for a protection group in the multigroup is reachable and the protection group is active on that cluster.

*protection-group-name*

Specifies the name of the protection group.

### Example 33 Forcing a Takeover by a Secondary Cluster

The following example forces the takeover of the protection group `example-pg` by the secondary cluster `cluster-newyork`.

The node `phys-newyork-1` is the first node of the secondary cluster. For a reminder of which node is `phys-newyork-1`, see [“Example Disaster Recovery Framework Cluster Configuration” in \*Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

```
phys-newyork-1# geopg takeover -f example-pg
```

**Next Steps** For information about the state of the primary and secondary clusters after a takeover, see [Appendix D, “Takeover Postconditions,” in \*Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4\*](#).

To fail back a protection group to the original cluster, see [“Recovering a Protection Group to a Cluster” on page 138](#).

## ▼ How to Take Over a Multigroup

Perform this procedure to transfer all operation of protection groups in a multigroup to partner clusters in the specified site.

---

**Note** - To take over a protection groups that is not configured as a multigroup, instead follow procedures in [“How to Force Immediate Takeover of a Protection Group by a Secondary Cluster” on page 135](#).

---






---

**Caution** - To avoid possible loss of replicated data, use instead the `geomg switchover` command. The `geomg takeover` command is intended only for situations where the potential risk of data loss is justified, such as when the primary site is not available or during unplanned downtime.

---

1. **From a node of a controller cluster of the target site, assume the root role or assume a role that is assigned the Geo Management rights profile.**

For more information, see “[Disaster Recovery Framework Rights Profiles](#)” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

---

**Note** - If you use a role with Geo Management rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management rights profile and data replication software.

---

2. **Initiate takeover of the multigroup by the standby site.**

```
# geomg takeover [-f] -s site multigroup
```

`-f`

Forces takeover, even if the primary cluster for a protection group in the multigroup is reachable and the protection group is active on that cluster.

`-s site`

The name of the standby site which is to take over the specified multigroup.

`multigroup`

The name of the multigroup to take over.

See the `geomg(8)` man page for information about additional options for the takeover subcommand.

**Next Steps** For information about failing back a protection group to the original cluster, see “[Recovering a Protection Group to a Cluster](#)” on page 138.

## Recovering a Protection Group to a Cluster

After a successful takeover operation, the secondary cluster becomes the primary for the protection group and the services are online on the secondary cluster. After the recovery of the original primary cluster the services can be brought online again on the original primary by using a process called *failback*.

The disaster recovery framework supports the following kinds of failback:

- **Failback-switchover.** During a failback-switchover, applications are brought online again on the original primary cluster after the data of the original primary cluster was resynchronized with the data on the secondary cluster.
- **Failback-takeover.** During a failback-takeover, applications are brought online again on the original primary cluster and use the current data on the original primary cluster. Any updates that occurred on the secondary cluster while it was acting as primary are discarded.

If you want to leave the new primary as the primary cluster and the original primary cluster as the secondary after the original primary restarts, you can resynchronize and revalidate the protection group configuration without performing a switchover or takeover.

For procedures to perform a failback-switchover or a failback-takeover, see the Oracle Solaris Cluster documentation for the data replication product you are using.

## Customizing Switchover and Takeover Actions

---

This chapter describes how to create a script that runs when the role of a protection group changes from secondary to primary. The chapter contains the following sections:

- [“Creating a Role-Change Action Script” on page 139](#)
- [“Configuring a Protection Group to Run a Script at Switchover or Takeover” on page 142](#)

### Creating a Role-Change Action Script

You can configure the disaster recovery framework to run a command when a cluster within a protection group changes from the secondary to the primary role. This change can happen as a result of either a switchover or takeover operation.

The action command runs during a switchover or takeover on the new primary cluster when the protection group is started on the new primary cluster. The script is invoked on the new primary cluster after the data replication role changes from secondary to primary and before the application resource groups are brought online. If the data replication role change does not succeed, then the script is not called.

Observe the following requirements for the role-change action script:

- The path to this script must be valid on all nodes of all partner clusters that can host the protection group.
- The script must have execute permissions for the user that launches the script. The intended user can be the `root` role, or an administrator with the necessary `solaris.cluster.*` authorization to execute a switchover or takeover operation plus any other actions that the script performs. For more information about user rights for the disaster recovery framework, see [Chapter 4, “Administering Rights Profiles”](#).
- The script must begin with the path to the shell, such as `#!/bin/ksh`.

The following command-line runs the script:

```
# custom-action-command-path -o primary -c cluster-name \  
-s partnership-name protection-group user-arguments
```

*custom-action-command-path*

Specifies a path to the action command you have created.

*-o primary*

Specifies that the role being assumed by the cluster is primary.

*-c cluster-name*

Specifies the name of the secondary cluster that is assuming the new role of primary cluster.

*-s partnership-name*

Specifies the name of the partnership that hosts the protection group.

*protection-group*

Specifies the name of the protection group that is undergoing the role change.

*user-arguments*

Specifies static arguments that are passed after all the disaster recovery framework supplied options.

This free-form string can be parsed by the script as required. For example, you could specify a list of key=value pairs, such as name=example.com, ip=192.0.2.3. You could also specify a sequence of options, such as -n example.com -a 192.0.2.3. The format of these arguments is not restricted by the disaster recovery framework.

The exit status of the role-change action script is reported as part of the result of the `geopg switchover` or `geopg takeover` command. The exit status is zero if the action script was started successfully. A nonzero exit status indicates an error or failure. The value of the exit status does not affect other aspects of the role-change actions. The switchover or takeover proceeds to bring the application resource groups in the protection group online, regardless of the exit status of the action script.

The exit status of the action script can impact the `geomg switchover` or `geomg takeover` commands. During switchover or takeover of a multigroup, if the action script returns a nonzero exit status for a protection group that has dependents, the switchover or takeover of those dependent protection groups is not performed.

The disaster recovery framework waits for the script to return before the software processes operations such as bringing online application resource groups. Therefore, you must know in

advance the amount of time required to run the script when you create the action script so that you can set the timeout period for the protection group accordingly. Setting the timeout period to include enough time for the script to complete to avoid switchovers or takeovers timing out and leaving the application resource group offline on the new primary.

**EXAMPLE 34** Switchover Action Script for Updating the DNS

This sample script uses the `nsupdate` command to reconfigure the host name to point to a new cluster. For more information about the `nsupdate` command, refer to the [nsupdate\(1\)](#) man page.

Clients that try to connect to `companyX.com` are referred by the name service to the address of the primary cluster for a protection group, `cluster-paris`. When the primary cluster fails to respond, the administrator performs a switchover of the protection group to the alternative cluster, `cluster-newyork`.

```
#!/bin/ksh
# sample script to update dns
# Assumes each cluster has an entry with name "lh-paris-1" in /etc/hosts
# but different value for the IP in each cluster
# for forward DNS (A) entry: will delete old entry for "lh-paris-1"
# and add one that is correct for "this cluster"
#
# For reverse (PTR) DNS entry, will just add one for this cluster.
# Will NOT delete PTR record left over from old cluster. So
# eventually you will just have reverse lookup for the IP for both clusters
# doing reverse resolution to the same name (lh-paris-1.odyssey.com)
# This should be fine, as long as the forward resolution stays "correct"
#
# The blank line of input at the end of nsupdate is REQUIRED
#
# A short TTL is put on the new records (600 = 10 minutes)
# but you can't really control what kind of caching goes on on
# the client side

# get IP corresponding to name "lh-paris-1" on THIS Cluster
NEWIP=$(getent hosts lh-paris-1|cut -f1)

# this bit splits out the octets in order to add the reverse PTR entry
IFS=.
set $NEWIP
unset IFS

/usr/sbin/nsupdate <<ENDNSUPDATE
update delete ora-lh.odyssey.com A
update add ora-lh.odyssey.com 600 A $NEWIP
```

```
update add $4.$3.$2.$1.in-addr.arpa 600 PTR ora-lh.odyssey.com.
```

```
ENDNSUPDATE
```

## Configuring a Protection Group to Run a Script at Switchover or Takeover

After you have created a script, you must configure the protection group to run the script when a switchover or takeover occurs. If a switchover or takeover occurs, the script runs on the cluster that is becoming the new primary cluster.

### ▼ How to Configure a Protection Group to Run a Script at Switchover or Takeover

**1. Log in to a cluster node.**

You must be assigned the Geo Management rights profile to complete this procedure. For more information, see [“Disaster Recovery Framework Rights Profiles”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

**2. Configure the `RoleChange_ActionCmd` and `RoleChange_ActionArgs` properties of the protection group.**

```
# geopg set-prop -p RoleChange_ActionCmd=fully-qualified-script \  
-p RoleChange_ActionArgs=script-arguments protection-group-name
```

*-p property-setting*

Specifies the properties of the protection group.

Specify the path to the command by using the `RoleChange_ActionCmd` property. This path should be valid on all nodes of all partner clusters that can host the protection group.

Define the arguments that you want to append to the command line when the action command is run by using the `RoleChange_ActionArgs` property.

For more information about the properties you can set, see [Appendix A, “Standard Disaster Recovery Framework Properties”](#).

*protection-group*

Specifies the name of the protection group.

**Example 35** Configuring a Protection Group to Run a Command at Cluster Switchover or Takeover

This example configures a protection group to run a custom command called newDNS.

```
# geopg set-prop -p RoleChange_ActionCmd=/usr/bin/newDNS \  
-p RoleChange_ActionArgs=domain=example.com,ip=192.0.2.4 avspg
```





## Script-Based Plug-Ins

---

This chapter provides information about disaster recovery framework script-based plug-ins. It covers the following topics:

- “Introduction to Disaster Recovery Framework Script-Based Plug-Ins” on page 145
- “Property Descriptions for Script-Based Plug-Ins” on page 149
- “Internals for Script-Based Plug-Ins” on page 160

### Introduction to Disaster Recovery Framework Script-Based Plug-Ins

The disaster recovery framework provides modules to support data replication software products including Oracle Data Guard, Oracle ZFS Storage Appliance, and Oracle MySQL. However, the creation of such modules requires detailed knowledge of both the replication software and the internals of the disaster recovery framework. The disaster recovery framework uses the common agent container with a number of Java management beans (MBeans) that form the interface for the disaster recovery framework monitoring and management infrastructure and the replication control software.

By providing a more generic interface module analogous to the Oracle Solaris Cluster Generic Data Service (GDS), the disaster recovery framework script-based plug-in enables you to rapidly integrate additional replication technologies by supplying a few interface scripts to fulfill the necessary control functions. This capability frees you from needing to learn the internals of the disaster recovery framework or needing any knowledge of Java technology or MBeans. Instead, you can focus on the replication technology you need to protect your enterprise data. For more information on the Generic Data Service, see [Developing Agents With the Generic Data Service \(GDS\)](#).

For simplicity, the term script is used throughout this document to represent any compiled binary or script-based executable.

This section provides the following information:

- [“Advantages and Disadvantages of Using Script-Based Plug-Ins” on page 146](#)
- [“Script-Based Plug-In Architecture” on page 146](#)
- [“Restrictions of Script-Based Plug-Ins” on page 148](#)
- [“Creating Script-Based Plug-Ins” on page 148](#)

## Advantages and Disadvantages of Using Script-Based Plug-Ins

The main advantage of using the script-based plug-in comes from reducing the barriers to implementing new replication mechanisms. Rather than spending time learning about the Java, JMX, MBeans, or common agent container technologies, you can focus on the critical logic needed, for example, to set up a replicated configuration or change the direction of the replication flow.

The disadvantage of this approach stems from the very generic nature of the plug-in that makes it so easy to use. Generic plug-ins lack some of the tight integration that a custom module can offer. For example, the arguments that you supply on the command line to a script-based plug-in configuration are at the script argument level rather than the highly specific replication variable level. So, whereas the disaster recovery framework Oracle Data Guard module has separate, specific arguments for properties like `standby_type`, an equivalent script-based plug-in version would pass these properties and their value as part of a single, anonymous bundle to a script. The script would need to determine the arguments and whether each argument is valid.

## Script-Based Plug-In Architecture

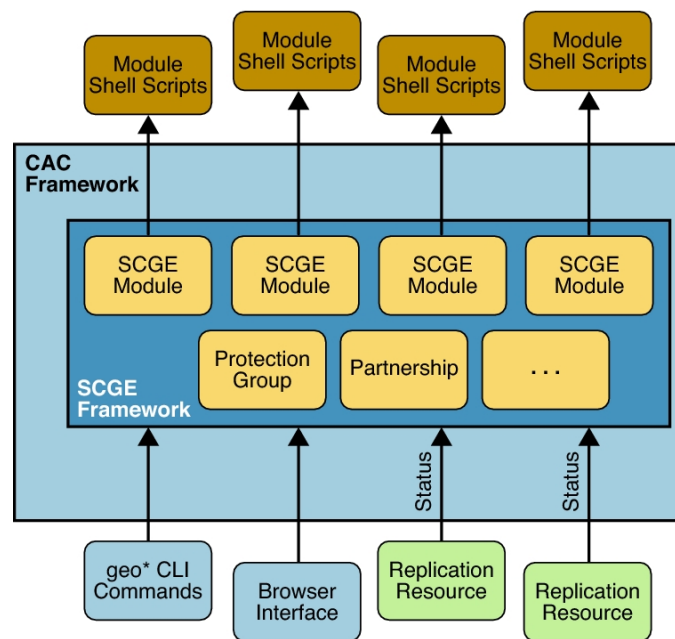
Unlike other replication modules, the script-based plug-in is generic and capable of supporting a wide range of replication technologies. Consequently, the script-based plug-in does not contain a specific set of scripts to control a particular piece of replication software. Instead, it provides a framework for integrating a set of scripts or programs that you, the developer, write and that a system administrator will later use.

This flexibility means that the script-based plug-in cannot directly enforce the inclusion or exclusion of application resource groups in a protection group. Furthermore, the script-based plug-in cannot even restrict the node lists of these entities, nor the relationship with the replication resource group that contains the replication resource needed to supply the replication status, or indeed any other resource group that is required.

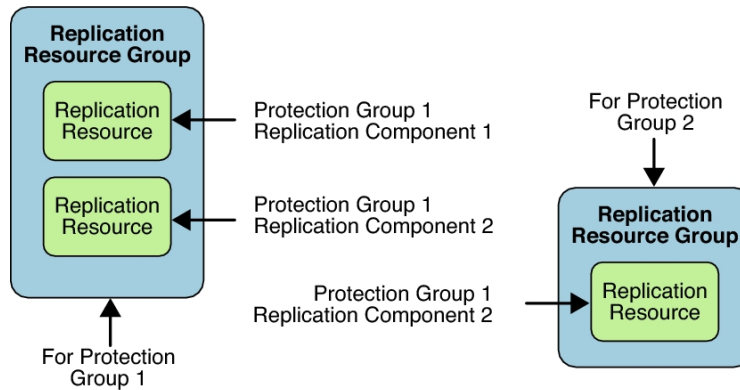
The following figure outlines the relationships between the various components within the disaster recovery framework system. Commands issued through the command-line interface

(CLI) or the Oracle Solaris Cluster Manager browser interface call the disaster recovery framework modules through their relevant common agent container modules. These modules then call out to shell scripts to perform specific tasks. Once a protection group has been instantiated, the replication resource, representing a particular replicated object entity, reports its status back to the module through the event framework. This process enables the overall replication status to be reflected in the disaster recovery framework output on the command line or in the browser interface.

**FIGURE 2** Script-Based Plug-Ins Framework



The script-based plug-in developer therefore is free to govern the relationships between any or all of these entities: application resource group, data replication resource group, and replication status resource group. As the following figure shows, the only constraints are the requirements to have a named replication resource per protection group and a named replication resource per device group or replicated component.

**FIGURE 3** Script-Based Plug-In Replication Resource Group

The consequence of these requirements is that the administrator must provide script-based plug-in configuration file for each protection group that is accessible from all cluster nodes and that details which nodes pertain to each script-based plug-in configuration. The purpose of this configuration file is to ensure that any subsequent developer-written scripts are called on one or more nodes on which the service is present.

In addition to the standard protection properties, the script-based plug-in enables the developer to name one or more scripts to perform the actions required by the disaster recovery framework. These actions fall into two separate groups: those actions that operate at a per protection level and those actions that operate at a per replicated component level.

## Restrictions of Script-Based Plug-Ins

There are no inherent restrictions regarding what you can do when creating script-based plug-in modules. However, using the script-based plug-in does not enable you to circumvent or overcome any inherent limitations present in the replication technology you intend to use.

## Creating Script-Based Plug-Ins

The scripts can be written using an integrated development environment (IDE), such as the NetBeans IDE. For more information on NetBeans IDE, see [NetBeans IDE \(https://netbeans.org/\)](https://netbeans.org/).

## Property Descriptions for Script-Based Plug-Ins

This section contains the following information:

- [“Protection Group Properties - Overview” on page 149](#)
- [“Replicated Component Properties - Overview” on page 150](#)
- [“Protection Group Property Descriptions” on page 151](#)

### Protection Group Properties - Overview

The table in this section lists the protection group properties, along with a brief description, type of property, and default value for each property.

The scripts named by the developer in these properties can reference independent executables, a single common executable, or a combination of the two. No restrictions are placed on the language used to implement these scripts with the exception that the scripts must be able to run by root, from the command line, without a graphical display, and they must return either a zero (success) or nonzero (failure) exit code. The script-based plug-in Mbean returns any error code resulting from a failure. For more information, see [Appendix F, “Error Return Codes for Script-Based Plug-Ins”](#).

Protection groups that use script-based plug-in replication have the global properties provided in the following table. Note that all of these properties are tunable when you are offline.

**TABLE 5** Protection Group Global Policies

Property Name	Description	Type	Default Value
add_app_rg_args	The arguments that are provided to the script, <code>add_app_rg_script</code> .	Optional	Not applicable
add_app_rg_script	The script used to validate and perform tasks relevant for adding an application resource group to a protection group.	Required	/bin/true
configuration_file	The per protection group script-based plug-in configuration file containing details of the nodes pertinent to script-based plug-in replicated components held in the protection group.	Required	/etc/opt/SUNWscgrepsbp/configuration
create_config_script	The script used to create, modify, and validate	Required	/bin/false

	a script-based plug-in replicated component instance.		
remove_app_rg_args	The arguments that are provided to the script, remove_app_rg_script.	Optional	Not applicable
remove_app_rg_script	The script used to validate and perform tasks relevant for removing an application resource group from a protection group.	Required	/bin/true
remove_config_script	The script used to remove a script-based plug-in replicated component instance.	Required	/bin/true
start_replication_script	The script used to start the data replication for a script-based plug-in replicated component instance.	Required	/bin/true
stop_replication_script	The script used to stop the data replication for a script-based plug-in replicated component instance.	Required	/bin/true
switchover_script	The script used to switch over the data replication direction for a script-based plug-in replicated component instance.	Required	/bin/true
takeover_script	The script used to take over the data replication for a script-based plug-in replicated component instance.	Required	/bin/true

The [“Protection Group Property Descriptions” on page 151](#) section describes in detail the actions that each script and its associated arguments should perform when called by the script-based plug-in Mbean. [“Standardized Script Command-Line Arguments” on page 161](#) explains how scripts can discriminate between the steps being performed.

## Replicated Component Properties - Overview

Each replication component added to a particular protection group uses the scripts named in [“Protection Group Properties - Overview” on page 149](#). Individual replications distinguish themselves by varying the properties passed to these scripts.

The script-based plug-in module provides for two site-specific password properties:

- A local service password property (`local_service_password`)
- A remote service password property (`remote_service_password`)

These properties enable administrators of a script-based plug-in deployment to supply passwords to log in to services or remote systems without having to provide these passwords at switchover or takeover time. For more information, see [“How the Disaster Recovery Framework Handles Password Properties”](#) on page 163.

The script-based plug-in module requires the developer to provide a property naming the replication resource contained in the replication resource group that holds the status of the replication.

Replicated components in script-based plug-in protection groups have the optional properties provided in the following table. Note that all of these properties are tunable when you are offline.

**TABLE 6** Optional Replicated Component Properties

Property Name	Description	Type
<code>create_config_args</code>	The arguments passed to the script named by the <code>create_config_script</code> protection group property.	Global
<code>local_service_password</code>	A password that might be needed by the scripts to perform some function on the local system that requires the entry of a password.	Local
<code>remote_service_password</code>	A password that might be needed by the scripts to perform some function on the remote system that requires the entry of a password.	Local
<code>remove_config_args</code>	The arguments passed to the script named by the <code>remove_config_script</code> protection group property.	Global
<code>start_replication_args</code>	The arguments passed to the script named by the <code>start_replication_script</code> protection group property.	Global
<code>stop_replication_args</code>	The arguments passed to the script named by the <code>stop_replication_script</code> protection group property.	Global
<code>switchover_args</code>	The arguments passed to the script named by the <code>switchover_script</code> protection group property.	Global
<code>takeover_args</code>	The arguments passed to the script named by the <code>takeover_script</code> protection group property.	Global

## Protection Group Property Descriptions

This section describes the following protection group properties:

- [“add\\_app\\_rg\\_script Property”](#) on page 152

- “[configuration\\_file Property](#)” on page 153
- “[create\\_config\\_script Property](#)” on page 154
- “[remove\\_app\\_rg\\_script Property](#)” on page 156
- “[remove\\_config\\_script Property](#)” on page 156
- “[start\\_replication\\_script Property](#)” on page 157
- “[stop\\_replication\\_script Property](#)” on page 158
- “[switchover\\_script Property](#)” on page 158
- “[takeover\\_script Property](#)” on page 159

### **add\_app\_rg\_script Property**

The script referenced by the `add_app_rg_script` property is responsible for checking that one or more application resource groups selected by the administrator are suitable for addition to the protection group. These checks might require that certain resource types be present or absent. Furthermore, the script must also set up any resource group affinities or dependencies within the confines of what is allowed by the disaster recovery framework. These affinities or dependencies are needed for the application resource group to produce the correct behavior.

Application resource groups must be in the unmanaged state when they are added to the configuration.

The `add_app_rg_script` is called at other points within the protection group life cycle, not just on the addition of application resource groups, to ensure that application resource groups continue to conform to the required rules. The script should be written to ensure that these rules are met at all times.

Resource groups are offline and unmanaged on the standby site so certain application resource groups that represent services with embedded data replication might be unsuitable for addition to the protection group directly. An example is database data replication such as MySQL and Oracle RAC. The `add_app_rg_script` script must accommodate such validation.

The script must also be able to validate the `add_app_rg_args` property supplied to it with the `validate_parameters=true` option without actually performing any of the steps associated with this task. This operation is called only at the time of protection group update and creation, as opposed to at the time of device group update, modification, or validation.

When executed with `validate_parameters=false`, the script must perform any task required to add the resource groups listed in the final comma-separated `rgList` parameter. These actions might include altering one or more of these resource group properties. The script is called on the local cluster to where the `geopg add-resource-group` command is run and called asynchronously on the remote cluster in response to the internal application resource group table being updated.



For example, if `add_app_rg_script = /var/tmp/addRGs` and `add_app_rg_args = -u root -d /mydir`, the resulting command looks like the following example:

```
# /var/tmp/addRGs -u root -d /mydir function=add_application_rgs \
validate_parameters={true|false} \
currentRole={PRIMARY|SECONDARY} pg=protection-group \
rgList=resource-group1,resource-group2,resource-group3,...
```

where the `rgList` parameter is the comma-separated list of application resource groups that the administrator has opted to add. The script is not responsible for creating these resource groups. Instead, the resource groups must already exist on both clusters. Furthermore, these resource groups must have the `auto_start_on_new_cluster` property set to `false`.

The function name for this step is `add_application_rgs`.

## configuration\_file Property

The `configuration_file` property specifies the file name of the configuration file used to drive the execution of replicated component-level scripts described in [“Plug-In Script Functional Requirements” on page 160](#). Because individual script-based plug-ins inside a protection group might be on disjoint node sets or individual nodes, you should call the user scripts only on the appropriate cluster node or nodes. For more information, see [“Plug-In Script Functional Requirements” on page 160](#).

The configuration file must exist on all cluster nodes on both the primary and standby clusters. The script-based plug-in module tries to read the file from each node in turn until it finds a readable copy, but makes no effort to determine whether all copies are identical.

The format of the configuration file is as follows:

```
script-based-plug-in-configuration-name | nodes-that-must-succeed-running-script | comma-separated-node-list
```

For example:

```
example2.com | any | phys-node1, phys-node2
example1.com | all | phys-node1, phys-node3
baz.com | any | phys-node4
boo | any | phys-node4
biff | all | phys-node2
```

The script-based plug-in configuration name field must match the name of the replicated component being added to the protection group through the `geopg add-device-group` command.

For example2.com, a particular function step is tried on phys-node1 and then, if it fails on phys-node2. The function step can succeed on either node. This configuration assumes that the service is a multinode service like Oracle RAC.

For example1.com, a particular function step must succeed on both phys-node1 and phys-node3 for the step to complete. Again, this configuration is only relevant to multinode services like Oracle RAC. This function step enables a script to perform a task on multiple nodes without needing to connect to a remote node using rsh or ssh between the nodes.

## **create\_config\_script Property**

The script referenced by the `create_config_script` property is responsible for creating, modifying, and validating a script-based plug-in configuration. The script must be able to validate the `create_config_args` property supplied to it with the `validate_parameters=true` option without actually performing the configuration creation.

When executed with `validate_parameters=false`, the script must create a replication group and an associated replication resource for the particular script-based plug-in. There must be only one replication resource group per script-based plug-in protection group and only one replication resource per replicated component. For example, a configuration with two script-based plug-in protection groups (`hr-pg` and `sales-pg`), each with two replicated components (`hr-west` and `hr-east` for `hr-pg`, and `sales-north` and `sales-south` for `sales-pg`), would have two resource groups (`hr_pg_rep-rg` and `sales_pg_rep-rg`). These resource groups would then have the following two resources:

- `hr_west-rep-rs` and `hr_east-rep-rs` in `hr_pg_rep-rg`
- `sales_north-rep-rs` and `sales_south-rep-rs` in `hr_pg_rep-rg`

When creating the second replicated component or validating either configuration, the script must handle the case where the resource group already exists.

On completion, the script must write the resource group name and resource to standard output. This task is checked by the script-based plug-in framework to both validate that the objects exist and to set up the appropriate notification handling for state change events. The format for the output is as follows:

```
reprg=replication-resource-group-name  
reprs=replication-resource-name
```

For example, for the case where the replication resource group is called `hr_pg_rep-rg` and the replication resource is called `hr_west-rep-rs`, the output would be as follows:

```
reprg=hr_pg_rep-rg  
reprs=hr_west-rep-rs
```

The script must also write a list of resource groups to standard output that it has either created, or that exist already, or that it considers internal to the protection group. The format of the output must be as follows, with a carriage return at the end of the line:

```
rglist=comma-separated-list-of-resource-groups
```

For example, for the case where `foo-rg` and `bar-rg` are internal, the output would be as follows:

```
rglist=foo-rg,bar-rg
```

If no resource groups exist, the output would be as follows:

```
rglist=
```

An example of such internal resource groups is the shadow RAC proxy server resource groups in the Oracle Data Guard module.

This script is called for each script-based plug-in created in any specific protection group because `create_config_script` is a global protection group property. For example, if a protection group has script-based plug-in configurations `foobar.com` and `baz.com`, the `create_config_script` script is called once when `foobar.com` is added with the `create_config_args` property given for the `foobar.com` property. The script is later called for `baz.com` when it is added to the protection group with the `baz.com` `create_config_args` property value. This process results in a replication resource group with two resources: one resource monitoring `foobar.com` replication and the other resource monitoring `baz.com`.

If the protection group is known to both the primary and standby sites, then adding the script-based plug-in configuration to the protection group will cause the `create_config_script` script to be executed on the site that the `geopg` command is run from and then on the remote site as a result of the internal Oracle Solaris Cluster Geographic protection group table transfer. The latter step happens asynchronously.

The `create_config_script` script is called with the `create_config_args` property followed by the standard command-line arguments and an additional `isModify` parameter. This parameter is set to `false` when the command has been called as a result of a `geopg create-device-group` or `geopg validate pgcommand`. This parameter is set to `true` when the command has been called as a result of a `geopg modify-device-group` command.

For example, if `create_config_script = /var/tmp/add` and `create_config_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
/var/tmp/add -u root -d /mydir function=create_configuration \  
validate_parameters={true|false} currentRole={PRIMARY|SECONDARY} \  
pg=protection-group isModify={true|false}
```

The function name for this step is `create_configuration`.

### **remove\_app\_rg\_script Property**

The script referenced by the `remove_app_rg_script` property is responsible for removing one or more application resource groups, selected by the administrator, from the protection group. A comma-separated list of resource groups to remove is passed to the script through the `rgList` parameter. The script is called on the local cluster to where the `geopg remove-resource-group` command is run and called asynchronously on the remote cluster in response to the internal application resource group table being updated.

The script must also be able to validate the `remove_app_rg_args` property supplied to it with the `validate_parameters=true` option without actually performing any of the steps associated with this task. This operation is called only at the time of protection group update and creation, as opposed to at the time of device group update, modification, or validation.

For example, if `remove_app_rg_script = /var/tmp/removeRGs` and `remove_app_rg_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
#!/var/tmp/removeRGs -u root -d /mydir\  
function=remove_application_rgs \  
validate_parameters={true|false} \  
currentRole={PRIMARY|SECONDARY} pg=protection-group\  
rgList=resource-group1,resource-group2,resource-group3,...
```

where the `rgList` parameter is the comma-separated list of application resource groups that the administrator has opted to remove. The script is not responsible for removing these resource groups, only for making the necessary changes to their properties that might be required as a result of removing them from disaster recovery framework protection group control.

The function name for this step is `remove_application_rgs`.

### **remove\_config\_script Property**

The script referenced by the `remove_config_script` property is responsible for reversing the work of the `create_config_script` script. The script must be able to validate the `remove_config_args` property supplied to it with the `validate_parameters=true` option without actually performing the configuration removal.

When executed with `validate_parameters=false`, the script must remove the replication resource (originally named by the `create_config_script` script `reprs=` output for the specific

script-based plug-in) from the replication resource group given by the `create_config_script` script `reprg=` output. If the resource is the last in the resource group, the script must also remove the resource group.

For example, if `remove_config_script = /var/tmp/remove` and `remove_config_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
# /var/tmp/remove -u root -d /mydir function=remove_configuration \
validate_parameters={true|false} \
currentRole={PRIMARY|SECONDARY} pg=protection-group
```

The function name for this step is `remove_configuration`.

## start\_replication\_script Property

The script referenced by the `start_replication_script` property is responsible for starting the data replication process and enabling the replication resource that is used to monitor the replication. The script must also be able to validate the `start_replication_args` property supplied to it with the `validate_parameters=true` option without actually starting the data replication.

When executed with `validate_parameters=false`, the script must start the actual data replication and enable the replication resource that is used to monitor the replication.

For example, if `start_replication_script = /var/tmp/start` and `start_replication_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
# /var/tmp/start -u root -d /mydir function=start_replication \
validate_parameters={true|false} \
currentRole={PRIMARY|SECONDARY} pg=protection-group
```

The `start_replication_script` script is called on one or both clusters depending on which of the following commands the administrator specifies:

For local clusters only:

```
# geopg start -e local protection-group      # local cluster only
```

For both clusters:

```
# geopg start -e global protection-group     # both clusters
```

The function name for this step is `start_replication`.

## stop\_replication\_script Property

The script referenced by the `stop_replication_script` property is responsible for stopping the data replication process and disabling the replication resource that is used to monitor the replication. The script must also be able to validate the `stop_replication_args` property supplied to it with the `validate_parameters=true` option without actually starting the data replication.

When executed with `validate_parameters=false`, the script must stop the actual data replication and disable the replication resource that is used to monitor the replication.

For example, if `stop_replication_script = /var/tmp/stop` and `stop_replication_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
# /var/tmp/stop -u root -d /mydir function=start_replication \  
validate_parameters={true|false} \  
currentRole={PRIMARY|SECONDARY} pg=protection-group
```

The `stop_replication_script` script is called on one or both clusters depending on which of the following commands the administrator specifies:

For local cluster only:

```
# geopg stop -e local protection-group # local cluster only
```

For both clusters:

```
# geopg stop -e global protection-group # both clusters
```

The function name for this step is `stop_replication`.

## switchover\_script Property

The script referenced by the `switchover_script` property is responsible for two functions:

- Checking that the service is in a position to switch over
- Performing the actual data replication switchover

The second step is only performed if the first step is completed successfully, meaning that the step exits with a zero exit code. In each case, the script is called on both clusters.

The `switchover_script` script is first called on the cluster on which the `geopg switchovercommand` is executed. Subsequent changes in the disaster recovery framework

status trigger an event on the remote cluster, causing the script to be executed asynchronously on that cluster, too. The arguments for the two calls are different.

For example:

If `switchover_script = /var/tmp/switchover` and `switchover_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
# /var/tmp/switchover -u root -d /mydir function=check_switchover \
validate_parameters=false currentRole={PRIMARY|SECONDARY} \
pg=protection-group newRole={PRIMARY|SECONDARY}
```

If that step succeeds:

```
# /var/tmp/switchover -u root -d /mydir \
function=perform_switchover \
validate_parameters=false \
currentRole={PRIMARY|SECONDARY} pg=protection-group \
newRole={PRIMARY|SECONDARY}
```

The argument `newRole` is the target role of the cluster after a successful switchover.

The function names for these steps are `check_switchover` and `perform_switchover` and just `switchover` for the `validate_parameter` step, which is called as follows:

```
# developer-switchover-program developer-switchover-program-arguments \
function=switchover validate_parameters=true \
currentRole={PRIMARY|SECONDARY} pg=protection-group
```

## takeover\_script Property

The script referenced by the `takeover_script` property is responsible for two functions:

- Checking that the service is in a position to be taken over
- Performing the actual data replication takeover

The second step is only performed if the first step is completed successfully, meaning that the step exits with a zero exit code. In each case, the script is called on both clusters. If the original primary cluster is available, the protection group is deactivated on that cluster. Deactivation involves stopping the application resource groups.

The `takeover_script` script must be called on the standby cluster by executing the `geopg takeover` command on that cluster. The arguments for the two calls are different.

For example, if `takeover_script = /var/tmp/switchover` and `takeover_args = "-u root -d /mydir"`, the resulting command looks like the following example:

```
# /var/tmp/switchover -u root -d /mydir function=check_takeover \  
validate_parameters=false currentRole={PRIMARY|SECONDARY} \  
pg=protection-group newRole={PRIMARY|SECONDARY}
```

Then, if that step succeeds:

```
# /var/tmp/switchover -u root -d /mydir function=perform_takeover \  
validate_parameters=false currentRole={PRIMARY|SECONDARY} \  
pg=protection-group newRole={PRIMARY|SECONDARY}
```

The argument `newRole` is the target role of the cluster after a successful takeover.

The function names for these steps are `check_takeover` and `perform_takeover` and just `takeover` for the `validate_parameter` step, which is called as follows:

```
# developer-takeover-program developer-takeover-program-arguments \  
function=takeover validate_parameters=true \  
currentRole={PRIMARY|SECONDARY} pg=protection-group
```

## Internals for Script-Based Plug-Ins

This section describes the internals for the script-based plug-ins. It covers the following topics:

- [“Plug-In Script Functional Requirements” on page 160](#)
- [“Script-Based Plug-In Replication Resource Groups and Resources” on page 162](#)
- [“Protection Group Status Mapped from Replication Resource Status” on page 163](#)
- [“How the Disaster Recovery Framework Handles Password Properties” on page 163](#)

## Plug-In Script Functional Requirements

A protection group has several global properties that are valid and relevant to both the primary and secondary clusters, and by extension all cluster nodes. Additionally, each replicated component has a set of local and global properties. Together, these properties describe and control the replication pertaining to one or more replicated services.

This section describes the following topics:

- [“Plug-In Script Argument Validation” on page 161](#)
- [“Standardized Script Command-Line Arguments” on page 161](#)



## Plug-In Script Argument Validation

Each script provided in one of the protection group properties must be capable of validating the arguments with which it has been called in order to determine whether the arguments are complete and acceptable. Validation ensures that scripts such as `switchover_script` and `takeover_script`, that are not called regularly, do not fail because their arguments have become incompatible. Failing to validate the arguments could lead to the inability to switch over or take over in an emergency.

Scripts must therefore be able to validate the arguments defined by the administrator through the Oracle Solaris Cluster Manager browser interface or command-line interface (CLI), and issue a return code of zero, if they are correct. The script must not perform its real function at this stage, for example, to switch over, take over, or create a script-based plug-in configuration. If you do not want to perform these checks, the script must still return without performing any additional work in response to the `validate arguments` call.

The `validate arguments` step is denoted by the disaster recovery framework script-based plug-in Mbean passing `validate_parameters=true` as one of the command-line arguments. When a script-based plug-in replication component is added to a protection group, all the replicated component-specific scripts listed in [“Protection Group Properties - Overview” on page 149](#) are called on to validate their arguments. This call is made on one or more nodes per cluster depending on the particular script-based plug-in replicated component configuration as defined in the configuration file. For more information, see [“configuration\\_file Property” on page 153](#) and [“Protection Group Properties - Overview” on page 149](#).

The same validation calls are made under the following circumstances:

- When the replication component is modified because the modification might result in program argument changes
- When there are protection group validation calls in response to the `geopg validate protection-group` command
- When the disaster recovery framework is starting and recreating the initial script-based plug-in replicated component objects that are stored in the Cluster Configuration Repository (CCR)

There are also two protection group level program properties, `add_app_rg_script` and `remove_app_rg_script`, that have associated protection group argument properties.

## Standardized Script Command-Line Arguments

All scripts are called using a standardized command-line structure. The format of the command line is as follows:

```
# developer-program-name administrator-supplied-program-arguments \  
function=step-name \  
validate_parameters={true|false} \  
currentRole={PRIMARY|SECONDARY} \  
pg=protection-group \  
additional-function-dependent-arguments
```

where *developer-program-name* is the name of one of the externally developed scripts and *administrator-supplied-program-arguments* provides the arguments given for this script by the administrator when setting up a script-based plug-in configuration.

The use of the `function=step-name` argument enables scripts to determine what action they are being called on to perform. This function is especially important if a single script has been written to perform one or more tasks. Two scripts in particular need to be concerned with this argument: `switchover_script` and `takeover_script`.

The `currentRole` argument indicates the current role of the local cluster, while the `pg` argument denotes the name of the protection group containing the script-based plug-in configuration. Scripts should be prepared to deal with values in either uppercase or lowercase. The same is true of the `newRole` argument for `switchover_script` and `takeover_script`.

All scripts, if successful, must return a zero exit code. On failure, all scripts must return a nonzero exit code and generate a localized error message on standard error (`stderr`). Any output sent to standard output (`stdout`) is generally ignored (with the exception of `create_config_script`), unless common agent container logging is turned on. In that case, the output is saved in the `/var/cacao/instances/default/logs/cacao.0` log file, along with other common agent container debugging information. Do not save debugging information as a matter of course because the volume of output can be substantial.

## Script-Based Plug-In Replication Resource Groups and Resources

The name of the replication resource group for a particular protection group is defined by the value returned by `create_config_script` in the `reprg=` string sent to standard output. This string contains one or more replication resources referenced by individual replication resources named by `create_config_script` in the `reprs=` string sent to standard output. For any one protection group, the value returned by `create_config_script` must be identical.

The function of the replication resources is to monitor the state of the replication associated with the resource and thus the replicated component. The replication resource status, which is set by a probe method, is used to determine the overall status of the protection group. The start and stop methods of the replication resource do not start and stop the actual data replication.

The replication resource must be enabled and disabled by `start_replication_script` and `stop_replication_script`.

## Protection Group Status Mapped from Replication Resource Status

The protection group status reflects the aggregated status of all replication resources in the replication resource group created by the developer-written `create_config_script` program.

The following table illustrates the mapping from the status of each replication resource to the protection group status. An X represents any possible status for the resource and demonstrates that the most restrictive status governs the overall status of the protection group.

Unknown	Faulted	Degraded	Online	Protection Group Status
True	X	X	X	UNKNOWN
False	True	X	X	FAULTED
False	False	True	X	DEGRADED
False	False	False	True	ONLINE

## How the Disaster Recovery Framework Handles Password Properties

This section describes the mechanism by which the disaster recovery framework handles password properties, when the entity added to a protection group (for example, an Oracle Data Guard or script-based plug-in configuration) requires a password property.

The password properties are read during the execution of the `geopg` command. These password properties are recognized by their conformance to the pattern `*_password`. When `geopgi` (a back-end program called by the `geopg` command) parses the protection group properties list, it looks for such arguments. If the password has been supplied in cleartext, as shown in the following example, then `geopg` warns the user that the password is insecure, but continues processing the password.

```
... -p sysdba_password=password ...
```

For any password properties that have been specified, the `geopgi` program enters non-echo mode and prompts for these passwords, as shown in the following example:

```
... -p local_service_password= -p remote_service_password= ...
```

Once all the arguments have been processed, these pairs are written into an internal password file on the local node, which is root readable only. A separate `internalPasswordField` argument is inserted into the properties list with the value *hostname:filename*.

Once in the core disaster recovery framework Java code, the `internalPasswordField` argument is unpacked, and the file is read remotely through an internal common agent container to common agent container call. For security, the passwords are then converted into the hexadecimal representation of their character codes before they are written to the Oracle Solaris Cluster CCR, if the rest of the properties are correct and complete, and the validation succeeds.

When required, the passwords can be queried and converted back from the CCR and supplied to the appropriate programs to achieve the relevant switchovers, takeovers, or status queries.

## Standard Disaster Recovery Framework Properties

---

This appendix provides the standard properties of disaster recovery framework heartbeats, heartbeat plug-in, partnerships, protection groups and data replication device groups, sites, and multigroups.

This appendix contains the following sections:

- [“Heartbeat Properties” on page 165](#)
- [“Heartbeat Plug-in Properties” on page 166](#)
- [“Partnership Properties” on page 167](#)
- [“Protection Group Properties” on page 168](#)
- [“Site Properties” on page 169](#)
- [“Multigroup Properties” on page 171](#)

---

**Note** - The property names and values, such as `Query_interval`, `True`, and `False`, are *not* case sensitive.

---

## Heartbeat Properties

The following table describes the heartbeat properties that the disaster recovery framework defines.

**TABLE 7** Disaster Recovery Framework Heartbeat Properties

Property Name	Description
<code>Query_interval</code> (integer)	Specifies the delay in seconds between heartbeat status requests.  <b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.

Property Name	Description
	<b>Category:</b> Optional  <b>Default:</b> 120 seconds

## Heartbeat Plug-in Properties

The following table describes the heartbeat plug-in properties that the disaster recovery framework defines.

**TABLE 8** Disaster Recovery Framework Heartbeat Plug-in Properties

Property	Description
Plugin_properties (string)	<p>Specifies a property string specific to the plug-in.</p> <p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> None except for heartbeats that use the default heartbeat plug-ins, tcp_udp_plugin and ping_plugin.</p> <p>For the tcp_udp_plugin plug-in, the format of this string is predefined as <i>remote-IP-address/UDP/2084/ipsec</i>, <i>remote-IP-address/TCP/2084/ipsec</i>. The <i>remote-IP-address</i> argument specifies the IP address of the partner cluster. The optional <i>ipsec</i> argument specifies if the plug-in uses IPsec with a Boolean value of <i>true</i> or <i>false</i>.</p> <p>For the ping_plugin, the format of this string is predefined as <i>remote-IP-address</i>, where <i>remote-IP-address</i> specifies the IP address of the partner cluster.</p>
Query_cmd (string)	<p>Specifies the path to the heartbeat status request command.</p> <p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.</p> <p><b>Category:</b> Required property if the plug-in does not specify a predefined plug-in.</p> <p><b>Default:</b> None</p>
Requester_agent (string)	<p>Specifies the absolute path to the requester agent.</p> <p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime. However, the Requester_agent property of the default plug-in should never need to be tuned except for testing purposes.</p> <p><b>Category:</b> Optional</p>

Property	Description
	<b>Default:</b> None
Responder_agent (string)	<p>Specifies the absolute path to the responder agent.</p> <p><b>Tuning recommendations:</b> The value is assigned at creation and can be tuned at runtime. However, the Responder_agent property of the default plug-in should never need to be tuned except for testing purposes.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> None</p>
Type (enum)	<p>Designates the type of plug-in. Set to either primary or backup.</p> <p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.</p> <p><b>Category:</b> Required</p> <p><b>Default:</b> None, except for the default heartbeat that is named ping_plugin. If using this plug-in, the default value is backup.</p>

## Partnership Properties

The following table describes the partnership properties that the disaster recovery framework defines.

**TABLE 9** Disaster Recovery Framework Partnership Properties

Property	Description
Description (string)	<p>Describes the partnership.</p> <p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> Empty string</p>
Notification_ActionCmd (string)	<p>Provides the path to the action script that is triggered when heartbeat-loss notification is issued.</p> <p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> Empty string</p>
Notification_EmailAdrs (string array)	<p>Lists the email addresses that are sent email when heartbeat-loss notification is issued. The list is comma delimited.</p>

Property	Description
	<p><b>Tuning recommendations:</b> The value of this property is assigned at creation and can be tuned at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> Empty string</p>

## Protection Group Properties

The following table describes the protection group properties that the disaster recovery framework defines.

**TABLE 10** Disaster Recovery Framework Protection Group Properties

Property	Description
Description (string)	<p>Describes the protection group.</p> <p><b>Tuning recommendations:</b> This property can be tuned at any time.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> Empty string</p>
External_Dependency_Allowed (Boolean)	<p>Allow dependencies between resource groups and resources that belong to this protection group and resource groups and resources that do not belong to this protection group when set to <code>true</code>.</p> <p><b>Tuning recommendations:</b> This property can be tuned at any time.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> <code>false</code></p>
RoleChange_ActionArgs (string)	<p>Defines a string of arguments that are appended to the end of the command line when the role-change action command, <code>RoleChange_ActionCmd</code>, is run.</p> <p><b>Tuning recommendations:</b> This property can be tuned at any time.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> Empty string</p>
RoleChange_ActionCmd (string)	<p>Specifies the path to an executable command. This script is invoked during a switchover or takeover on the new primary cluster when the protection group is started on the new primary cluster. The script is invoked on the new primary cluster after the data replication role changes from secondary to primary and before the application resource groups are brought online. If the data replication role change does not succeed, then the script is not called.</p>



Property	Description
	<p>This file should be valid on all nodes of all partner clusters that can host the protection group, and have execute permissions for the user that launches the script.</p> <p><b>Tuning recommendations:</b> This property can be tuned at any time.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> Empty string</p>
Timeout (integer)	<p>Specifies the timeout period for the protection group in seconds. The timeout period is the longest time disaster recovery framework waits for a response after you run a geopg command, such as <code>geopg start</code>, <code>geopg stop</code>, <code>geopg switchover</code>, and <code>geopg takeover</code>. If the command does not respond within the timeout period, the disaster recovery framework reports the operation as timed out, even if the underlying command eventually completes successfully.</p> <p>You should identify the amount of time required to perform a role-reversal of the data replication, and set the timeout value to 150% to 200% of that value to ensure enough time for the role-reversal to complete.</p> <p>To ensure that an operation has finished on the remote cluster, check system status after a timeout before attempting the operation again. For more information, see <a href="#">“Troubleshooting Migration Problems” on page 182</a>.</p> <p>The timeout period applies to operations on a per-cluster basis. An operation with a local scope times out if the operation does not complete after the specified timeout period.</p> <p>An operation with a global scope consists of an action on the local cluster and an action on the remote cluster. The local and remote action are timed separately so that an operation with a global scope times out during one of the following conditions:</p> <ul style="list-style-type: none"> <li>■ The local operation does not complete after the specified timeout period.</li> <li>■ The remote operation does not complete after the specified timeout period.</li> </ul> <p><b>Tuning recommendations:</b> This property can be tuned only when the protection group is offline.</p> <p><b>Category:</b> Optional</p> <p><b>Range:</b> 20-1000000 seconds</p> <p><b>Default:</b> 200</p>

## Site Properties

The following table describes the site properties that the disaster recovery framework defines.

**TABLE 11** Disaster Recovery Framework Site Properties

Property	Description
Description (string)	<p>Describes the site. The system sets this property on the local cluster, then propagates the value to the other clusters in the site.</p> <p><b>Tuning recommendations:</b> This property is assigned at creation and tunable at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> None</p>
joinTimeout (integer)	<p>Specifies, in seconds, the longest time that the JMX client, the site, waits for the invocation of an MBean-server method to return. If a method does not return by the end of the timeout period, the client moves to its next set of instructions and reports the operation as timed out. By default, a client waits indefinitely for a method to return. If the MBean server is unable to complete an invocation, the JMX client will hang indefinitely. The timeout period is site-wide and applies to operations on a per-cluster basis.</p> <p>The joinTimeout property is used when the join operation requires more time to complete than the default setting allows before the geosite joincommand times out.</p> <p><b>Tuning recommendations:</b> This property is assigned at creation, but it is not tunable at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> 30 seconds</p> <p><b>Minimum:</b> 20 seconds</p> <p><b>Maximum:</b> 3600 seconds</p>
Query_interval (integer)	<p>Specifies, in seconds, the frequency between heartbeat status requests used by the site clusters. The plug-in enters emergency mode if three Query_interval periods pass without response. The plug-in times out and enters error mode if a further Query_interval period passes with no response.</p> <p><b>Tuning recommendations:</b> This property is assigned at creation and tunable at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> 120 seconds</p> <p><b>Minimum:</b> 20 seconds</p> <p><b>Maximum:</b> 300 seconds</p>
Timeout (integer)	<p>Specifies, in seconds, the longest time that the JMX client, the site, waits for the invocation of an MBean-server method to return. If a method does not return by the end of the timeout period, the client moves to its next set of instructions and reports the operation as timed out. By default, a client waits indefinitely for a method to return. If the MBean server is unable to complete</p>

Property	Description
	<p>an invocation, the JMX client will hang indefinitely. The timeout period is site-wide and applies to operations on a per-cluster basis.</p> <p><b>Tuning recommendations:</b> This property is assigned at creation and tunable at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> 30 seconds</p> <p><b>Minimum:</b> 30 seconds</p> <p><b>Maximum:</b> 3600 seconds</p>

## Multigroup Properties

The following table describes the multigroup properties that the disaster recovery framework defines.

**TABLE 12** Disaster Recovery Framework Multigroup Properties

Property	Description
Description (string)	<p>Describes the multigroup. The system sets this property on the local cluster, then propagates the value to the other clusters in the site where the multigroup is defined.</p> <p><b>Tuning recommendations:</b> This property is assigned at creation and tunable at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> None</p>
Timeout (integer)	<p>Specifies, in seconds, the longest time that the JMX client, the site, waits for the invocation of an MBean-server method to return. If a method does not return by the end of the timeout period, the client moves to its next set of instructions and reports the operation as timed out. By default, a client waits indefinitely for a method to return. If the MBean server is unable to complete an invocation, the JMX client will hang indefinitely. The timeout period is site-wide and applies to operations on a per-cluster basis.</p> <p>The timeout period applies to operations on a per-cluster basis. An operation with a local scope times out if the operation is not completed after the specified Timeout period.</p> <p>An operation with a global scope consists of an action on the local cluster and an action on the remote cluster. The local and remote actions are timed separately. So, an operation with a global scope times out if the local operation</p>

Property	Description
	<p>is not completed after the specified timeout period or if the remote operation is not completed after the specified timeout period.</p> <p>For example, the following command is started with a local scope:</p> <pre data-bbox="748 489 1114 514"># geomg start -e local multigroup</pre> <p>If you set the Timeout property to 3000 seconds, the <code>geomg start</code> command times out if the operation does not complete after 3000 seconds.</p> <p>You can start the same command with a global scope as follows:</p> <pre data-bbox="748 657 1127 682"># geomg start -e global multigroup</pre> <p>If the Timeout property is set to 3000 seconds, the <code>geomg start</code> command times out if the operation is not completed on the local cluster after 3000 seconds or if the operation is not completed on the remote cluster after 3000 seconds. If the local action takes 1500 seconds and the remote action takes 1500 seconds, the operation is not timed out.</p> <p>The multigroup timeout value is an estimated value that is applied to some sub-operations. The timeout value does not apply to the entire operation, so not every operation on a protection group is timed against the timeout period. For example, the time taken to initialize the data structure and to check for the precondition of the operation are not timed in the timeout period.</p> <p><b>Tuning recommendations:</b> This property is assigned at creation and tunable at runtime.</p> <p><b>Category:</b> Optional</p> <p><b>Default:</b> 30 seconds</p> <p><b>Minimum:</b> 20 seconds</p> <p><b>Maximum:</b> 3600 seconds</p>

# ◆◆◆ APPENDIX B

## Legal Names and Values of Disaster Recovery Framework Entities

---

This appendix lists the requirements for legal characters for the names and values of disaster recovery framework entities.

This appendix contains the following sections:

- [“Legal Names for Disaster Recovery Framework Entities” on page 173](#)
- [“Legal Values for Disaster Recovery Framework Entities” on page 174](#)

### Legal Names for Disaster Recovery Framework Entities

disaster recovery framework entity names consist of the following:

- Host names
- Cluster names, which must follow the naming requirements for host names
- Partnership names
- Protection group names
- Custom heartbeat names

All names must comply with the following rules:

- Must start with a letter
- Must not exceed 255 characters
- Can contain the following:
  - Upper and lowercase letters
  - Digits
  - Dashes (-), except as the last character of a host name or cluster name
  - Underscores (\_), except in a host name or cluster name

For more information about host name requirements, see RFC 1123 at <http://www.rfcs.org/>.

## Legal Values for Disaster Recovery Framework Entities

The disaster recovery framework entity values fall into two categories: property values and description values. Both types of values share the following 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 a newline or a semicolon

## Disaster Recovery Administration Example

---

This appendix provides an example of a disaster recovery scenario and the actions an administrator might perform.

Example Company has two geographically separated clusters, `cluster-paris` in Paris, and `cluster-newyork` in New York. These clusters are configured as partner clusters. The cluster in Paris is configured as the primary cluster and the cluster in New York is the secondary.

The `cluster-paris` cluster fails temporarily as a result of power outages during a windstorm. An administrator can expect the following events:

1. The heartbeat communication is lost between `cluster-paris` and `cluster-newyork`. Because heartbeat notification was configured during the creation of the partnership, a heartbeat-loss notification email is sent to the administrator.  
For information about the configuring partnerships and heartbeat notification, see [“Modifying Partnership Properties” on page 41](#).
2. The administrator receives the notification email and follows the company procedure to verify that the disconnect occurred because of a situation that requires a takeover by the secondary cluster. Because a takeover might take a long time, depending on the requirements of the applications being protected, Example Company does not allow takeovers unless the primary cluster cannot be repaired within two hours.  
For information about verifying a disconnect on a system, see [“Detecting Cluster Failure” on page 127](#).
3. Because the `cluster-paris` cluster cannot be brought online again for at least another day, the administrator runs a `geopg takeover` command on a node in the cluster in New York. This command starts the protection group on the secondary cluster `cluster-newyork` in New York.  
For information about performing a takeover on a system, see [“Forcing a Takeover of a Protection Group” on page 133](#).
4. After the takeover, the secondary cluster `cluster-newyork` becomes the new primary cluster. The failed cluster in Paris is still configured to be the primary cluster. Therefore, when the `cluster-paris` cluster restarts, the cluster detects that the primary cluster was

---

down and lost contact with the partner cluster. Then, the `cluster-paris` cluster enters an error state that requires administrative action to clear. You might also be required to recover and resynchronize data on the cluster.

For information about recovering data after a takeover, see the disaster recovery framework guide for your data replication product.



# ◆◆◆ APPENDIX D

## Takeover Postconditions

---

This appendix provides details about the state of the primary and secondary clusters after you run the `geopg takeover` command or the `geomg takeover` command.

This appendix contains the following sections:

- [“Results of a Takeover When the Partner Cluster Can Be Reached” on page 177](#)
- [“Results of a Takeover When the Partner Cluster Cannot Be Reached” on page 178](#)

### Results of a Takeover When the Partner Cluster Can Be Reached

This section describes the activation state of the primary and secondary clusters before and after you run the `geopg takeover` command. The results described in this section assume that the partner cluster can be reached.

The following table describes the states of the clusters when you run the `geopg takeover` command on the secondary cluster, `cluster-newyork`.

**TABLE 13** Takeover Results of Running the `geopg takeover` Command on the Secondary Cluster

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
<code>cluster-paris: primary, deactivated</code>	<code>cluster-paris: secondary, deactivated</code>
<code>cluster-newyork: secondary, deactivated</code>	<code>cluster-newyork: primary, deactivated</code>
<code>cluster-paris: primary, activated</code>	<code>cluster-paris: secondary, deactivated</code>
<code>cluster-newyork: secondary, deactivated</code>	<code>cluster-newyork: primary, deactivated</code>
<code>cluster-paris: primary, deactivated</code>	<code>cluster-paris: secondary, deactivated</code>
<code>cluster-newyork: secondary, activated</code>	

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
	cluster-newyork: primary, activated, with data replication stopped
cluster-paris: primary, activated	cluster-paris: secondary, deactivated
cluster-newyork: secondary, activated	cluster-newyork: primary, activated, with data replication stopped

The following table describes the states when you run the `geopg takeover` command on the primary cluster, `cluster-paris`.

**TABLE 14** Takeover Results of Running the `geopg takeover` Command on the Primary Cluster

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
cluster-paris: primary, deactivated	cluster-paris: primary, deactivated
cluster-newyork: secondary, deactivated	cluster-newyork: secondary, deactivated
cluster-paris: primary, activated	cluster-paris: primary, activated, with data replication stopped
cluster-newyork: secondary, deactivated	cluster-newyork: secondary, deactivated
cluster-paris: primary, deactivated	cluster-paris: primary, deactivated
cluster-newyork: secondary, activated	cluster-newyork: secondary, deactivated
cluster-paris: primary, activated	cluster-paris: primary, activated, with data replication stopped
cluster-newyork: secondary, activated	cluster-newyork: secondary, deactivated

## Results of a Takeover When the Partner Cluster Cannot Be Reached

This section describes the activation state of the primary and secondary clusters before and after you run a `geopg takeover` command when the partner cluster cannot be reached or when the protection group on the partner cluster is busy.

The following table describes the states when you run the `geopg takeover` command on the secondary cluster, `cluster-newyork`, and the primary cluster cannot be reached or the protection group on the primary cluster is busy.

---

**Note** - The cluster role and state after the takeover, which is given in the table, is available only when the partner cluster can be reached again.

---

**TABLE 15** Takeover Results of Running the `geopg takeover` Command on the Secondary Cluster When the Primary Cluster Cannot Be Reached

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
<code>cluster-paris: primary, deactivated, synchronization status Unknown</code>	<code>cluster-paris: primary, deactivated, synchronization status Error</code>
<code>cluster-newyork: secondary, deactivated, synchronization status Unknown</code>	<code>cluster-newyork: primary, deactivated, synchronization status Error</code>
<code>cluster-paris: primary, activated, synchronization status Unknown</code>	<code>cluster-paris: primary, activated, synchronization status Error</code>
<code>cluster-newyork: secondary, deactivated, synchronization status Unknown</code>	<code>cluster-newyork: primary, deactivated, synchronization status Error</code>
<code>cluster-paris: primary, deactivated, synchronization status Unknown</code>	<code>cluster-paris: primary, deactivated, synchronization status Error</code>
<code>cluster-newyork: secondary, activated, synchronization status Unknown</code>	<code>cluster-newyork: primary, activated, with data replication stopped, synchronization status Error</code>
<code>cluster-paris: primary, activated, synchronization status Unknown</code>	<code>cluster-paris: primary, activated, synchronization status Error</code>
<code>cluster-newyork: secondary, activated, synchronization status Unknown</code>	<code>cluster-newyork: primary, activated, with data replication stopped, synchronization status Error</code>

The following table describes the states when you run the `geopg takeover` command on the primary cluster, `cluster-paris`, and the secondary cluster cannot be reached or the protection group on the secondary cluster is busy.

**TABLE 16** Takeover Results of Running the `geopg takeover` Command on the Primary Cluster When the Secondary Cluster Cannot Be Reached

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
<code>cluster-paris: primary, deactivated, synchronization status Unknown</code>	<code>cluster-paris: primary, deactivated, synchronization status OK, Error, or Mismatch</code>
<code>cluster-newyork: secondary, deactivated, synchronization status Unknown</code>	<code>cluster-newyork: secondary, deactivated, synchronization status OK, Error, or Mismatch</code>
<code>cluster-paris: primary, activated, synchronization status Unknown</code>	<code>cluster-paris: primary, activated, with data replication stopped, synchronization status OK, Error, or Mismatch</code>
<code>cluster-newyork: secondary, deactivated, synchronization status Unknown</code>	<code>cluster-newyork: secondary, deactivated, synchronization status OK, Error, or Mismatch</code>
<code>cluster-paris: primary, deactivated, synchronization status Unknown</code>	<code>cluster-paris: primary, deactivated, synchronization status OK, Error, or Mismatch</code>
<code>cluster-newyork: secondary, activated, synchronization status Unknown</code>	<code>cluster-newyork: secondary, activated, synchronization status OK, Error, or Mismatch</code>

## Results of a Takeover When the Partner Cluster Cannot Be Reached

---

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
cluster-paris: primary, activated, synchronization status Unknown	cluster-paris: primary, activated, with data replication stopped, synchronization status OK, Error, or Mismatch
cluster-newyork: secondary, activated, synchronization status Unknown	cluster-newyork: secondary, activated, synchronization status OK, Error, or Mismatch

# ◆◆◆ APPENDIX E

## Troubleshooting the Disaster Recovery Framework

---

This appendix describes procedures for troubleshooting your application of the disaster recovery framework.

This appendix contains the following sections:

- [“Troubleshooting Monitoring and Logging” on page 181](#)
- [“Troubleshooting Migration Problems” on page 182](#)
- [“Troubleshooting Cluster Start and Restart” on page 183](#)

### Troubleshooting Monitoring and Logging

This section provides the following information about setting up logging and problems that you might encounter with monitoring the disaster recovery framework:

- [“Configuring the Logger File to Avoid Too Many Traces” on page 181](#)
- [“Configuring the Logger File to Avoid Detailed Messages From the gc r Agent” on page 182](#)
- [“Configuring the Logger File to Avoid jmx Remote Traces” on page 182](#)

For information about logging, see [“Viewing the Disaster Recovery Framework Log Messages” on page 122](#).

### Configuring the Logger File to Avoid Too Many Traces

Configure the logger file, `/etc/cacao/instances/default/private/logger.properties`, as following depending on the `cmass` messages you want logged:

- To select only WARNING and SEVERE messages, the first line of the file should read as follows:

```
com.sun.cluster.level=WARNING
```

- To enable all geocontrol messages, the second line of the file should read as follows:

```
com.sun.cluster.agent.geocontrol.level=ALL
```

The enabled traces are copied to the `/var/cacao/instances/default/logs/cacao.0` file.

## Configuring the Logger File to Avoid Detailed Messages From the gcr Agent

If you want to avoid too detailed messages in your log file from the gcr agent, use entries similar to the following in your logger file `/etc/cacao/instances/default/private/logger.properties`:

```
com.sun.cluster.level=WARNING
com.sun.cluster.agent.geocontrol.gcr.level=INFO
com.sun.cluster.agent.geocontrol.level=ALL
```

This property file is updated each time you reinstall the SUNWscmasa package.

## Configuring the Logger File to Avoid jmx Remote Traces

To avoid jmx remote traces add the following lines to the beginning of your `logger.properties` file:

```
javax.management.remote.level=OFF
com.sun.jmx.remote.level=OFF
java.io.level=OFF
```

## Troubleshooting Migration Problems

This section provides information about problems that you might encounter when services are migrated by using the disaster recovery framework.

## Resolving Problems With Application Resource Group Failover When Communication Lost With the Storage Device

When a loss of communication occurs between a node on which the application is online and the storage device, some application resource groups might not failover gracefully to the nodes from which the storage is accessible. The application resource group might result in a `ERROR_STOP_FAILED` state.

**Solution or Workaround** – The disaster recovery framework does not initiate a switchover when I/O errors occur in a volume or its underlying devices. Because no switchover or failover occurs, the device service remains online on this node despite the fact that storage has been rendered inaccessible.

If this problem occurs, restart the application resource group on the correct nodes by using the standard Oracle Solaris Cluster procedures. Refer to [“Clearing the STOP\\_FAILED Error Flag on Resources” in \*Planning and Administering Data Services for Oracle Solaris Cluster 4.4\*](#) about recovering from the `ERROR_STOP_FAILED` state and restarting the application.

The disaster recovery framework detects state changes in the application resource group and displays the states in the output of the `geoadm status` command. For more information about using this command, see [“Monitoring the Runtime Status of the Disaster Recovery Framework” on page 115](#).

## Troubleshooting Cluster Start and Restart

This section provides the following information about troubleshooting problems that you might encounter with starting and restarting the disaster recovery framework:

- [“Validating Protection Groups in an Error State” on page 183](#)
- [“Administering Stopped Protection Groups After a Cluster Restart” on page 184](#)
- [“Restarting the Common Agent Container” on page 184](#)

## Validating Protection Groups in an Error State

After a cluster reboot the protection group configuration might be in an error state. This problem might be caused by the common agent container process not being available on one of the nodes of the cluster when the protection group is initialized after the reboot.

**Solution or Workaround** – To fix the configuration error, use the `geopg validate` command on the protection group that is in an error state.

## Administering Stopped Protection Groups After a Cluster Restart

If an entire cluster goes down then comes back up, the expected behavior is that disaster recovery framework does not restart the protection groups when the local cluster boots back up. After an entire cluster is restarted, the protection groups are deactivated on the local cluster, and the application resource groups in those protection groups will be in unmanaged state on the local cluster. After cluster restart, the administrator must determine whether data replication can be safely restarted and, if so, which cluster should have the primary role.

**Solution or Workaround** – After a cluster has rebooted, evaluate the state of the clusters and the condition of the storage or data. Then manually reset the roles of the protection groups and restart them, or perform a failback procedure, whichever is appropriate for the situation. If the cluster is a single-node cluster, protection groups must always be manually restarted after a node reboot, because only one node exists in the cluster. For more information, see [Chapter 11, “Migrating Services”](#) and see failback and post-takeover procedures in the disaster recovery framework guide for the replication component that you are using.

## Restarting the Common Agent Container

The Oracle Solaris Cluster software enables the common agent container only during the Oracle Solaris Cluster software installation. Therefore, if you disable the common agent container at any time after the installation, the common agent container remains disabled.

**Solution or Workaround** – To enable the common agent container after a node reboot, use the `/usr/lib/cacao/bin/cacaoadm enable` command.



## Error Return Codes for Script-Based Plug-Ins

---

### Error Codes That Are Returned in Script-Based Plug-Ins

The script-based plug-in MBean can return any of the error codes shown in the following table.

Return Code	Error Message	Description
101	E_SBP_PROGRAM_FAILED_TO_READ_CCR	Program {0} failed to read the cluster configuration repository (CCR)
110	E_SBP_PROGRAM_EXITED_NON_ZERO	Program {0} returned a nonzero exit code.
112	E_SBP_UNEXPECTED_ERROR	Unexpected error - {0}.
125	E_SBP_ONE_OR_MORE_RGS_NON_EXISTENT	One or more of the resource groups ({0}) returned by program {1} do not exist.
126	E_SBP_RG_LIST_WRONG_FORMAT,	The output {0} returned by program {1} is invalid. The output must conform to the format <code>rglist=comma-separated-resource-groups</code>
127	E_SBP_NO_SUCH_FILE	An attempt was made to execute a null or nonexistent command. Check the logs for more details.
128	E_SBP_CANNOT_READ_CONFIG_FILE	Unable to read configuration file {0} from any cluster node. This file must be available on all cluster nodes.
129	E_SBP_ENTRY_NOT_FOUND_IN_CONFIG_FILE	No entry for script-based plug-in configuration {0} exists in configuration file {1}.
130	E_SBP_CONFIG_FILE_FORMAT_ERROR	Field {0} in configuration file {1} must be {2}.
131	E_SBP_CONFIG_FILE_FIELD_FORMAT_ERROR	Configuration file {0} must have three fields per script-based plug-in entry. The fields must be separated by " ".
132	E_SBP_CONFIG_FILE_INVALID_NODE	The entry for script-based plug-in {0} in configuration file {1} contains an invalid cluster node {2} in the node list field.

## Error Codes That Are Returned in Script-Based Plug-Ins

133	E_SBP_FAILED_TO_CHECK_X_BIT	Failed to check whether {0} is executable on cluster node {1}.
134	E_SBP_SCRIPT_DOES_NOT_EXIST	Script or program {0} does not exist on cluster node {1}.
135	E_SBP_SCRIPT_FILE_IS_NOT_EXECUTABLE	Script or program {0} is not executable on cluster node {1}.
136	E_SBP_INTERRUPTED_OR_TIMED_OUT	The command or probe was interrupted or timed out.
138	E_SBP_COULD_NOT_GET_MBEAN_PROXY	Unable to get MBean proxy for {0} on node {1}.
139	E_SBP_FILE_SECURITY_ACCESS_REFUSED	The Java Security Manager refused access to file {0} on node {1}.
140	E_SBP_NULL_FILE_NAME	File name for property {0} was null.
141	E_SBP_UNABLE_TO_CREATE_SBP_CONFIG	Unable to create script-based plug-in configuration {0}.
142	E_SBP_UNABLE_TO_MODIFY_SBP_CONFIG	Unable to modify script-based plug-in configuration {0}.
143	E_SBP_UNABLE_TO_DELETE_DG	Unable to delete script-based plug-in configuration {0}.
144	E_SBP_UNABLE_TO_CREATE_PROPERTY	Unable to create property {0}.
146	E_SBP_UNABLE_TO_UPDATE_PG_PROPERTY	Unable to update protection group property {0}.
148	E_SBP_UNABLE_TO_UPDATE_PROPERTY	Unable to update property {0}.
150	E_SBP_UNABLE_TO_GET_PROPERTY	Unable to retrieve property.
151	E_SBP_UNABLE_TO_GET_CLUSTER_NODELIST	Unable to get cluster nodelist.
200	E_SBP_CONFIG_ERROR	Configuration error detected for protection group {0}.
201	E_SBP_SCRIPT_FAILED	The user supplied script-based plug-in command {0} failed with error code {1} on node {2}. The script error message is {3}.
210	E_SBP_INVALID_PROPERTY_FILE	Invalid property file {0}.
221	E_SBP_MISSING_PROPERTY	Property {0} is not set.
222	E_SBP_DUPLICATE_PROPERTY	Duplicate property {0}.
223	E_SBP_INVALID_PROPERTY	Invalid property {0}.
224	E_SBP_INVALID_PROPERTY_VALUE	Invalid value for property {0}.
225	E_SBP_SBP_CONFIG_ALREADY_IN_PG	Script-based plug-in configuration {0} already in protection group {1}.
226	E_SBP_SBP_CONFIG_NOT_FOUND_IN_PG	Script-based plug-in configuration {0} is not found in protection group {1}.
231	E_SBP_UNABLE_TO_NOTIFY_STATUS_CHANGE	Unable to send change notification for data replication status.
233	E_SBP_RG_OFFLINE_EXCEPTION	Failed to take resource group {0} offline.

---

## Error Codes That Are Returned in Script-Based Plug-Ins

---

234	E_SBP_SAME_PROPERTY_VALUE	Property value already set. No modification is needed.
235	E_SBP_UNEXPECTED_EXCEPTION	Unexpected exception - {0}.
236	E_SBP_SERVER_REQUEST_FAILED_DUE_TO_TIMEOUT	Error in running control script on host {0}. Operation timed out after {1} seconds.
237	E_SBP_SERVER_REQUEST_FAILED_WITH_REASON	Error in running control script on host {0} due to system error - {1}\



# Index

---

## A

accepting an invitation to join a site, 91

activating, 30

*See also* enabling

*See also* starting

disaster recovery framework, 30

protection groups, 73, 74

guidelines, 73

adding

clusters to a site, 89

custom heartbeat plug-in

to a custom heartbeat, 67

to the default heartbeat, 66

new cluster nodes to a partnership, 43

protection groups to a multigroup, 105

administering

access, 39

disaster recovery framework tasks, 25

heartbeats, 55

Oracle Solaris Cluster tasks, 23

overview, 19

protection groups, 73

security, 39

## B

booting a cluster, 36

## C

clusters

adding a new node, 43

adding to a site, 89

administration concepts, 23

booting, 36

changing a site role, 93

forcibly changing a site role, 95

joining a site, 91

leaving a partnership, 50

removing from a site, 98

removing from a site when unreachable, 100

sample configuration, 26

status of, 115

command-line interface

overview, 21

common agent container

restarting, 184

configuring

logger.properties file, 181

logical hostname, 24

role-change action script, 142

creating

custom heartbeat plug-in, 67

heartbeat plug-ins, 58

heartbeats, 56

role-change action script, 139

custom heartbeats

action script, 71

## D

data replication components

modifying, 82

removing from a protection group, 85

deactivating

protection groups, 73, 77

- guidelines, 73
- DEBUG property
  - script-based plug-ins, 123
- deleting, 50
  - See also* removing
  - heartbeat plug-ins, 61
  - heartbeats, 60
  - multigroups, 113
  - partnerships, 50
  - protection groups, 84
  - sites, 101
- dependency chains
  - modifying protection groups in, 107
  - syntax, 106
- detecting failure, 127
  - primary cluster, 127
  - secondary cluster, 128
- device groups
  - overview, 25
- disabling disaster recovery framework, 33
- disaster recovery framework
  - disabling, 33
  - enabling, 30
- disaster recovery overview, 175
- displaying
  - heartbeat configuration, 62
  - partnership configuration, 124
  - protection group configuration, 125

**E**

- enabling, 30
  - See also* activating
  - See also* starting
- enabling disaster recovery framework, 30
- examples
  - activating a protection group
    - globally, 77
    - locally, 77
  - adding a custom heartbeat plug-in to a new custom heartbeat, 68
  - adding a custom heartbeat plug-in to the default heartbeat, 66

- configuring a protection group custom command, 143
- configuring heartbeat-loss notification, 70
- creating a heartbeat, 57
- creating a heartbeat plug-in, 58
- deactivating a protection group
  - globally, 79
  - keeping application resource groups online, 79
  - locally, 79
- deleting a heartbeat, 60
- deleting a partnership, 52
- deleting a plug-in from a heartbeat, 61
- deleting a protection group, 85
- deleting a protection group with application resource groups online, 85
- disabling a cluster, 34
- displaying heartbeat configuration information, 62
- displaying partnership configuration information, 125
- displaying protection-group configuration information, 125
- displaying the disaster recovery framework status, 36
- enabling the disaster recovery framework, 32
- leaving a partnership, 51
- modifying a data replication component, 83
- modifying heartbeat plug-in properties, 60
- modifying partnership properties, 42
- modifying properties of the default heartbeat, 64
- notification action script, 72
- removing a data replication component from a protection group, 87
- renaming a cluster in a partnership, 47
- resynchronizing a partnership, 53
- sample clusters configuration, 26
- starting a protection group
  - globally, 77
  - locally, 77
- stopping a protection group
  - globally, 79
  - keeping application resource groups online, 79
  - locally, 79

stopping data replication on an online protection group, 79  
 switchover action script, 141  
 takeover of a protection group, 136

## F

failback-switchover, 138  
 failback-takeover, 138  
 failure  
   detecting, 127  
   detecting for secondary cluster, 128  
   primary cluster, 127

## G

geo-cluster-name, 29  
 geo-clusterstate, 29  
 geo-failovercontrol, 29  
 geo-hbmonitor, 29  
 geo-infrastructure, 29  
 geoadm show, 36  
 geoadm status, 115  
 guidelines  
   activating a protection group, 73  
   deactivating a protection group, 73  
   starting a protection group, 73  
   stopping a protection group, 73

## H

heartbeat plug-ins  
   adding a custom plug-in  
     to a custom heartbeat, 67  
     to the default heartbeat, 66  
   creating, 58  
   creating a custom plug-in, 64, 67  
   deleting, 61  
   modifying properties, 59  
 heartbeat-loss notification, 69  
 heartbeats, 58  
   *See also* heartbeat plug-ins  
   adding a custom heartbeat plug-in

    to a custom heartbeat, 67  
     to the default heartbeat, 66  
 administering, 55  
 creating, 56  
 creating a custom heartbeat plug-in, 64  
 custom action script, 71  
 deleting, 60  
 description, 55  
 displaying configuration, 62  
 heartbeat plug-in properties, 166  
 loss notification, 69  
 modifying properties, 63  
 properties, 165  
 tuning, 62

## J

joining a site, 91

## L

leaving  
   partnerships, 50  
 logger.properties file  
   configuring, 181  
 logging, 122  
   troubleshooting, 181  
 logical hostname  
   configuring, 24  
 loss of heartbeat notification, 69  
   creating action shell script, 71  
   properties, 70

## M

migration  
   troubleshooting, 182  
 modifying  
   data replication components, 82  
   heartbeat plug-in properties, 59  
   heartbeat properties, 63  
   partnership properties, 41  
   protection groups, 81

- protection groups in a dependency chain, 107
  - rights, 39
  - monitoring
    - disaster recovery framework, 115
    - resource groups, 29
    - troubleshooting, 181
  - multigroups
    - actions during switchover, 129
    - actions during takeover, 134
    - adding protection groups, 105
    - deleting, 113
    - properties, 171
    - removing a protection group, 108
    - starting, 110
    - status of, 115
    - stopping, 111
    - switchover, 132
    - synchronizing configuration information, 112
    - takeover, 136
- N**
- nodes
    - renaming, 44
  - notification\_actioncmd, 69
  - notification\_emailaddrs, 69
- O**
- operations
    - status of, 115
  - Oracle Solaris Cluster Manager
    - overview of, 20
    - tasks you can perform
      - activating protection groups, 74
      - adding a cluster to a site, 89
      - adding a protection group to a multigroup, 106
      - changing a protection group in a multigroup, 107
      - changing a site membership role, 94
      - checking disaster recovery framework status, 36
      - deactivating protection groups, 77
      - deleting a multigroup, 114
      - deleting a partnership, 51
      - deleting a site, 102
      - deleting protection groups, 84
      - displaying heartbeat configuration, 62
      - displaying partnership configuration information, 124
      - displaying protection group configuration information, 125
      - forcibly changing a site membership role, 95
      - joining a site, 91
      - leaving a partnership, 51
      - modifying data replication component properties, 82
      - modifying heartbeat plug-in properties, 59
      - modifying heartbeat properties, 63
      - modifying partnership properties, 41
      - modifying protection group properties, 81
      - removing a cluster from a site, 99
      - removing a protection group from a multigroup, 109
      - removing an unreachable cluster from a site, 100
      - removing data replication components, 86
      - removing trust, 49
      - resynchronizing a partnership, 53
      - resynchronizing a protection group, 80
      - starting all protection groups in a multigroup, 110
      - starting protection groups, 74
      - stopping all protection group in a multigroup, 111
      - stopping protection groups, 77
      - switching over protection groups, 131
      - synchronizing a multigroup, 113
      - synchronizing a site, 97
      - taking over protection groups, 135
  - Oracle Solaris Cluster software
    - administration concepts, 23
    - resources, 23
  - Oracle wallet, 75



**P**

- partnerships
  - adding a new cluster node, 43
  - deleting, 50
  - displaying configuration information, 124
  - leaving, 50
  - modifying, 41
  - properties, 167
  - removing trust, 49
  - renaming a cluster, 44
  - renaming a node in a cluster, 44
  - resynchronizing, 52
  - status of, 115
- primary cluster
  - failure detection, 127
  - recovering from a switchover, 138
  - recovering from a takeover, 138
  - switchover, 129
- properties
  - heartbeat, 165
  - heartbeat plug-in, 166
  - multigroup, 171
  - partnership, 167
  - protection group, 168
  - sites, 169
  - tuning heartbeat, 62
- protection groups
  - actions during switchover, 129
  - actions during takeover, 134
  - activating, 73, 74
    - guidelines, 73
  - adding to a multigroup, 105
  - administering, 73
  - configuring
    - role-change action script, 142
  - deactivating, 73, 77
    - guidelines, 73
  - deleting, 84
  - displaying configuration information, 125
  - modifying, 81
  - modifying in a dependency chain, 107
  - properties, 168
  - removing data replication components, 85

- removing from a multigroup, 108
- resynchronizing, 80
- starting, 74
  - guidelines, 73
  - in a multigroup, 110
- status of, 115
- stopping, 77
  - guidelines, 73
  - in a multigroup, 111
- switchover, 131
- takeover, 135
- troubleshooting after cluster restart, 184
- troubleshooting error state, 183

**R**

- recovering
  - after a switchover, 138
  - after a takeover, 138
- removing, 50
  - See also* deleting
  - all clusters from a site, 101
  - clusters from a site, 98
  - clusters from a site when unreachable, 100
  - data replication components from a protection group, 85
  - protection groups from a multigroup, 108
  - trust, 49
- renaming
  - a cluster, 44
  - cluster nodes, 44
- requirements
  - naming conventions, 173
  - role-change action script, 139
  - script-based plug-in scripts, 160
- resource groups
  - configuring, 23
  - disaster recovery framework, 29
- resources
  - configuring, 23
- resynchronizing
  - partnerships, 52
  - protection groups, 80

- rights
  - modifying, 39
- role-change action script, 139
  - configuring protection group for, 142
  - creating, 139
  - requirements, 139
- runtime status
  - disaster recovery framework, 115

**S**

- script-based plug-ins, 145
  - DEBUG property, 123
  - error codes, 185
- scripts
  - custom loss of heartbeat action, 71
  - switchover and takeover action, 139
- secondary cluster
  - failure detection, 128
  - switchover, 129
- security
  - administering, 39
- sites
  - accepting an invitation to join, 91
  - adding a cluster, 89
  - changing a cluster role, 93
  - deleting, 101
  - forcibly changing a cluster role, 95
  - joining, 91
  - properties, 169
  - removing a cluster, 98
  - removing all clusters, 101
  - removing an unreachable cluster, 100
  - status of, 115
  - synchronizing configuration information, 97
  - troubleshooting
    - joining a site, 93
- starting, 30
  - See also* activating
  - See also* enabling
  - cluster troubleshooting, 183
  - protection groups, 74
    - guidelines, 73

- protection groups in a multigroup, 110
- status
  - descriptions, 115
  - disaster recovery framework, 36
- stopping
  - all protection groups in a multigroup, 111
  - protection groups, 77
    - guidelines, 73
- switchover, 129
  - actions performed, 129
  - configuring role-change action script, 142
  - custom action script, 139
  - multigroups, 132
  - protection groups, 131
- synchronizing
  - multigroup configuration information, 112
  - site configuration information, 97

**T**

- takeover
  - actions performed, 134
  - configuring role-change action script, 142
  - custom action script, 139
  - multigroups, 136
  - protection groups, 135
  - results after, 177
- timeout
  - description of, 168, 171
- troubleshooting
  - cluster restart, 183
  - cluster start, 183
  - forcibly changing a site cluster role, 96
  - joining a site, 93
  - log messages, 122
  - logging, 181
  - migration problems, 182
  - monitoring, 181
  - protection group error state, 183
  - protection groups stopped after cluster restart, 184
  - restarting common agent container, 184
- trust
  - removing, 49

tuning

heartbeat properties, 62

