Oracle® Solaris Cluster Geographic Edition System Administration Guide

Part No: 821–2706–10
May 2011, Revision A

Copyright © 2004, 2011, Oracle and/or its affiliates. All rights reserved.

License Restrictions Warranty/Consequential Damages Disclaimer

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.

Warranty Disclaimer

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.

Restricted Rights Notice

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

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

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.

Trademark Notice

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 in the United States and other countries.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Copyright © 2004, 2011, 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 disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d’interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

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

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

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d’Advanced Micro Devices. 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. UNIX est une marque déposée concédé sous license par X/Open Company, Ltd.

110622@25097

Contents

Figures

Tables

Examples

Preface

Oracle Solaris Cluster Geographic Edition System Administration Guide provides procedures for administering Oracle Solaris Cluster Geographic Edition (Geographic Edition) software. This document is intended for experienced system administrators with extensive knowledge of Oracle software and hardware. This document is not to be used as a planning or presales guide.

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

Related Documentation

Information about related Geographic Edition topics is available in the documentation that is listed in the following table. All Geographic Edition documentation is available at `http://www.oracle.com/technetwork/indexes/documentation/index.html`.

Topic	Documentation
Overview	*Oracle Solaris Cluster Geographic Edition Overview* *Oracle Solaris Cluster Geographic Edition 3.3 5/11 Documentation Center*
Installation	*Oracle Solaris Cluster Geographic Edition Installation Guide*
Data Replication	*Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Oracle Data Guard* *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
System administration	*Oracle Solaris Cluster Geographic Edition System Administration Guide*

For a complete list of Geographic Edition documentation, see *Oracle Solaris Cluster Geographic Edition 3.3 5/11 Release Notes*.

Topic	Documentation
Concepts	*Oracle Solaris Cluster Concepts Guide*
Hardware installation and administration	*Oracle Solaris Cluster 3.3 Hardware Administration Manual* Individual hardware administration guides
Software installation	*Oracle Solaris Cluster Software Installation Guide*
Data service installation and administration	*Oracle Solaris Cluster Data Services Planning and Administration Guide* Individual data service guides
Data service development	*Oracle Solaris Cluster Data Services Developer's Guide*
System administration	*Oracle Solaris Cluster System Administration Guide* *Oracle Solaris Cluster Quick Reference*
Software upgrade	*Oracle Solaris Cluster Upgrade Guide*
Error messages	*Oracle Solaris Cluster Error Messages Guide*
Command and function references	*Oracle Solaris Cluster Reference Manual* *Oracle Solaris Cluster Data Services Reference Manual* *Oracle Solaris Cluster Quorum Server Reference Manual*

Related Third-Party Web Site References

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Using UNIX Commands

This document contains information about commands that are used to install, configure, or administer an Geographic Edition configuration. This document might not contain complete information on basic UNIX commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following sources for this information:

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

Documentation and Support

See the following web sites for additional resources:

- Documentation (`http://www.oracle.com/technetwork/indexes/documentation/index.html`)
- Support (`http://www.oracle.com/us/support/systems/index.html`)

Oracle Software Resources

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the Discussion Forums (`http://forums.oracle.com`).
- Get hands-on step-by-step tutorials with Oracle By Example (`http://www.oracle.com/technetwork/tutorials/index.html`).

Obtaining Help

If you have problems installing or using Geographic Edition software, contact your service provider and provide the following information:

- Your name and email address (if available)
- Your company name, address, and phone number
- The model and serial numbers of your systems
- The release number of the operating system (for example, Solaris 10)
- The release number of the Geographic Edition software (for example, 3.3 5/11)
- The contents of the `/var/cacao/instances/default/logs/cacao.0/1/2` file

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

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices

Command	Function
`psrinfo -v`	Displays information about processors
`showrev –p`	Reports which patches are installed
`prtdiag -v`	Displays system diagnostic information
`geoadm -V`	Displays the Geographic Edition software release information
`cluster status`	Provides a snapshot of the cluster status
`cluster show`	Lists cluster configuration information
`geoadm status`	Displays the Geographic Edition runtime status of the local cluster

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

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

CHAPTER 1

Introduction to Administering the Geographic Edition Software

Oracle Solaris Cluster Geographic Edition (Geographic Edition) software protects applications from unexpected disruptions by using multiple clusters that are geographically separated. These clusters contain identical copies of the Geographic Edition infrastructure, which manage replicated data between the clusters. Geographic Edition software is a layered extension of the Oracle Solaris Cluster software.

This chapter contains the following sections:

Geographic Edition Administration

Familiarize yourself with the planning information in the *Oracle Solaris Cluster Geographic Edition Installation Guide* and the *Oracle Solaris Cluster Geographic Edition Overview* before beginning administration tasks. This guide contains the standard tasks that are used to administer and maintain the Geographic Edition 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 Geographic Edition software without causing any nodes or the cluster to fail. You can install, configure, start, use, stop, and uninstall the Geographic Edition software 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.

Geographic Edition Administration Tools

You can perform administrative tasks on a cluster that is running Geographic Edition software by using a graphical user interface (GUI) or the command-line interface (CLI).

- "Graphical User Interface" on page 22
- "Command-Line Interface" on page 22

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

Graphical User Interface

Oracle Solaris Cluster software supports Oracle Solaris Cluster Manager, a GUI tool that you can use to perform various administrative tasks on your cluster. For specific information about how to use Oracle Solaris Cluster Manager, see the Oracle Solaris Cluster online help.

Note – To administer Geographic Edition software by using the Oracle Solaris Cluster Manager – Geographic Edition GUI, ensure that the root passwords are the same on all nodes of both clusters in the partnership.

You can only use the GUI to administer Geographic Edition software after the software infrastructure has been enabled by using the `geoadm start` command. Use a shell to run the `geoadm start` and `geoadm stop` commands. For information about enabling and disabling the Geographic Edition infrastructure, see Chapter 3, "Administering the Geographic Edition Infrastructure."

The GUI 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 start the GUI, go to the following URL from any Java-enabled and Javascript-enabled browser, and log in as root.

Note – RBAC is not supported in the GUI.

```
https://clustername:6789
```

Command-Line Interface

Table 1–1 lists the commands that you can use to administer the Geographic Edition software. For more information about each command, refer to the *Oracle Solaris Cluster Geographic Edition Reference Manual*.

TABLE 1–1 Geographic Edition CLI

Command	Description
geoadm	Enables or disables the Geographic Edition software 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 Geographic Edition software
geops	Creates and manages the partnerships between clusters
geopg	Configures and manages protection groups

Analyzing the Application for Suitability

This section describes the guidelines you must follow in creating applications to be managed by Geographic Edition software.

Before you create an application to be managed by Geographic Edition software, determine whether the application satisfies the following requirements for being made highly available or scalable.

Note – If the application fails to meet all requirements, modify the application source code to make it highly available or scalable.

- Both network-aware (client-server model) and network-unaware (client-less) applications are potential candidates for being made highly available or scalable in the Geographic Edition environment. However, Geographic Edition cannot provide enhanced availability in timesharing environments in which applications are run on a server that is accessed through `telnet` or `rlogin`.
- The application must be crash tolerant. That is, it must recover disk data (if necessary) when it is started after an unexpected node death. Furthermore, the recovery time after a crash must be bounded. Crash tolerance is a prerequisite for making an application highly available because the ability to recover the disk and restart the application is a data integrity issue. The data service is not required to be able to recover connections.
- The application must not depend on the physical host name of the node on which it is running.
- The application must operate correctly in environments in which multiple IP addresses are configured to go up. Examples include environments with multihomed hosts, in which the node is located on more than one public network, and environments with nodes on which multiple, logical interfaces are configured to go up on one hardware interface.

- Application binaries and libraries can be located locally on each node or in the cluster file system. The advantage of being located in the cluster file system is that a single installation is sufficient. The disadvantage is that when you use rolling upgrade for Oracle Solaris Cluster software, the binaries are in use while the application is running under the control of the Resource Group Manager (RGM).
- The client must have capacity to retry a query automatically if the first attempt times out. If the application and the protocol already handle the case of a single server crashing and rebooting, they also can handle the containing resource group failing over or switching over.
- The application must not have UNIX domain sockets or named pipes in the cluster file system.

A scalable service must meet all the preceding conditions for high availability as well as the following additional requirements.

- The application must have the ability to run multiple instances, all operating on the same application data in the cluster file system.
- The application must provide data consistency for simultaneous access from multiple nodes.
- The application must implement sufficient locking with a globally visible mechanism, such as the cluster file system.

For a scalable service, application characteristics also determine the load-balancing policy. For example, the load-balancing policy `Lb_weighted`, which allows any instance to respond to client requests, does not work for an application that makes use of an in-memory cache on the server for client connections. In this case, you should specify a load-balancing policy that restricts a given client's traffic to one instance of the application. The load-balancing policies `Lb_sticky` and `Lb_sticky_wild` repeatedly send all requests by a client to the same application instance, where they can make use of an in-memory cache. If multiple client requests come in from different clients, the RGM distributes the requests among the instances of the service.

See Chapter 2, “Developing a Data Service,” in *Oracle Solaris Cluster Data Services Developer's Guide* for more information about setting the load-balancing policy for scalable data services.

The application must be able to meet the following data replication requirements:

- Information replicated must not be host– or cluster-specific.

  When the application fails over to the remote site, the application might run on a host with a different IP address. To allow client nodes to find the remote site, use a Geographic Edition action script to update the DNS/NIS mapping.
- If you don't want your application to tolerate any data loss, the application should use synchronous replication.

CHAPTER 2

Before You Begin

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

This chapter contains the following sections:

- "Overview of Oracle Solaris Cluster Administration Concepts" on page 25
- "Overview of Geographic Edition Administration Tasks" on page 28
- "Example Geographic Edition Cluster Configuration" on page 33

Overview of Oracle Solaris Cluster Administration Concepts

You must be an experienced Oracle Solaris Cluster administrator to administer Geographic Edition software.

This section describes the Oracle Solaris Cluster administration topics that you need to understand before you administer the Geographic Edition software.

Configuring Resources and Resource Groups

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

For more information about administering resources and resource groups in Oracle Solaris Cluster software, see the *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

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 Geographic Edition Entities."

For more information about using the `geoadm start` command , see "Enabling the Geographic Edition Software" on page 36.

Note – If you are using Sun StorageTek Availability Suite for data replication, a logical hostname is created for each device group to be replicated. For more information, see Chapter 1, "Replicating Data With Sun StorageTek Availability Suite Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*.

The following table lists the Oracle Solaris Cluster and Geographic Edition 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 Solaris OS software

TABLE 2–1 IP Addresses Required by Geographic Edition Software

Component	Number of IP Addresses Needed
Oracle Solaris Cluster administrative console	1 per subnet
IP Network Multipathing groups	■ Single-adapter groups – 1 primary IP address. For the Solaris 8 release, also 1 test IP address for each adapter in the group. ■ Multiple-adapter groups – 1 primary IP address plus 1 test IP address for each adapter in the group.
Cluster nodes	1 per node, per subnet
Domain console network interface (Sun Fire 15000)	1 per domain
Console-access device	1
Logical addresses	1 per logical host resource, per subnet

TABLE 2–1 IP Addresses Required by Geographic Edition Software *(Continued)*

Component	Number of IP Addresses Needed
Geographic Edition infrastructure hostname	1 logical IP address per cluster infrastructure. For example, if you have two clusters in your Geographic Edition infrastructure, you need two IP addresses.
Replication with Sun StorageTek Availability Suite software	1 dedicated logical IP address on the local cluster for each device group to be replicated. For example, if you have two clusters in your Geographic Edition infrastructure, 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 *Oracle Solaris Cluster Software Installation Guide*.

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, Solaris Volume Manager disk sets, and VERITAS Volume Manager disk groups.

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 *Oracle Solaris Cluster System Administration Guide*.

The Geographic Edition software configures Oracle Solaris Cluster device groups to include replication.

For more information about configuring data replication in Geographic Edition software, see Chapter 1, "Replicating Data With Sun StorageTek Availability Suite Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*, Chapter 1, "Replicating Data With Hitachi TrueCopy and Universal Replicator Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*, and Chapter 1, "Replicating Data With EMC Symmetrix Remote Data Facility Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*.

Overview of Geographic Edition Administration Tasks

This section provides a starting point for administering the Geographic Edition software. This section contains the following tasks:

- "Prerequisite Administration Tasks" on page 28
- "Geographic Edition Administration Tasks" on page 28

Prerequisite Administration Tasks

Before you begin administering the Geographic Edition software, you must identify the Oracle Solaris Cluster installations you need to host protection groups. Then, you need to adjust the Oracle Solaris Cluster configuration and environment to support the formation of partnerships and protection groups with the Geographic Edition software. The following table describes these prerequisite tasks.

TABLE 2–2 Geographic Edition Prerequisite Tasks

Task	Description
Set the `SC-clustername` to the cluster name you want to use with the Geographic Edition software.	Use the `cluster`(1CL) command. For more information, see "How to Enable Geographic Edition Software" on page 36.
Set up the IP address and host maps for the cluster that is enabled to run Geographic Edition software.	See Chapter 2, "Installing Software on Global-Cluster Nodes," in *Oracle Solaris Cluster Software Installation Guide*.
Install and configure your data replication product.	See the Sun StorageTek Availability Suite, Hitachi TrueCopy and Universal Replicator, or EMC Symmetrix Remote Data Facility documentation. This step is required before you can create protection groups with the `geopg create` command.
Port and configure application configuration and corresponding resource groups on clusters that are candidates for partnership.	You can use the Oracle Solaris Cluster `scsnapshot` tool to facilitate porting of application resource groups. See "Creating and Modifying a Partnership" on page 56 for more information.
Enable the common agent container on all nodes of both clusters.	See "Enabling the Geographic Edition Software" on page 36.

Geographic Edition Administration Tasks

After you have completed the prerequisite administration tasks, you can install, configure, and administer the Geographic Edition software as described in the following table.

TABLE 2–3 Geographic Edition Administration Tasks

Task	Description and Documentation
Install Geographic Edition software.	See the *Oracle Solaris Cluster Geographic Edition Installation Guide*.
Set up security between the candidate partner clusters.	■ Exchange certificates, as described in "Configuring Secure Cluster Communication Using Security Certificates" on page 49. ■ (Optional) Configure a secure logical hostname that uses IP Security Architecture (IPsec), as described in "Configuring Secure Cluster Communication Using IPsec" on page 50.
Enable the Geographic Edition software.	Use the `geoadm start` command. For more information, see "Enabling the Geographic Edition Software" on page 36.
Create partnerships.	See "How to Create a Partnership" on page 57. This procedure includes the following: ■ Modifying the default heartbeat. For more information, see Chapter 6, "Administering Heartbeats." ■ Configuring loss of heartbeat notification. For more information, see "Configuring Heartbeat-Loss Notification" on page 86.
Configure data replication.	For information about replicating data by using Sun StorageTek Availability Suite, see Chapter 1, "Replicating Data With Sun StorageTek Availability Suite Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*. For information about replicating data by using Hitachi TrueCopy and Universal Replicator, see Chapter 1, "Replicating Data With Hitachi TrueCopy and Universal Replicator Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*. For information about replicating data by using EMC Symmetrix Remote Data Facility, see Chapter 1, "Replicating Data With EMC Symmetrix Remote Data Facility Software," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*.

TABLE 2–3 Geographic Edition Administration Tasks *(Continued)*

Task	Description and Documentation
Create protection groups.	■ Create a protection group. See one of the following data replication guides: ■ "How to Create and Configure a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "How to Create and Configure a Hitachi TrueCopy or Universal Replicator Protection Group That Does Not Use Oracle Real Application Clusters" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "How to Create and Configure an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* ■ Add data replication device groups. See one of the following data replication guides: ■ "How to Add a Data Replication Device Group to a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "How to Add a Data Replication Device Group to a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "How to Add a Data Replication Device Group to an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* ■ Add application resource groups to the protection group. See one of the following data replication guides: ■ "How to Add an Application Resource Group to a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "How to Add an Application Resource Group to a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "How to Add an Application Resource Group to an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* ■ Create a protection group that does not require data replication. See "Creating a Protection Group That Does Not Require Data Replication" on page 92.

TABLE 2–3 Geographic Edition Administration Tasks *(Continued)*

Task	Description and Documentation
Bring the protection groups online.	See one of the following data replication guides: ■ "How to Activate a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "How to Activate a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "How to Activate an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*
Test the configured partnership and protection groups to validate the setup.	Perform a trial switchover or takeover and test some simple failure scenarios. See one of the following data replication guides: ■ Chapter 3, "Migrating Services That Use Sun StorageTek Availability Suite Data Replication," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ Chapter 3, "Migrating Services That Use Hitachi TrueCopy and Universal Replicator Data Replication," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ Chapter 3, "Migrating Services That Use SRDF Data Replication," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* **Note** – You cannot perform personality swaps if you are running EMC Symmetrix Remote Data Facility/Asynchronous data replication.
Migrate services to the partner cluster.	See one of the following data replication guides: ■ "How to Switch Over a Sun StorageTek Availability Suite Protection Group From Primary to Secondary" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "How to Switch Over a Hitachi TrueCopy or Universal Replicator Protection Group From Primary to Secondary" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "How to Switch Over an SRDF Protection Group From Primary to Secondary" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* **Note** – You cannot perform personality swaps if you are running EMC Symmetrix Remote Data Facility/Asynchronous data replication.

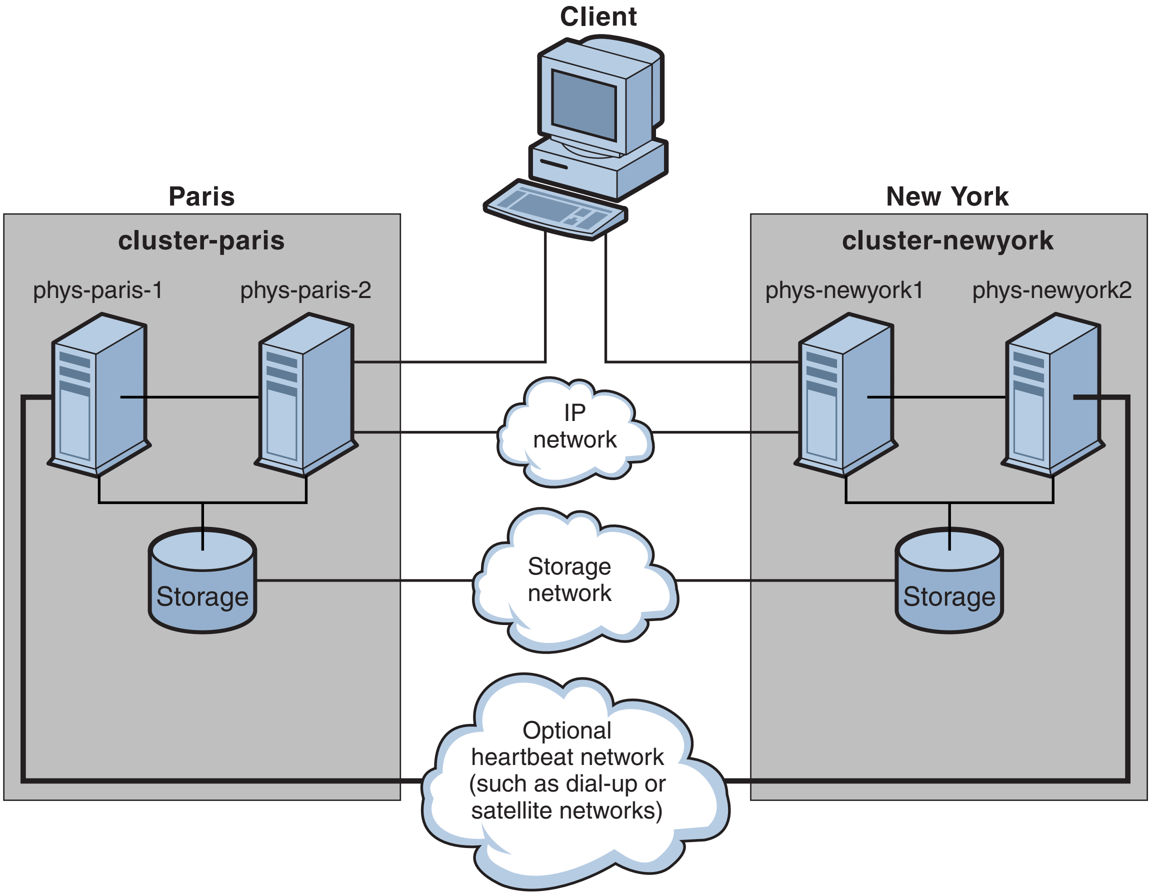
TABLE 2–3 Geographic Edition Administration Tasks *(Continued)*

Task	Description and Documentation
Take over services from primary to secondary during a disaster.	See one of the following data replication guides: ■ "How to Force Immediate Takeover of Sun StorageTek Availability Suite Services by a Secondary Cluster" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "How to Force Immediate Takeover of Hitachi TrueCopy or Universal Replicator Services by a Secondary Cluster" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "How to Force Immediate Takeover of SRDF Services by a Secondary Cluster" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*
Recover from a takeover.	■ Data recovery and error repair outside of the Geographic Edition infrastructure. See the Sun StorageTek Availability Suite, Hitachi TrueCopy and Universal Replicator, or EMC Symmetrix Remote Data Facility documentation. ■ Resynchronize the partner clusters. See "Recovering Sun StorageTek Availability Suite Data After a Takeover" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*, "Recovering Services to a Cluster on a System That Uses Hitachi TrueCopy or Universal Replicator Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*, or "Recovering Services to a Cluster on a System That Uses SRDF Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*.
Take a protection group offline.	See "How to Deactivate a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*, "How to Deactivate a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*, or "How to Deactivate an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*.
Delete a protection group.	See "How to Delete a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*, "How to Delete a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*, or "How to Delete an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*.
Delete a partnership.	See "Leaving or Deleting a Partnership" on page 69.
Disable the Geographic Edition software.	See "How to Disable the Geographic Edition Software" on page 39.
Uninstall the Geographic Edition software.	See the *Oracle Solaris Cluster Geographic Edition Installation Guide*.

Example Geographic Edition Cluster Configuration

The following figure describes an Geographic Edition cluster configuration that is used throughout this guide to illustrate the Geographic Edition 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 2–1 Example Cluster Configuration



CHAPTER 3

Administering the Geographic Edition Infrastructure

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

This chapter contains the following sections:

Geographic Edition Infrastructure Resource Groups

When you enable the Geographic Edition infrastructure, the following Oracle Solaris Cluster resource groups are created:

- `geo-clusterstate` – A scalable resource group that the Geographic Edition software uses to distinguish between node failover and cluster reboot scenarios. This resource group does not contain any resources. The resource group contains the following resources:
  - `geo-servicetag` - A scalable resource that is started on all nodes of a cluster when Geographic Edition is present. When the Geographic Edition software is started, this resource checks for the existence of a Solaris service tag for the running version of Geographic Edition on each node, and creates a service tag if necessary. The service tag indicates that Geographic Edition has been used on the cluster. This service tag is removed from the node when the Geographic Edition packages are removed.
  - `geo-zc-sysevent` - (For zone clusters only) Runs resource methods in the global zone when Geographic Edition 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 Geographic Edition infrastructure. The resource group contains the following resources:
  - `geo-clustername` – The logical hostname for the Geographic Edition software. The Geographic Edition software uses the logical hostname of a cluster for inter-cluster 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 Geographic Edition software.
  - `geo-failovercontrol` – Encapsulates the Geographic Edition software itself. The Geographic Edition 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 Geographic Edition infrastructure.

You can monitor the status of these resources by using the `clresource status` command. For more information about this command, see the `clresource(1CL)` man page.

Enabling the Geographic Edition Software

When you enable the Geographic Edition software, the cluster is ready to enter a partnership with another enabled cluster. You can use the CLI commands or the GUI to create a cluster partnership.

For more information about setting up and installing the Geographic Edition software, see the *Oracle Solaris Cluster Geographic Edition Installation Guide*.

▼ How to Enable Geographic Edition Software

This procedure enables the Geographic Edition infrastructure 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 Solaris Operating System and the Oracle Solaris Cluster software.
- The Oracle Solaris Cluster management-agent container for Oracle Solaris Cluster Manager is running.
- The Geographic Edition 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.

When you upgrade to Oracle Solaris Cluster 3.3 5/11 software, the security certificates must be identical on all nodes of the cluster. Therefore, you must copy the security certificates manually from one node of the cluster to the other nodes of the cluster. For more information on copying the security files for the common agent container, see the procedures in "How to Finish Upgrade to Oracle Solaris Cluster 3.3 5/11 Software" in *Oracle Solaris Cluster Upgrade Guide*.

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

You must be assigned the Geo Operation RBAC rights profile to complete this procedure. For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

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 64.

Follow cluster naming guidelines as described in "Planning Required IP Addresses and Hostnames" in *Oracle Solaris Cluster Geographic Edition Installation Guide*. Cluster names must follow the same requirements as for host names.

```
# cluster rename -c newclustername oldclustername
```

For more information, see the `cluster(1CL)` man page.

Note – After you have enabled the Geographic Edition infrastructure, you must not change the cluster name while the infrastructure 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 Geographic Edition infrastructure.**

```
# geoadm start
```

The `geoadm start` command enables the Geographic Edition infrastructure on the local cluster only. For more information, see the `geoadm(1M)` man page.

6 **Verify that you have enabled the infrastructure and that the Geographic Edition resource groups are online.**

For a list of the Geographic Edition resource groups, see "Geographic Edition Infrastructure Resource Groups" on page 35.

```
# geoadm show
# clresourcegroup status
# clresource status
```

The output for the `geoadm show` command displays that the Geographic Edition infrastructure is active from a particular node in the cluster.

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

For more information, see the `clresourcegroup`(1CL) and `clresource`(1CL) man pages.

Example 3–1 Enabling the Geographic Edition Infrastructure in a Cluster

This example enables the Geographic Edition software on the `cluster-paris` cluster.

1. Start the Geographic Edition software on `cluster-paris`.

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

2. Ensure that the Geographic Edition infrastructure was successfully enabled.

   ```
   phys-paris-1# geoadm show

   --- CLUSTER LEVEL INFORMATION ---
   Sun Cluster Geographic Edition is active on cluster-paris from node phys-paris-1
   Command execution successful
   phys-paris-1#
   ```

3. Verify the status of the Geographic Edition resource groups and resources.

```
phys-paris-1# clresourcegroup 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-clustername       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 Geographic Edition Data Replication Guide that corresponds to the type of data replication software you are using.

Disabling the Geographic Edition Software

You can disable the Geographic Edition infrastructure by using the following procedure.

▼ How to Disable the Geographic Edition Software

Before You Begin

- Ensure that all protection groups on the local cluster are offline.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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 Geographic Edition Software" on page 95.

Caution – If you want to keep the application resource groups online while deactivating a protection group, follow the procedure described in the following data replication guides:

- "How to Deactivate a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
- "How to Deactivate a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- "How to Deactivate an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

3 **Disable the Geographic Edition software.**

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

This command removes the infrastructure resource groups that were created when you enabled the Geographic Edition infrastructure.

For more information about this command, see the `geoadm`(1M) man page.

Note – Disabling the Geographic Edition software removes only the infrastructure 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 Geographic Edition resource groups are no longer displayed.**

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

For more information, see the `clresourcegroup`(1CL) man page.

Example 3–2 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 Geographic Edition infrastructure.

```
phys-paris-1# geoadm stop
... verifying pre conditions and performing pre remove operations ... done
...removing product infrastructure ... please wait ...
```

3. Confirm that the Geographic Edition infrastructure was successfully disabled.

```
phys-paris-1# geoadm show

--- CLUSTER LEVEL INFORMATION ---
Sun Cluster Geographic Edition 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 Geographic Edition resource groups and resources have been removed.

```
phys-paris-1# clresourcegroup status
phys-paris-1#
```

Checking the Status of the Geographic Edition Infrastructure

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

EXAMPLE 3–3 Displaying Whether the Geographic Edition Infrastructure Has Been Enabled

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

EXAMPLE 3–3 Displaying Whether the Geographic Edition Infrastructure Has Been Enabled *(Continued)*

```
phys-paris-1# geoadm show

--- CLUSTER LEVEL INFORMATION ---
Sun Cluster Geographic Edition 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 Oracle Solaris Cluster infrastructure is enabled, the Geographic Edition software 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 Geographic Edition Software" on page 95.

Applying Patches to a Geographic Edition System

Observe the following guidelines and requirements to patch Geographic Edition software:

- You must run the same patch levels for Oracle Solaris Cluster software and the common agent container software on all nodes of the same cluster.
- Within a cluster, the patch level for each node on which you have installed Geographic Edition software must meet the Oracle Solaris Cluster software patch-level requirements.
- All nodes in the same cluster must have the same version of Geographic Edition software and the same patch level. However, primary and secondary clusters can run different versions of Geographic Edition software, provided that each version of Geographic Edition is correctly patched and the versions are no more than one release different.
- To ensure that the patches have been installed properly, install the patches on your secondary cluster before you install the patches on the primary cluster.
- For additional information about Geographic Edition patches, see the patch `README` file.
- See the Geographic Edition release notes for a list of required patches.

▼ How to Prepare an Geographic Edition System for Patches

1. **Ensure that the cluster is functioning properly.**

   To view the current status of the cluster, run the following command from any node:

   ```
   % cluster status
   ```

   See the cluster(1CL) man page for more information.

   Search the /var/adm/messages log on the same node for unresolved error messages or warning messages.

2. **Become superuser on a node of the global cluster.**

3. **Remove all application resource groups from protection groups.**

   This step ensures that resource groups are not stopped when you later stop the protection groups.

   ```
   # geopg remove-resource-group resourcegroup protectiongroup
   ```

   See the geopg(1M) man page for more information.

4. **Perform the preceding steps on all clusters that have a partnership with this cluster.**

5. **Stop all protection groups that are active on the cluster.**

   ```
   # geopg stop -e local protectiongroup
   ```

   See the geopg(1M) man page for more information.

6. **Stop the Geographic Edition infrastructure.**

   ```
   # geoadm stop
   ```

   Shutting down the infrastructure ensures that a patch installation on one cluster does not affect the other cluster in the partnership.

   See the geoadm(1M) man page for more information.

7. **On each node, stop the common agent container.**

   ```
   # /usr/sbin/cacaoadm stop
   ```

Next Steps Install the required patches for the Geographic Edition software. Go to "How to Install Patches on an Geographic Edition System" on page 44.

▼ How to Install Patches on an Geographic Edition System

Perform this procedure on all nodes of the cluster.

Patch the secondary cluster before you patch the primary cluster, to permit testing.

Before You Begin Perform the following tasks:

- Ensure that the Solaris OS is installed to support Geographic Edition software.

  If Solaris software is already installed on the node, you must ensure that the Solaris installation meets the requirements for Geographic Edition software and any other software that you intend to install on the cluster.
- Ensure that Geographic Edition software packages are installed on the node.
- Ensure that you completed all steps in "How to Prepare an Geographic Edition System for Patches" on page 43.

1 **Ensure that all the nodes are online and part of the cluster.**

To view the current status of the cluster, run the following command from any node:

```
% cluster status
```

See the `cluster`(1CL) man page for more information.

Search the `/var/adm/messages` log on the same node for unresolved error messages or warning messages.

2 **Become superuser in the global zone of a node.**

3 **Install any necessary patches to support Geographic Edition software by using the `patchadd` command.**

If you are applying Oracle Solaris Cluster patches, use the Oracle Solaris Cluster methods on both clusters.

4 **Repeat Step 2 and Step 3 on each remaining node.**

5 **After you have installed all required patches on all nodes of the cluster, on each node of the global cluster or zone cluster that you are configuring with Geographic Edition, start the common agent container.**

```
# /usr/sbin/cacaoadm start
```

6 **On one node, enable Geographic Edition software.**

```
# geoadm start
```

7 **Add all application resource groups that you removed while you were preparing the cluster for a patch installation back to the protection group.**

```
# geopg add-resource-group resourcegroup protectiongroup
```

See the geopg(1M) man page for more information.

8 **Start all the protection groups that you have added.**

```
# geopg start -e local [-n] protectiongroup
```

See the geopg(1M) man page for more information.

Next Steps After you patch the secondary cluster, perform a sanity test on the Geographic Edition software, and then repeat this procedure on the primary cluster.

◆ ◆ ◆ CHAPTER 4

Administering Access and Security

This chapter describes how to administer access and security. It contains the following sections:

- "Geographic Edition Software and RBAC" on page 47
- "Configuring Secure Cluster Communication Using Security Certificates" on page 49
- "Configuring Firewalls" on page 49
- "Configuring Secure Cluster Communication Using IPsec" on page 50

Geographic Edition Software and RBAC

This section describes role-based access control (RBAC) in Geographic Edition software. It contains the following sections:

- "Setting Up and Using RBAC" on page 47
- "RBAC Rights Profiles" on page 48
- "Modifying a User's RBAC Properties" on page 48

Setting Up and Using RBAC

Geographic Edition software bases its RBAC profiles on the RBAC rights profiles that are used in the Oracle Solaris Cluster software. For general information about setting up and using RBAC with Oracle Solaris Cluster software, refer to Chapter 2, "Oracle Solaris Cluster and RBAC," in *Oracle Solaris Cluster System Administration Guide*.

Geographic Edition software adds the following new RBAC entities to the appropriate file in the `/etc/security` directory:

- RBAC authentication names to `auth_attr`
- RBAC execution profiles to `prof_attr`
- RBAC execution attributes to `exec_attr`

Note – The default search order for the `auth_attr` and `prof_attr` databases is `files nis`, which is defined in the `/etc/nsswitch.conf` file. If you have customized the search order in your environment, confirm that `files` is in the search list. Including `files` in the search list enables your system to find the RBAC entries that Geographic Edition defined.

RBAC Rights Profiles

The Geographic Edition CLI and GUI use RBAC rights to control end-user access to operations. The general conventions for these rights are described in Table 4–1.

TABLE 4–1 Geographic Edition RBAC Rights Profiles

Rights Profile	Included Authorizations	Role Identity Permission
Geo Management	`solaris.cluster.geo.read`	Read information about the Geographic Edition entities
	`solaris.cluster.geo.admin`	Perform administrative tasks with the Geographic Edition software
	`solaris.cluster.geo.modify`	Modify the configuration of the Geographic Edition software
Basic Solaris User	Solaris authorizations	Perform the same operations that the Basic Solaris User role identity can perform
	`solaris.cluster.geo.read`	Read information about the Geographic Edition entities

When you use the Geo Management RBAC rights profile to administer configurations that use Oracle Data Guard or script-based plug-ins, ensure that the correct ACLs for `/var/cluster/geo` are set on each node of both partner clusters. If necessary, use the following command to set the ACLs:

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

Modifying a User's RBAC Properties

When you grant authorization to users other than superuser, you must do so on all nodes of both partner clusters. Otherwise, some operations that have a global scope might fail, due to insufficient user rights on one or more nodes in the partnership.

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

For example, you can assign the Geo Management RBAC 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 RBAC properties for a user, refer to Chapter 2, "Oracle Solaris Cluster and RBAC," in *Oracle Solaris Cluster System Administration Guide*.

Configuring Secure Cluster Communication Using Security Certificates

You must configure the Geographic Edition software for secure communication between partner clusters. The configuration must be reciprocal, so cluster `cluster-paris` must be configured to trust its partner cluster `cluster-newyork`and cluster `cluster-newyork` must be configured to trust its partner cluster `cluster-paris`.

If you are using the GUI to administer the Geographic Edition software, the root password must be the same on all nodes of both partner clusters.

For information about setting up security certificates for partner clusters, see "Configuring Trust Between Partner Clusters" on page 53.

For information about the example cluster configuration, see "Example Geographic Edition Cluster Configuration" on page 33.

Configuring Firewalls

Geographic Edition partner clusters communicate using transport services and ICMP echo requests and replies (pings). Their packets must therefore pass data center firewalls, including any firewalls configured on cluster nodes in partner clusters. The table below contains a list of required and optional services and protocols used by Geographic Edition partnerships, and the associated ports that you must open in your firewalls for these services to function. The ports listed are defaults, so if you customize the port numbers serving the specified transfer protocols, the customized ports must be opened instead.

Ports other than those listed in Table 4–2 might be required by storage replication services such as the Sun StorageTek Availability Suite product. See product documentation for details.

TABLE 4–2 Ports and Protocols Used by Oracle Solaris Cluster Geographic Edition Partnerships

Port Number	Protocols	Use in Geographic Edition partnership
Required Services		
22	UDP and TCP	Secure shell (ssh). Used during the initial certificate transfer that establishes trust between partner clusters.
2084	UDP (default), TCP	Inter-cluster heartbeat
11162	TCP	The Java Management Extensions (JMX) port (`jmxmp-connector-port`). A messaging protocol used for the exchange of configuration and status information between the two sites in a partnership.
-	ICMP Echo Request/Reply	Backup heartbeat between partner clusters
Optional Services		
161	TCP and UDP	Simple Network Management Protocol (SNMP) communications
162	TCP and UDP	SNMP traps
6789	TCP and UDP	The Oracle Solaris Cluster Manager GUI

Configuring Secure Cluster Communication Using IPsec

You can use IP Security Architecture (IPsec) to configure secure communication between partner clusters. IPsec enables you to set policies that permit or require either secure datagram authentication, or actual data encryption, or both, between machines communicating by using IP. Consider using IPsec for the following cluster communications:

- Secure Sun StorageTek Availability Suite communications, if you use the Sun StorageTek Availability Suite software for data replication
- Secure TCP/UDP heartbeat communications

Oracle Solaris Cluster software and Geographic Edition software support IPsec by using only manual keys. Keys must be stored manually on the cluster nodes for each combination of server and client IP address. The keys must also be stored manually on each client.

Refer to the Part IV, "IP Security," in *System Administration Guide: IP Services* for a full description of IPsec configuration parameters.

How to Configure IPsec for Secure Cluster Communication

In the Geographic Edition infrastructure, the hostname of a logical host is identical to the cluster name. The logical hostname is a special HA resource. You must set up a number of IP addresses for various Geographic Edition components, depending on your cluster configuration.

On each partner cluster, you must configure encryption and authorization for exchanging inbound and outbound packets from a physical node to the logical-hostname addresses. The values for the IPsec configuration parameters on these addresses must be consistent between partner clusters.

IPsec uses two configuration files:

- **IPsec policy file**, `/etc/inet/ipsecinit.conf`. Contains directional rules to support an authenticated, encrypted heartbeat. The contents of this file are different on the two clusters of a partnership.
- **IPsec keys file**, `/etc/init/secret/ipseckeys`. Contains keys files for specific authentication and encryption algorithms. The contents of this file are identical on both clusters of a partnership.

The following procedure configures a cluster, `cluster-paris`, for IPsec secure communication with another cluster, `cluster-newyork`. The procedure assumes that the local logical hostname on `cluster-paris` is `lh-paris-1` and that the remote logical hostname is `lh-newyork-1`. Inbound messages are sent to `lh-paris-1` and outbound messages are sent to `lh-newyork-1`.

Use the following procedure on each node of `cluster-paris`.

1 **Log in to the first node of the primary cluster, `phys-paris-1`, as superuser.**

For a reminder of which node is `phys-paris-1`, see "Example Geographic Edition Cluster Configuration" on page 33.

2 **Set up an entry for the local address and remote address in the IPsec policy file.**

The policy file is located at `/etc/inet/ipsecinit.conf`. Permissions on this file should be `644`. For more information about this file, see the `ipsecconf`(1M) man page.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

a. Configure the communication policy.

The default port for the `tcp_udp` plug-in is 2084. You can specify this value in the`etc/cacao/instances/default/modules/com.sun.cluster.geocontrol.xml` file.

The following command configures a policy with no preference for authorization or encryption algorithms.

```
# {raddr lh-newyork-1 rport 2084} ipsec {auth_algs any encr_algs any \
sa shared} {laddr lh-paris-1 lport 2084} ipsec {auth_algs any encr_algs \
any sa shared}
```

When you configure the communication policy on the secondary cluster, `cluster-newyork`, you must reverse the policies.

```
# {laddr lh-newyork-1 lport 2084} ipsec {auth_algs any encr_algs \
any sa shared} {raddr lh-paris-1 rport 2084} ipsec {auth_algs any encr_algs \
any sa shared}
```

b. Add the policy by rebooting the node or by running the following command.

```
# ipsecconf -a /etc/inet/ipsecinit.conf
```

3 Set up encryption and authentication keys for inbound and outbound communication.

The communication file is located at `/etc/init/secret/ipseckeys`. Permissions on the file should be `600`.

Add keys:

```
# ipseckey -f /etc/init/secret/ipseckeys
```

Key entries have the following general format:

```
# inbound to cluster-paris
add esp spi paris-encr-spi dst lh-paris-1 encr_alg paris-encr-algorithm \
encrkey paris-encrkey-value
add ah spi newyork-auth-spi dst lh-paris-1 auth_alg paris-auth-algorithm \
authkey paris-authkey-value

# outbound to cluster-newyork
add esp spi newyork-encr-spi dst lh-newyork-1 encr_alg newyork-encr-algorithm \
encrkey newyork-encrkey-value
add ah spi newyork-auth-spi dst lh-newyork-1 auth_alg newyork-auth-algorithm \
authkey newyork-authkey-value
```

For more information about the communication files, see the `ipsecconf(1M)` man page.

CHAPTER 5

Administering Cluster Partnerships

This chapter provides the procedures for administering partnerships between two Geographic Edition software-enabled clusters.

This chapter contains the following sections:

Configuring Trust Between Partner Clusters

Before you create a partnership between two clusters, you must configure the Geographic Edition software for secure communication between the two clusters. The configuration must be reciprocal. For example, you must configure the cluster `cluster-paris` to trust the cluster `cluster-newyork`, and you must also configure the cluster `cluster-newyork` to trust the cluster `cluster-paris`.

▼ How to Configure Trust Between Two Clusters

Before You Begin Ensure that the following conditions are met:

- The cluster on which you want to create the partnership is running.
- The `geoadm start` command must have already been run on this cluster and the partner cluster. For more information about using the `geoadm start` command, see "Enabling the Geographic Edition Software" on page 36.

- The cluster name of the partner cluster is known.
- The host information of the partner cluster must defined in the local host file. The local cluster needs to know how to reach the partner cluster by name.

  If the clusters are in different domains, include the domain name in the entry, as *logicalhostname.domainname*. However, the cluster name itself must not include the domain.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 **Import the public keys from the remote cluster to the local cluster.**

Running this command on one node of the local cluster imports the keys from the remote cluster to one node of the cluster.

```
# geops add-trust -c remotepartnerclustername
```

`-c` *remotepartnerclustername*[.*domainname*]
Specifies the logical hostname of the cluster with which to form a partnership. The logical hostname is used by the Geographic Edition software and maps to the name of the remote partner cluster. For example, a remote partner cluster name might resemble the following:

```
cluster-paris
```

If the clusters are on different domains, also specify the fully qualified domain name. For example, two clusters in a partnership that have different domains might resemble the following:

```
cluster-paris.france
cluster-newyork.usa
```

When you use this option with the `add-trust` or `remote-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:

remotepartnercluster`.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(1M) man page.

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

If you choose to use Oracle Solaris Cluster Manager, skip this step. Oracle Solaris Cluster Manager handles all nodes in a single operation.

4 **Verify trust from one node of each cluster.**

```
# geops verify-trust -c remotepartnerclustername[.domainname]
```

This command verifies the trust from the node on which you run the command to all nodes of the partner cluster. If you choose to use Oracle Solaris Cluster Manager, it verifies the trust from all nodes of the local cluster to all nodes of the partner cluster

See Also For a complete example of how to configure and join a partnership, see Example 5–4.

▼ How to Remove Trust Between Two Clusters

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 defined in the local host file. The local cluster needs to know how to reach the partner cluster by name.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the /var/cluster/geo ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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 remotepartnerclustername
```

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

`-c` *remotepartnerclustername*
: 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 `remote-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:

  remotepartnercluster`.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`(1M) man page.

If you choose to use Oracle Solaris Cluster Manager, it handles all nodes of a cluster in a single operation.

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

Creating and Modifying a Partnership

The Geographic Edition software enables clusters to form partnerships between clusters to provide mutual protection against disasters. The clusters in a partnership monitor each other by sending heartbeat messages to each other in the same way that nodes of a single cluster do. Unlike local clusters, the clusters in a partnership use the public network for these messages, but support additional, plug-in mechanisms as well.

You create only one partnership between two specific clusters by using the `geops`(1M) command. After you have created a partnership, you can use this command to modify the properties of this partnership.

When creating partnerships, ensure that the name of all the clusters in the partnership are unique. For example, if you have a cluster wholly within the domain `.france`, you can use hostnames like `paris` and `grenoble`. However, if you have a cross-domain cluster, you must specify the hostnames with enough qualification to identify the host on the network. You can link `paris` and `munich` with hostnames `paris.france` and `munich.germany`, and the cluster names remain `paris` and `munich`.

You cannot create a partnership between clusters `paris.france` and `paris.texas` because of a collision on the cluster name `paris`.

The names of the application resource groups that are managed by the Geographic Edition software must be the same on both partner clusters. You can configure the names of these resource groups manually or by using the `scsnapshot` command.

The `scsnapshot` command replicates configuration data on a cluster that does not have configured resource groups, resource types, and resources. The `scsnapshot` command retrieves the configuration data from the cluster on which it is launched and generates a script called `scriptfile`. Edit the script to adapt it to the specific features of the cluster where you want to replicate the configuration data. For example, you might have to change the IP address and host names in the script. Launch the script from any node in the cluster where you want to replicate the configuration data. For more information about using this command, see the `scsnapshot`(1M) man page.

You can define only one partnership between two specific clusters. A single cluster can participate in other partnerships with different clusters.

▼ How to Create a Partnership

Before You Begin Ensure that the following conditions are met:

- The cluster on which you want to create the partnership is up and running.
- If a partner cluster is a zone cluster, either application-based replication such as Oracle Data Guard is configured or no data replication is used.
- The `geoadm start` command must have already been run on the this cluster and the partner cluster. For more information about using the `geoadm start` command, see "Enabling the Geographic Edition Software" on page 36.
- The cluster name of the partner cluster is known.
- The host information of the partner cluster must defined in the local host file. The local cluster needs to know how to reach the partner cluster by name.
- Security has been configured on the two clusters by installing the appropriate certificates.

  See "Configuring Trust Between Partner Clusters" on page 53 for more information.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 Create the partnership.

```
# geops create -c remotepartnerclustername[.domainname] [-h heartbeatname] \
[-p propertysetting [-p...]] partnershipname
```

`-c` *remotepartnerclustername*[*.domainname*]

Specifies the name of the remote cluster that will participate in the partnership. If clusters in the partnership are in different domains, you must also specify the domain name of the remote cluster.

This name matches the logical hostname used by the Geographic Edition infrastructure on the remote cluster.

`-h` *heartbeatname*

Specifies a custom heartbeat to use in the partnership to monitor the availability of the partner cluster.

If you omit this option, the default Geographic Edition heartbeat is used.

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. For more information about configuring custom heartbeats, see Chapter 6, “Administering Heartbeats.”

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

You must configure the custom heartbeat that you provide in this option before you run the `geops` command.

Note – 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 you run the `geops create` command.

`-p` *propertysetting*

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_emailaddrs` and `Notification_actioncmd` properties. For more information about configuring heartbeat-loss notification, see "Configuring Heartbeat-Loss Notification" on page 86.

For more information about the properties you can set, see Appendix A, "Standard Geographic Edition Properties."

partnershipname
Specifies the name of the partnership.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geops` command, refer to the `geops`(1M) man page.

3 **Verify that the partnership was created and the status of the partnership.**

```
# geoadm status
```

Example 5–1

Creating a Partnership

This example creates the `paris-newyork-ps` partnership on the `cluster-paris.usa` cluster.

```
# geops create -c cluster-newyork.usa -p Description=Transatlantic \
-p Notification_emailaddrs=sysadmin@companyX.com paris-newyork-ps
# geoadm status
```

See Also For a complete example of how to configure and join a partnership, see Example 5–4.

▼ How to Modify Partnership Properties

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 **Modify partnership properties.**

```
# geops set-prop -p propertysetting [-p...] partnershipname
```

`-p` *propertysetting* 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_emailaddrs` and `Notification_actioncmd` properties. For more information about configuring heartbeat-loss notification, see "Configuring Heartbeat-Loss Notification" on page 86.

For more information about the properties you can set, see Appendix A, "Standard Geographic Edition Properties."

partnershipname Specifies the name of the partnership.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geops` command, refer to the `geops`(1M) man page.

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

```
# geops list
```

Example 5–2 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@companyX.com \
paris-newyork-ps
# geops list
```

Joining an Existing Partnership

When you define and configure a partnership, the partnership specifies a second cluster to be a member of that partnership. Then, you must configure this second cluster to join the partnership.

▼ How to Join a Partnership

Perform this procedure from a node of the cluster that is joining the partnership.

Before You Begin Ensure that the following conditions are met:

- The local cluster is enabled to run the Geographic Edition software.

- The partnership you want the cluster to join is defined and configured on another cluster (`cluster-paris`) and the local cluster (`cluster-newyork`) is specified as a member of this partnership.
- If a partner cluster is a zone cluster, either application-based replication such as Oracle Data Guard is configured or no data replication is used.
- Security has been configured on the clusters by installing the appropriate certificates.

  See "Configuring Secure Cluster Communication Using Security Certificates" on page 49 for more information.

1. **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

   For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

   Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

   ---

2. **Confirm that the remote cluster that originally created the partnership, `cluster-paris`, can be reached at its logical hostname.**

   ```
   # ping lh-paris-1
   ```

   For information about the logical hostname of the cluster, see "How to Enable Geographic Edition Software" on page 36.

3. **Join the partnership.**

   ```
   # geops join-partnership [-h heartbeatname] remoteclustername partnershipname
   ```

   `-h` *heartbeatname*

   Specifies a custom heartbeat to use in the partnership to monitor the availability of the partner cluster.

   If you omit this option, the default Geographic Edition heartbeat is used.

   Custom heartbeats are provided for special circumstances and require careful configuration. Consult your Sun specialist for assistance if your system requires the use of custom heartbeats. For more information about configuring custom heartbeats, see Chapter 6, "Administering Heartbeats."

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

You must configure the custom heartbeat that you provide in this option before you run the geops command.

remoteclustername Specifies the name of a cluster that is currently a member of the partnership that is being joined. This cluster is used to retrieve the partnership configuration information.

partnershipname Specifies the name of the partnership.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the geops command, refer to the geops(1M) man page.

4 **Verify that the cluster was added to the partnership and that the partnership properties were defined correctly.**

```
# geops list
# geoadm status
```

Example 5–3 Joining a Partnership

This example joins the cluster-newyork cluster in the partnership that was created on cluster-paris in Example 5–1.

```
# geops join-partnership cluster-paris paris-newyork-ps
# geops list
# geoadm status
```

Example 5–4 Creating and Joining a Partnership With a Remote Cluster in a Different Domain

This example creates and configures the paris-newyork-ps partnership between clusters cluster-paris.france and cluster-newyork.usa.

1. On one node of cluster-paris.france, configure trust for the partnership.

   ```
   phys-paris-1# geops add-trust -c cluster-newyork.usa
   ```

2. On one node of cluster-newyork.usa, configure trust for the partnership.

   ```
   phys-newyork-1# geops add-trust -c cluster-paris.france
   ```

3. On each node of both clusters, verify that trust has been set up properly, both between the local cluster and partner cluster and among nodes of the local cluster.

   ```
   phys-newyork-1# geops verify-trust -c cluster-paris.france
   phys-newyork-2# geops verify-trust -c cluster-paris.france
   phys-newyork-1# geops verify-trust
   phys-newyork-2# geops verify-trust
   ```

```
phys-paris-1# geops verify-trust -c cluster-newyork.usa
phys-paris-2# geops verify-trust -c cluster-newyork.usa
phys-paris-1# geops verify-trust
phys-paris-2# geops verify-trust
```

4. On `cluster-paris.france`, create the partnership `paris-newyork-ps`.

   ```
   cluster-paris# geops create -c cluster-newyork.usa -p Description=Transatlantic \
   -p Notification_emailaddrs=sysadmin@companyX.com paris-newyork-ps
   ```

5. On `cluster-newyork.usa`, join the partnership `paris-newyork-ps`.

   ```
   cluster-newyork# geops join-partnership cluster-paris.france paris-newyork-ps
   ```

6. Verify that the partnership has been created successfully.

   ```
   # geops list
   # geoadm status
   ```

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 Geographic Edition 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, "Adding and Removing a Node," in *Oracle Solaris Cluster System Administration Guide*.

2 **Install Geographic Edition, data replication, and application software on the new node.**

- To install Geographic Edition software, see *Oracle Solaris Cluster Geographic Edition Installation Guide*.
- To install data replication and application software, see the appropriate manual 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 resourcegroup protectiongroup
```

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

```
# geopg stop -e local protectiongroup
```

5 **Stop the Geographic Edition infrastructure.**

```
# geoadm stop
```

6 **Re-enable the Geographic Edition infrastructure.**

This action recreates each Geographic Edition 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 protectiongroup
```

8 **Restore any application resource groups that you removed in Step 3.**

```
# geopg add-resource-group resourcegroup protectiongroup
```

Renaming a Cluster Node

You can rename a node in a Geographic edition 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 Geographic edition cluster, see "How to Rename a Node" in *Oracle Solaris Cluster System Administration Guide*.

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 app-rg pg1
```

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

```
# geopg list pg1
```

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

This task stops data replication.

```
# geopg stop pg1 -e global
```

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

```
# geopg delete pg1
```

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

```
# geops leave-partnership ps1
```

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 Geographic Edition software.**

```
# geoadm stop
```

8 **From one node of each cluster, confirm that Geographic Edition software 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 *Oracle Solaris Cluster Geographic Edition Installation Guide*.

```
# cluster rename -c new-clustername
```

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 Geographic Edition 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-clustername          there should be no response
# echo "IPaddress  new-clustername" >> /etc/inet/hosts
```

12 **From one node of each cluster, start Geographic Edition software.**

```
# geoadm start
```

If Geographic Edition software 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 `cacaoadm restart` command, then start Geographic Edition software.

13 **From one node of each cluster, verify that Geographic Edition software is successfully started.**

```
# geoadm status
```

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

```
# geops add-trust -c remotepartnerclustername[.domainname]
```

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

Note – Do not specify a domain name to the `verify-trust` subcommand.

```
# geops verify-trust -c remotepartnerclustername
```

16 **Create and join a new partnership between the clusters.**

a. **From the primary cluster, create the partnership.**

```
# geops create -c remotepartnerclustername[.domainname] partnershipname
```

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

```
# geops join-partnership remotepartnerclustername[.domainname] partnershipname
```

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 5–5 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 data-replication guide:

- Chapter 2, “Administering Sun StorageTek Availability Suite Protection Groups,” in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*

- Chapter 2, "Administering Hitachi TrueCopy and Universal Replicator Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- Chapter 2, "Administering Hitachi TrueCopy and Universal Replicator Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- Chapter 2, "Administering SRDF Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*
- Chapter 2, "Administering Oracle Data Guard Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Oracle Data Guard*

Leaving or Deleting a Partnership

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

Because this command destroys the local partnership configuration information, when the last member leaves a partnership, the partnership no longer exists.

▼ How to Leave a Partnership

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 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the /var/cluster/geo ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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

```
# geopg list
```

If you find that the partnership contains protection groups, you can delete them with the `geopg delete` command. For information about deleting protection groups, see one of the following data replication guides:

- "How to Delete a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
- "How to Delete a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- "How to Delete an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

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

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

partnershipname 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`(1M) man page.

Example 5–6 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 5–7 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

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

Before You Begin Ensure that the following conditions are met:

- The local cluster is Geographic Edition 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 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see “Geographic Edition Software and RBAC” on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 **Resynchronize the partnership.**

```
# geops update partnershipname
```

partnershipname Specifies the name of the partnership

Example 5–8 Resynchronizing a Partnership

This example resynchronizes a partnership.

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

CHAPTER 6

Administering Heartbeats

Geographic Edition software 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

A heartbeat in Geographic Edition 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 Geographic Edition 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 Geographic Edition 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_`*localclustername~remoteclustername*. For more information about the `geops` command, refer to the `geops`(1M) 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 Sun 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.**

2. **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

   For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the /var/cluster/geo ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

3 **Create the heartbeat.**

```
# geohb create -r remoteclustername \
[-p propertysetting [-p...]] heartbeatname
```

-r *remoteclustername* Specifies the name of the remote, secondary partner cluster.

-p *propertysetting* 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 Geographic Edition Properties.”

heartbeatname 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(1M) man page.

Example 6–1

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. **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

   For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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

```
# geohb add-plugin heartbeatname pluginname \
[-p propertysetting [-p...]]
```

heartbeatname Specifies the identifier for heartbeat on the local cluster.

pluginname Specifies the name of the heartbeat plug-in.

`-p`*propertysetting* 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 Geographic Edition Properties."

For more information about the `geohb` command, refer to the `geohb`(1M) man page.

Example 6–2 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

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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

```
# geohb modify-plugin -p propertysetting \
[-p...] pluginname  heartbeatname
```

heartbeatname Specifies an identifier for the heartbeat.

pluginname Specifies the name of the heartbeat plug-in.

`-p` *propertysetting* 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 Geographic Edition Properties."

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

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geohb` command, refer to the `geohb`(1M) man page.

Example 6–3 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 describes procedures for deleting heartbeats and heartbeat plug-ins.

▼ How to Delete a Heartbeat

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 **Delete the heartbeat.**

```
# geohb delete heartbeatname
```

heartbeatname Specifies an identifier for the heartbeat settings.

For more information about the `geohb` command, refer to the `geohb`(1M) man page.

Example 6–4 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 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the /var/cluster/geo ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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

```
# geohb remove-plugin pluginname heartbeatname
```

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

---

pluginname	Specifies the name of the custom heartbeat plug-in
heartbeatname	Specifies an identifier for the heartbeat that contains this plug-in

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the geohb command, refer to the geohb(1M) man page.

Example 6–5 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

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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

```
# geohb list [heartbeatnamelist]
```

heartbeatnamelist 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`(1M) man page.

Example 6–6 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`(1M) man page.

Note – Custom heartbeats are provided for special circumstances and require careful configuration. Consult your Sun 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.geocontol.xml` file.

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

```
Query_interval > 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:

```
Query_interval > 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 a Sun service representative.

▼ How to Modify the Heartbeat Properties

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 Modify the heartbeat properties.

```
# geohb set-prop -p propertysetting \
[-p...] heartbeatname
```

`-p` *propertysetting* 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 Geographic Edition Properties."

heartbeatname Specifies an identifier for the heartbeat settings.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geohb` command, refer to the `geohb`(1M) man page.

Example 6–7 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.

Creating a Custom Heartbeat Plug-in

When a heartbeat is created, your custom heartbeat plug-in is passed the following arguments by the Geographic Edition software:

queryinterval — 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`.

pluginpropertyvalues — 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 Geographic Edition 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 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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

```
# geohb add-plugin -p propertysetting [-p...] \
pluginname hb_localclustername-remoteclustername
```

`-p` *propertysetting* — 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 Geographic Edition Properties."

pluginname

Specifies the name of the custom heartbeat plug-in.

`hb_`*localclustername*`-`*remoteclustername*

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 6–8 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 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 Create the new custom heartbeat.

```
# geohb create -r remoteclustername \
[-p propertysetting [-p...]] heartbeatname
```

-r *remoteclustername* Specifies the name of the remote, secondary partner cluster.

-p *propertysetting* 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 Geographic Edition Properties.”

heartbeatname 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`(1M) man page.

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

```
# geohb add-plugin -p propertysetting [-p...] \
pluginname  heartbeatname
```

-p *propertysetting* 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 Geographic Edition Properties.”

pluginname Specifies the name of the custom heartbeat plug-in.

heartbeatname 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 remoteclustername -h heartbeatname \
[-p propertysetting [-p...]] partnershipname
```

-c *remoteclustername* Specifies the name of remote cluster that will participate in the partnership.

This name matches the logical hostname used by the Geographic Edition infrastructure on the remote cluster.

-h *heartbeatname*

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

-p *propertysetting*

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 Geographic Edition Properties.”

partnershipname

Specifies the name of the partnership.

For more information about using geops create command to create a partnership, see “How to Create a Partnership” on page 57.

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

```
# geoadm status
```

Example 6–9 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

You can configure the Geographic Edition software 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_emailaddrs 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 Geographic Edition software runs after a heartbeat-loss event.

Configuring the Heartbeat-Loss Notification Properties

You can configure heartbeat-loss notification by using two partnership properties, `Notification_emailaddrs` 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" on page 57. You can also modify these properties by using the procedure that is described in "How to Modify the Heartbeat Properties" on page 81.

If you want to be notified of heartbeat loss by email, set the `Notification_emailaddrs` 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 the *Solaris System Administration Guide: Network Services*.

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

EXAMPLE 6–10 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_emailaddrs=ops@paris.com,ops@newyork.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:

```
# customactioncommandpath -c localclustername -r remoteclustername -e 1 \
-n nodename -t time
```

customactioncommandpath Specifies a path to the action command you have created.

`-c` *localclustername* Specifies the name of the local cluster.

`-p` *remoteclustername* Specifies the name of the remote partner cluster.

`-e1` Specifies that `HBLOST=1`, which indicates that a heartbeat-loss event has occurred. The Geographic Edition software only supports heartbeat-loss notification, so `-e 1` is the only value that can be passed to the action shell script.

`-n`*nodename* 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 6–11 How a Notification Action Script Parses the Command-Line Information Provided by the Geographic Edition Software

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 $*
```

EXAMPLE 6–11 How a Notification Action Script Parses the Command-Line Information Provided by the Geographic Edition Software *(Continued)*

```
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
```

CHAPTER 7

Administering Protection Groups

This chapter contains the procedures for creating and configuring protection groups that do not require data replication. The chapter contains the following sections:

- "Introduction to Protection Groups" on page 91
- "Creating a Protection Group That Does Not Require Data Replication" on page 92

Introduction to Protection Groups

Protection groups enable a set of clusters to tolerate and recover from disaster by managing the resource groups for services. Protection groups can exist only in a partnership. You must create a partnership before you can create a protection group for that partnership. A protection group contains application resource groups and properties for managing data replication for those application resource groups.

You can duplicate the application resource group configuration on partner clusters. The configuration for a protection group is identical on partner clusters, so partner clusters must have the application resource groups of the protection group defined in their configuration. The Geographic Edition software propagates protection group configurations between partners.

You can specify a data replication type in the protection group to indicate the mechanism that is used for data replication between partner clusters. When a service is protected from disaster by data replication, the protection group also contains replication resource groups. Protection groups link an application in a resource group with the application data that should be replicated. This linkage and replication enable the application to fail over seamlessly from one cluster to another cluster.

For information about how to create a protection group that requires data replication, see the following data replication guides:

- Chapter 2, "Administering Sun StorageTek Availability Suite Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*

- Chapter 2, "Administering Hitachi TrueCopy and Universal Replicator Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- Chapter 2, "Administering Hitachi TrueCopy and Universal Replicator Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- Chapter 2, "Administering SRDF Protection Groups," in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

Creating a Protection Group That Does Not Require Data Replication

Some protection groups do not require data replication. If you are using the Geographic Edition software to manage only resource groups, you can create protection groups that do not replicate data. The `geoadm status` command displays that these protection groups are in the `Degraded` state. This section describes how to configure your protection group not to use data replication.

Note – You cannot add device groups to a protection group that does not use data replication.

▼ How to Create a Protection Group That Is Configured Not to Use Data Replication

Before You Begin Before you create a protection group, ensure that the following conditions are met:

- The local cluster is a member of a partnership.
- The protection group that you are creating does not already exist.

Note – Protection group names are unique in the global Geographic Edition namespace. You cannot use the same protection group name in more than one partnership on the same system.

---

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 **Create a new protection group by using the `geopg create` command.**

This command creates a protection group on the local cluster.

```
# geopg create -s partnershipname -o localrole \
[-p property [-p...]] \
protectiongroupname
```

`-s` *partnershipname*

Specifies the name of the partnership.

`-o` *localrole*

Specifies the role of this protection group on the local cluster as either `Primary` or `Secondary`.

`-p` *propertysetting*

Specifies the properties of the protection group.

You can specify the following properties:

- `Description` – Describes the protection group.
- `External_Dependency_Allowed` – Specifies whether to allow any 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.
- `RoleChange_ActionArgs` – Specifies a string that follows system-defined arguments at the end of the command line when the role-change callback command runs.
- `RoleChange_ActionCmd` – 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.

  This path should be valid on all nodes of all partner clusters that can host the protection group.

- `Timeout` – Specifies the timeout period for the protection group in seconds. You can change the timeout period from the default value depending on the complexity of your data replication configuration. For more information on setting the timeout period, see Table A–4.

For more information about the properties you can set, see Appendix A, "Standard Geographic Edition Properties."

protectiongroupname Specifies the name of the protection group.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geopg` command, refer to the `geopg`(1M) man page.

Example 7–1 Creating and Configuring a Protection Group That Is Configured to Not Use Data Replication

This example creates a protection group that is configured to not use data replication.

```
# geopg create -s paris-newyork-ps -o primary example-pg
```

Next Steps See one of the following guides for information about adding resource groups to a protection group.

- "Administering Sun StorageTek Availability Suite Application Resource Groups" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
- "Administering Hitachi TrueCopy and Universal Replicator Application Resource Groups" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- "Administering SRDF Application Resource Groups" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

CHAPTER 8

Monitoring and Validating the Geographic Edition Software

This chapter describes the files and tools that you can use to monitor and validate the Geographic Edition software.

This chapter contains the following sections:

- "Monitoring the Runtime Status of the Geographic Edition Software" on page 95
- "Viewing the Geographic Edition Log Messages" on page 101
- "Displaying Configuration Information for Partnerships and Protection Groups" on page 101

Monitoring the Runtime Status of the Geographic Edition Software

You can display the runtime status of the local Geographic Edition 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 all partnership, 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
- Pending operations – Provides status information about any ongoing transaction processes

You must be assigned the Basic Solaris User RBAC rights profile to run the `geoadm status` command. For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

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
      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       : Activated
   Configuration             : OK
   Data replication          : OK
   Resource groups           : OK

  Cluster cluster-newyork    : OK
     Role                    : Secondary
     PG activation state     : Activated
     Configuration           : OK
     Data replication        : OK
     Resource groups         : OK

Pending Operations
Protection Group     : "tcpg"
Operation            : start
```

The information displayed shows that the protection group, `tcpg`, is activated 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 8–1 Status Value Descriptions

Field	Value Descriptions
`Partnership`	`OK` – The partners are connected. `Error` – The connection between the partner clusters is lost. `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.

TABLE 8–1 Status Value Descriptions *(Continued)*

Field	Value Descriptions
`Synchronization`	`OK` – The configuration information is synchronized between partner clusters. `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. For information about resynchronizing a partnership, see "Resynchronizing a Partnership" on page 71. For information about resynchronizing a protection group, see one of the following data replication guides: ■ "Resynchronizing a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "Resynchronizing a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "Resynchronizing an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* `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 `geopg get` command. `Unknown` – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.
`ICRM Connection`	`OK` – The Intercluster Resource Management (ICRM) module is running properly. `Error` – The ICRM module on the local cluster is unable to communicate with the ICRM module on the remote cluster.
`Heartbeat`	`OK` – Heartbeat checks are running and the partner cluster responds within the specified timeout and retry periods. `Offline` – Heartbeat checks are not running. `Error` – Heartbeat checks are running but the partner is not responding and retries have timed out. `Degraded` – Heartbeat checks are running but one of the primary plug-ins is degraded or not running.

TABLE 8–1 Status Value Descriptions *(Continued)*

Field	Value Descriptions
`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` – The synchronization state is `OK` and the state of the protection group on each cluster is `OK`. `Degraded` – The synchronization state is `OK`. The state of the protection group is `Degraded` on either one or both clusters in the partnership. `Unknown` – The synchronization state or the state of the protection group on one or both clusters is unavailable. The protection group can be online or offline. `Error` – The synchronization state or the state of the protection group on one or both clusters is in `Error`. The protection group can be online or offline.
`Protection group` > `Cluster` (state of protection group on each cluster)	`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. `Unknown` – The state of some components of the protection group, such as configuration data, data replication, or resource groups, is unavailable. `Error` – The state of some components of the protection group, such as configuration data, data replication, or resource groups, is in `Error`.
`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.
`Protection group` > `Cluster` > `PG activation state`	`Activated` – The protection group is activated. `Deactivated` – The protection group is deactivated. `Unknown` – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.

TABLE 8–1 Status Value Descriptions *(Continued)*

Field	Value Descriptions
`Protection group > Cluster > Configuration`	`OK` – Protection group configuration has been validated without errors on the cluster. `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 one of the following data replication guides: ■ "How to Validate a Sun StorageTek Availability Suite Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite* ■ "Validating a Hitachi TrueCopy or Universal Replicator Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator* ■ "Validating an SRDF Protection Group" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility* `Unknown` – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.

TABLE 8–1 Status Value Descriptions *(Continued)*

Field	Value Descriptions
`Protection group` > `Cluster` > `Data replication`	`None` – Data replication is not configured. `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. `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. `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. `Unknown` – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached. `N/A` – The data replication state of the protection group could not be mapped. Data replication is in a valid state on its own but in an `Error` state for the protection group. This state is available only if you are using Sun StorageTek Availability Suite data replication.
`Protection group` > `Cluster` > `Resource groups`	`None` – No resource group is protected by this protection group. `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. `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. `Unknown` – Information is not accessible because the partners are disconnected or because some components of the protection group cannot be reached.

For more specific information about checking the runtime status of replication, see one of the following data replication guides:

- "Checking the Runtime Status of Sun StorageTek Availability Suite Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
- "Checking the Runtime Status of Hitachi TrueCopy and Universal Replicator Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
- "Checking the Runtime Status of SRDF Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

Viewing the Geographic Edition Log Messages

All the Geographic Edition components produce messages that are stored in log files.

Information about the loading, running, and stopping Geographic Edition 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 Geographic Edition modules are updated on the node where the Geographic Edition software is currently active. The data replication control-log messages are updated on the node where the data replication resource is currently `Online`.

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.

▼ How to Display Configuration Information About Partnerships

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

   You must be assigned the Basic Solaris User RBAC rights profile to complete this procedure. For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

2. **Display information about the partnership.**

   ```
   # geops list partnershipname
   ```

   partnershipname 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 Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

Example 8–1 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

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

   You must be assigned the Basic Solaris User RBAC rights profile to complete this procedure. For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

2. **Display information about a protection group.**

   ```
   # geopg list [protectiongroupname]
   ```

   protectiongroupname 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 8–2 Displaying Configuration Information About a Protection Group

This example displays configuration information for avspg, which is configured on cluster-paris.

```
# geopg list avspg
```

CHAPTER 9

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 105
- "Configuring a Protection Group to Run a Script at Switchover or Takeover" on page 107

Creating a Role-Change Action Script

You can configure the Geographic Edition software 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.

The path to this script should be valid on all nodes of all partner clusters that can host the protection group.

The following command-line runs the script:

```
# custom-action-command-path -o primary -c clustername \
-s partnershipname protectiongroupname userarguments
```

customactioncommandpath Specifies a path to the action command you have created.

`-o primary` Specifies that the role being assumed by the cluster is `primary`.

`-c` *clustername* Specifies the name of the secondary cluster that is assuming the new role of primary cluster.

-s *partnershipname*
: Specifies the name of the partnership that hosts the protection group.

protectiongroupname
: Specifies the name of the protection group that is undergoing the role change.

userarguments
: Specifies static arguments that are passed after all the Geographic Edition 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=sun.com,ip=10.1.2.3. You could also specify a sequence of options, such as -n sun.com -a 10.1.2.3.4. The format of these arguments is not restricted by the Geographic Edition software.

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 Geographic Edition software 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 9–1 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(1M) 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"
#
```

EXAMPLE 9–1 Switchover Action Script for Updating the DNS *(Continued)*

```
# 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 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see “Geographic Edition Software and RBAC” on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 **Configure the `RoleChange_ActionCmd` and `RoleChange_ActionArgs` properties of the protection group.**

```
# geopg set-prop -p RoleChange_ActionCmd=fullyqualifiedscript -p RoleChange_ActionArgs=scriptarguments
```

`-p` *propertysetting*
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 Geographic Edition Properties."

protectiongroupname
Specifies the name of the protection group.

Example 9–2 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=companyx.com,ip=1.2.3.4 avspg
```

CHAPTER 10

Script-Based Plug-Ins

This chapter provides information about Geographic Edition script-based plug-ins. It covers the following topics:

- "Overview of Geographic Edition Script-Based Plug-Ins" on page 109
- "Property Descriptions for Script-Based Plug-Ins" on page 113
- "Internals for Script-Based Plug-Ins" on page 123

Overview of Geographic Edition Script-Based Plug-Ins

Geographic Edition supports several replication technologies: Sun StorageTek Availability Suite software (AVS), Hitachi TrueCopy and Universal Replicator, and EMC Symmetrix Remote Data Facility (SRDF). However, the creation of these modules requires detailed knowledge of both the replication software and the internals of the Geographic Edition product. Geographic Edition uses the common agent container with a number of Java management beans (MBeans) that form the interface for the Geographic Edition monitoring and management infrastructure and the replication control software. For more information about the Sun StorageTek Availability Suite software (AVS), see Sun StorageTek Availability Suite software (AVS).

By providing a more generic interface module analogous to the Oracle Solaris ClusterGeneric Data Service (GDS), the Geographic Edition 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 Geographic Edition 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 *Oracle Solaris Cluster Reference Manual*.

For simplicity, the term script is used throughout this document to represent any compiled binary or script-based executable.

This section contains the following information:

- "Advantages and Disadvantages of Using Script-Based Plug-Ins" on page 110

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 Geographic Edition SRDF module has built-in arguments for the Symmetrix ID that the Geographic Edition browser interface prompts for separately, an equivalent script-based plug-in version would pass the value as one of a bundle of arguments to a script. The script would need to determine the arguments and whether each argument is valid.

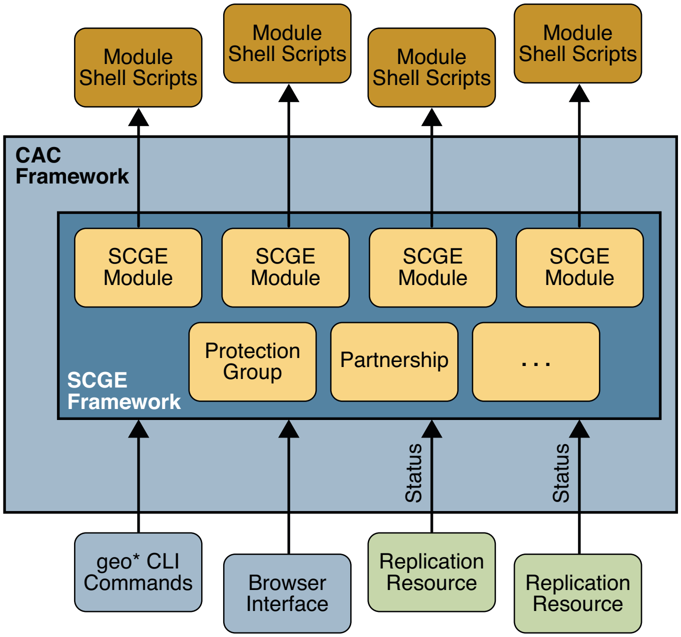
Script-Based Plug-In Architecture

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.

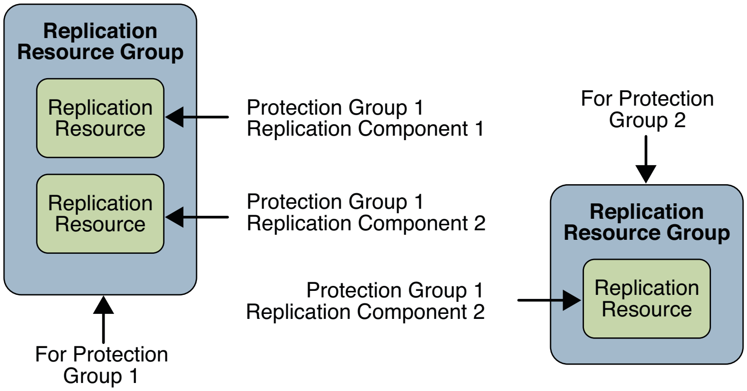
The following figure outlines the relationships between the various components within the Geographic Edition system. Commands issued through the command-line interface (CLI) or the browser interface call the Geographic Edition modules through their relevant common agent container modules. These modules 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 Geographic Edition output on the command line or in the browser interface.

FIGURE 10–1 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 group per protection group and a named replication resource per device group or replicated component.

FIGURE 10–2 Script-Based Plug-In Replication Resource Group


Replication Resource Group
Replication Resource
Replication Resource
Protection Group 1 Replication Component 1
Protection Group 1 Replication Component 2
For Protection Group 1
For Protection Group 2
Replication Resource Group
Protection Group 1 Replication Component 2
Replication Resource

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 Geographic Edition 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.

Ways to Create Script-Based Plug-Ins

The preferred method for creating a script-based plug-in module is to use the Generic Data Service (GDS) toolkit, which contains the extensions for the script-based plug-in.

Alternatively, 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.

Property Descriptions for Script-Based Plug-Ins

This section contains the following information:

- "Protection Group Properties - Overview" on page 113
- "Replicated Component Properties - Overview" on page 115
- "Protection Group Property Descriptions" on page 116

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 non-zero (failure) exit code. The script-based plug-in Mbean returns any error code resulting from a failure. For more information, see Appendix G, "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 10–1 Protection Group Global Policies

Property Name	Description	Type	Default Value
`add_app_rg_args`	The arguments that are provided to the script, `add_app_rg_script.`	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`

TABLE 10–1 Protection Group Global Policies *(Continued)*

`create_config_script`	The script used to create, modify, and validate a script-based plug-in replicated component instance.	Required	`/bin/false`
`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 Property Descriptions 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 Parameters explains how scripts can discriminate between the steps being performed. For more information, see “Protection Group Property Descriptions” on page 116 and “Standardized Script Command-Line Arguments” on page 125

Replicated Component Properties - Overview

Each replication component added to a particular protection group uses the scripts named in Protection Group Properties - Overview. Individual replications distinguish themselves by varying the properties passed to these scripts. The replication component script properties are listed in the following table. For more information, see "Protection Group Properties - Overview" on page 113

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 Geographic Edition Handles Password Properties" on page 126.

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 10–2 Optional Replicated Component Properties

Property Name	Description	Type
`create_config_args`	The arguments passed to the script named by the `create_config_script` protection group property.	Global
`remove_config_args`	The arguments passed to the script named by the `remove_config_script` protection group property.	Global
`start_replication_args`	The arguments passed to the script named by the `start_replication_script` protection group property.	Global
`stop_replication_args`	The arguments passed to the script named by the `stop_replication_script` protection group property.	Global
`switchover_args`	The arguments passed to the script named by the `switchover_script` protection group property.	Global
`takeover_args`	The arguments passed to the script named by the `takeover_script` protection group property.	Global

TABLE 10–2 Optional Replicated Component Properties *(Continued)*

`local_service_password`	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
`remote_service_password`	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

Protection Group Property Descriptions

This section describes the following protection group properties:

- "`add_app_rg_script` Property" on page 116
- "`configuration_file` Property" on page 117
- "`create_config_script` Property" on page 118
- "`remove_app_rg_script` Property" on page 120
- "`remove_config_script` Property" on page 120
- "`start_replication_script` Property" on page 121
- "`stop_replication_script` Property " on page 121
- "`switchover_script` Property" on page 122
- "`takeover_script` Property" on page 123

`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 Geographic Edition. 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=trueoption 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/addRGsand 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=pgName \
rgList=rg1,rg2,rg3,...
```

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 124. 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 124.

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:

SBP-configuration-name|*nodes-that-must-succeed-running-script*|*comma-separated-node-list*

For example:

```
foo.com|any|phys-node1,phys-node2
bar.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 foo.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 multi-node service like Oracle RAC.

For bar.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 multi-node 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-rgand 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-rgs*

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=
```

Examples of such internal resource groups are the lightweight resource groups in the AVS module or 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` `pg`command. 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=pgName 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=pgName\
    rgList=rg1,rg2,rg3,...
```

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 Geographic Edition 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=pgName
```

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=pgName
```

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 pgname      # local cluster only
```

For both clusters:

```
# geopg start -e global pgname     # 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=pgName
```

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 pgname      # local cluster only
```

For both clusters:

```
# geopg stop -e global pgname     # 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 switchover`command is executed. Subsequent changes in Geographic Edition 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=pgName  newRole=PRIMARY|SECONDARY
```

If that step succeeds:

```
# /var/tmp/switchover -u root -d /mydir \
    function=perform_switchover \
    validate_parameters=false \
    currentRole=PRIMARY|SECONDARY pg=pgName \
    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=pgName
```

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=pgName 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=pgName 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=pgName
```

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 124
- "Script-Based Plug-In Replication Resource Groups and Resources" on page 125
- "Protection Group Status Mapped from Replication Resource Status" on page 126
- "How Geographic Edition Handles Password Properties" on page 126

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 124
- "Standardized Script Command-Line Arguments" on page 125

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 Geographic Edition graphical user interface (GUI) 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 Geographic Edition 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 113 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 (see "`configuration_file` Property" on page 117). For more information, see "Protection Group Properties - Overview" on page 113.

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 `geopg validate` `pg name`
- When the Geographic Edition software is starting and re-creating 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-name> \
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 non-zero 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 Geographic Edition Handles Password Properties

This section describes the mechanism by which Geographic Edition 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 `geopg`) 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=foobar ...
```

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 `internalPasswordFile` argument is inserted into the properties list with the value *hostname*:*filename*.

Once in the core Geographic Edition Java code, the `internalPasswordFile` 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 Solaris Cluster CCR, if the rest of the properties are correct and complete, and the validation succeeds.

The passwords are only available from the CCR for users with root access. These passwords are also secure from casual users who might see the contents of the CCR displayed on the screen.

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.

APPENDIX A

Standard Geographic Edition Properties

This appendix provides the standard properties of Geographic Edition heartbeats, heartbeat plug-in, partnerships, protection groups, and data replication device groups.

This appendix contains the following sections:

Note – The property names and values, such as `Query_interval`, `True`, and `False`, are *not* case sensitive.

General Heartbeat Properties

The following table describes the heartbeat properties that the Geographic Edition software defines.

TABLE A–1 General Heartbeat Properties

Property Name	Description
`Query_interval` (integer)	Specifies the delay in seconds between heartbeat status requests. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Optional Default: 120 seconds

General Heartbeat Plug-in Properties

The following table describes the general heartbeat plug-in properties that the Geographic Edition software defines.

TABLE A–2 General Heartbeat Plug-in Properties

Property	Description
`Plugin_properties` (string)	Specifies a property string specific to the plug-in. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Optional Default: None except for heartbeats that use the default heartbeat plug-ins, `tcp_udp_plugin` and `ping-plugin`. For the `tcp_udp_plugin` plug-in, the format of this string is predefined as *remoteIPaddress*`/UDP/2084/ipsec`, *remoteIPaddress*`/TCP/2084/ipsec`. The *remote_IP_address* argument specifies the IP address of the partner cluster. The optional `ipsec` argument specifies if the plug-in uses IPsec with a Boolean value of `true` or `false`. For the `ping-plugin`, the format of this string is predefined as *remote_IP_address*, where *remote_IP_address* specifies the IP address of the partner cluster.
`Query_cmd` (string)	Specifies the path to the heartbeat status request command. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Required property if the plug-in does not specify a predefined plug-in. Default: None
`Requester_agent` (string)	Specifies the absolute path to the requester agent. Tuning recommendations: 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. Category: Optional Default: None

TABLE A–2 General Heartbeat Plug-in Properties *(Continued)*

Property	Description
`Responder_agent` (string)	Specifies the absolute path to the responder agent. Tuning recommendations: 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. Category: Optional Default: None
`Type` (enum)	Designates the type of plug-in. Set to either `primary` or `backup`. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Required Default: None, except for the default heartbeat that is named `ping_plugin`. If using this plug-in, the default value is `backup`.

Partnership Properties

The following table describes the partnership properties that the Geographic Edition software defines.

TABLE A–3 Partnership Properties

Property	Description
`Description` (string)	Describes the partnership. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Optional Default: Empty string
`Notification_ActionCmd` (string)	Provides the path to the action script that is triggered when heartbeat-loss notification is issued. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Optional Default: Empty string

TABLE A–3 Partnership Properties *(Continued)*

Property	Description
`Notification_EmailAddrs` (string array)	Lists the email addresses that are sent email when heartbeat-loss notification is issued. The list is comma delimited. Tuning recommendations: The value of this property is assigned at creation and can be tuned at runtime. Category: Optional Default: Empty string

General Properties of a Protection Group

The following table describes the protection group properties that the Geographic Edition software defines.

TABLE A–4 General Properties of a Protection Group

Property	Description
`Description` (string)	Describes the protection group. Tuning recommendations: This property can be tuned at any time. Category: Optional Default: Empty string
`External_Dependency_Allowed` (Boolean)	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 `true`. Tuning recommendations: This property can be tuned at any time. Category: Optional Default: `false`
`RoleChange_ActionArgs` (string)	Defines a string of arguments that are appended to the end of the command line when the role-change action command, `RoleChange_ActionCmd`, is run. Tuning recommendations: This property can be tuned at any time. Category: Optional Default: Empty string

TABLE A–4 General Properties of a Protection Group *(Continued)*

Property	Description
`RoleChange_ActionCmd` (string)	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. This path should be valid on all nodes of all partner clusters that can host the protection group. Tuning recommendations: This property can be tuned at any time. Category: Optional Default: Empty string

TABLE A–4 General Properties of a Protection Group *(Continued)*

Property	Description
`Timeout` (integer)	Specifies the timeout period for the protection group in seconds. The timeout period is the longest time Geographic Edition waits for a response after you run a `geopg` command, such as `geopg start`, `geopg stop`, `geopg switchover`, and `geopg takeover`. If the command does not respond within the timeout period, the Geographic Edition software reports the operation as timed out, even if the underlying command eventually completes successfully. 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. 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 "Troubleshooting Migration Problems" on page 144. 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. 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: ■ The local operation does not complete after the specified timeout period. ■ The remote operation does not complete after the specified timeout period. Tuning recommendations: This property can be tuned only when the protection group is offline. Category: Optional Range: 20-1000000 seconds Default: 200

APPENDIX B

Legal Names and Values of Geographic Edition Entities

This appendix lists the requirements for legal characters for the names and values of Geographic Edition entities.

This appendix contains the following sections:

Legal Names for Geographic Edition Entities

Geographic Edition 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 Geographic Edition Entities

The Geographic Edition 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

APPENDIX C

Disaster Recovery Administration Example

This appendix provides an example of a disaster recovery scenario and the actions an administrator might perform.

Company X 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 "Creating and Modifying a Partnership" on page 56.

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, Company X does not allow takeovers unless the primary cluster cannot be repaired within two hours.

   For information about verifying a disconnect on a system, see one of following data replication guides:

   - "Detecting Cluster Failure on a System That Uses Sun StorageTek Availability Suite Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
   - "Detecting Cluster Failure on a System That Uses Hitachi TrueCopy or Universal Replicator Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*

- "Detecting Cluster Failure on a System That Uses SRDF Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

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 one of the following data replication guides:

   - "Forcing a Takeover on Systems That Use Sun StorageTek Availability Suite" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
   - "Forcing a Takeover on a System That Uses Hitachi TrueCopy or Universal Replicator Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
   - "Forcing a Takeover on a System That Uses SRDF Data Replication" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

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 one of the following data replication guides:

   - "Recovering Sun StorageTek Availability Suite Data After a Takeover" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite*
   - "Recovering From a Hitachi TrueCopy or Universal Replicator Data Replication Error" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator*
   - "Recovering From an SRDF Data Replication Error" in *Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility*

APPENDIX D

Takeover Postconditions

This appendix provides details about the state of the primary and secondary clusters after you run the geopg takeover command.

This appendix contains the following sections:

- "Results of a Takeover When the Partner Cluster Can Be Reached" on page 139
- "Results of a Takeover When the Partner Cluster Cannot Be Reached" on page 140

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 D–1 Takeover Results of Running the geopg takeover Command on the Secondary Cluster

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
cluster-paris: primary, deactivated cluster-newyork: secondary, deactivated	cluster-paris: secondary, deactivated cluster-newyork: primary, deactivated
cluster-paris: primary, activated cluster-newyork: secondary, deactivated	cluster-paris: secondary, deactivated cluster-newyork: primary, deactivated

TABLE D–1 Takeover Results of Running the `geopg takeover` Command on the Secondary Cluster *(Continued)*

Cluster Role and State Before Takeover	Cluster Role and State After Takeover
`cluster-paris`: primary, deactivated `cluster-newyork`: secondary, activated	`cluster-paris`: secondary, deactivated `cluster-newyork`: primary, activated, with data replication stopped
`cluster-paris`: primary, activated `cluster-newyork`: secondary, activated	`cluster-paris`: secondary, deactivated `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 D–2 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-newyork`: secondary, deactivated	`cluster-paris`: primary, deactivated `cluster-newyork`: secondary, deactivated
	`cluster-paris`: primary, activated, with data replication stopped `cluster-newyork`: secondary, deactivated
`cluster-paris`: primary, activated `cluster-newyork`: secondary, deactivated	
`cluster-paris`: primary, deactivated `cluster-newyork`: secondary, activated	`cluster-paris`: primary, deactivated `cluster-newyork`: secondary, deactivated
`cluster-paris`: primary, activated `cluster-newyork`: secondary, activated	`cluster-paris`: primary, activated, with data replication stopped `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 D–3 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

`cluster-paris`: primary, deactivated, synchronization status `Unknown`
`cluster-newyork`: secondary, deactivated, synchronization status `Unknown`
	`cluster-paris`: primary, deactivated, synchronization status `Error`
	`cluster-newyork`: primary, deactivated, synchronization status `Error`

	`cluster-paris`: primary, activated, synchronization status `Error`
	`cluster-newyork`: primary, deactivated, synchronization status `Error`

`cluster-paris`: primary, activated, synchronization status `Unknown`
`cluster-newyork`: secondary, deactivated, synchronization status `Unknown`	

`cluster-paris`: primary, deactivated, synchronization status `Unknown`
`cluster-newyork`: secondary, activated, synchronization status `Unknown`
	`cluster-paris`: primary, deactivated, synchronization status `Error`
	`cluster-newyork`: primary, activated, with data replication stopped, synchronization status `Error`

`cluster-paris`: primary, activated, synchronization status `Unknown`
`cluster-newyork`: secondary, activated, synchronization status `Unknown`
	`cluster-paris`: primary, activated, synchronization status `Error`
	`cluster-newyork`: primary, activated, with data replication stopped, synchronization status `Error`

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 D–4 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

`cluster-paris`: primary, deactivated, synchronization status `Unknown`
`cluster-newyork`: secondary, deactivated, synchronization status `Unknown`
	`cluster-paris`: primary, deactivated, synchronization status `OK`, `Error`, or `Mismatch`
	`cluster-newyork`: secondary, deactivated, synchronization status `OK`, `Error`, or `Mismatch`

TABLE D–4 Takeover Results of Running the `geopg takeover` Command on the Primary Cluster When the Secondary Cluster Cannot Be Reached *(Continued)*

Cluster Role and State Before Takeover	Cluster Role and State After Takeover

`cluster-paris`: primary, activated, synchronization status `Unknown`
`cluster-newyork`: secondary, deactivated, synchronization status `Unknown`
	`cluster-paris`: primary, activated, with data replication stopped, synchronization status `OK`, `Error`, or `Mismatch`
	`cluster-newyork`: secondary, deactivated, synchronization status `OK`, `Error`, or `Mismatch`

`cluster-paris`: primary, deactivated, synchronization status `Unknown`
`cluster-newyork`: secondary, activated, synchronization status `Unknown`
	`cluster-paris`: primary, deactivated, synchronization status `OK`, `Error`, or `Mismatch`
	`cluster-newyork`: secondary, activated, synchronization status `OK`, `Error`, or `Mismatch`

`cluster-paris`: primary, activated, synchronization status `Unknown`
`cluster-newyork`: secondary, activated, synchronization status `Unknown`
	`cluster-paris`: primary, activated, with data replication stopped, synchronization status `OK`, `Error`, or `Mismatch`
	`cluster-newyork`: secondary, activated, synchronization status `OK`, `Error`, or `Mismatch`

APPENDIX E

Troubleshooting Geographic Edition Software

This appendix describes procedures for troubleshooting your application of the Geographic Edition software.

This appendix contains the following sections:

Troubleshooting Monitoring and Logging

This section provides information about setting up logging and problems that you might encounter with monitoring the Geographic Edition software.

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 Log 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 Log 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 Geographic Edition software.

Resource Groups Not Brought Online on Expected Node

Geographic Edition operations that bring resource groups online do so by passing the request to the Oracle Solaris Cluster framework and report the success of the request. When the cluster framework attempts to bring a resource group online on a cluster node, the cluster framework might encounter an error that would cause it to retry the operation on another node of the cluster. This might result in the following message being written to the log file:

```
resource group failed to start on chosen node; it may end up failing over to other node(s)
```

Solution or Workaround

If the Geographic Edition operation seems to have completed successfully, but the resource group did not come online as expected, review the log file and correct any underlying cluster problems.

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 Oracle Solaris Cluster infrastructure 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 *Oracle Solaris Cluster Data Services Planning and Administration Guide* about recovering from the `ERROR_STOP_FAILED` state and restarting the application.

The Geographic Edition software 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 Geographic Edition Software" on page 95.

Troubleshooting Cluster Start and Restart

This section provides information about troubleshooting problems that you might encounter with starting and restarting the Geographic Edition software.

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.

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.

Matching the `Nodelist` Property of a Protection Group to Those of Its Device Group and Resource Group

When you add resource groups, or Sun StorageTek Availability Suite device groups to a protection group, or when you run the command `geopg get` on a protection group, the order of the hosts in the `nodelist` property of each device group and resource group in the protection group must match the order of the hosts in the `nodelist` property of the protection group, or the operation will fail with a message similar to:

```
Application resource group app-rg must have a nodelist whose physical host components match those
of protection group app-pg and the resources it contains.
```

The Oracle Solaris Cluster Geographic Edition software requires that the entries in the `nodelist` property of a Sun StorageTek Availability Suite protection group match those of any device group or resource group added to the protection group. The order of the entries in their `nodelist` properties must also be identical.

Solution or Workaround

Ensure that the entries, and the order of the entries in the `nodelist` properties of a protection group, of its device groups, and of its resource groups are identical.

◆ ◆ ◆ APPENDIX F

Deployment Example: Replicating Data With MySQL

This appendix describes the MySQL data replication. It covers the following topics:

- “Overview of MySQL Replication ” on page 147
- “Initial Configuration of MySQL Replication” on page 149
- “Administering MySQL Protection Groups” on page 153
- “Creating, Modifying, Validating, and Deleting a MySQL Protection Group ” on page 154
- “Administering MySQL Application Resource Groups” on page 160
- “Administering MySQL Data-Replicated Components” on page 163
- “Replicating a MySQL Protection Group Configuration to a Partner Cluster” on page 167
- “Activating and Deactivating a MySQL Protection Group” on page 169
- “Resynchronizing a MySQL Protection Group” on page 172
- “Recovery Strategy After a Takeover of a MySQL Protection Group ” on page 173

Overview of MySQL Replication

This section provides an overview of the MySQL replication resource groups. A protection group that secures MySQL databases with MySQL replication consists of the following two resource groups securing a third resource group that is not part of the protection group on each cluster:

- “MySQL Database Resource Group” on page 148
- “MySQL Replication Resource Group” on page 148
- “MySQL Application Resource Group” on page 148

The MySQL database resource group holding a MySQL database is the foundation underneath the protection group. So, there must be strong positive dependencies with failover delegation should exist from the two resource groups in the protection group and the database resource group.

MySQL Database Resource Group

The MySQL database resource group typically contains the following resources:

- HAStoragePlus resource, which manages database storage
- Logical host resource, which provides the address to connect with the MySQL replication user
- MySQL database resource, which resides on top of the first two resources to make the database highly available locally

On single-node clusters, the HAStoragePlus resource can be omitted. The creation of the database resource group and its resources is the topic of *Oracle Solaris Cluster Data Service for MySQL Guide*.

The database resource group and its objects can have different names across the clusters.

MySQL Replication Resource Group

The MySQL replication resource group contains the MySQL replication resource. This resource does not start or stop any process. Its only purpose is to monitor the status of the MySQL database replication.

MySQL Application Resource Group

The MySQL application resource group must contain at least a logical host resource, which provides the address for all the clients to use for connections to the database.

The MySQL application resource group also contains any application that depends on the MySQL database in the cluster. For example, if an HA for Apache Tomcat application server is configured in the cluster, its failover resource is added to the MySQL application resource group, assuming that the servlet that is deployed in Apache Tomcat uses for database access the logical host that is in the MySQL application resource group. This resource group is added to the protection group, which then ensures that clients connect only to the master database and not to the slave database. Connection of the application resource group to the slave database instead of the master database might compromise data consistency.

Initial Configuration of MySQL Replication

This section provides examples of the following tasks:

- "Installing MySQL and Configuring the MySQL Database Resource Group" on page 149
- "Configuring the MySQL Application Resource Group" on page 152

Installing MySQL and Configuring the MySQL Database Resource Group

You can install MySQL and configure both clusters as described in *Oracle Solaris Cluster Data Service for MySQL Guide*.

The database resource group and its resources do not have to have the same name on both clusters.

Requirements for Geographic Edition controlling MySQL are as follows:

- Do not use the bind-address keyword in the MySQL configuration file `my.cnf`.
- Specify the nodes of all clusters in the `mysql_config` file for database preparation.

When preparing the MySQL database for cluster usage with the `mysql_register` script, you must provide all the physical node names or zone names of your clusters in the variable: `MYSQL_NIC_HOSTNAME`. For example:

```
MYSQL_NIC_HOSTNAME="cl1-phys1,cl1,phys2,cl2-phys3,cl2-phys4"
```

When you are configuring the database resource, keep the following restrictions in mind:

- The configuration of the MySQL database resource as a Service Management Facility (SMF) component on top of a failover container resource is not allowed.
- The MySQL replication between the two clusters must be the only MySQL replication configured in the two databases.
- The MySQL databases on both clusters must be configured to listen on the same port.

▼ How to Configure the MySQL Replication

Before You Begin Before you can configure the replication, you must decide which cluster will contain the master database at the first start.

1 **Prevent the startup of the slave threads.**

On the master cluster at the node where the MySQL database is active, add the `skip-slave-start` keyword to the `my.cnf` file. For example:

```
cl1-node1 # echo skip-slave-start >> /mysql-data-directory/my.cnf
```

2 **Prevent non-superuser modifications.**

On the slave cluster at the node where the MySQL database is active, add the `read-only=true` directive to the `my.cnf` file, and restart your database.

For example:

```
cl2-node1 # echo read-only=true >> /mysql-data-directory/my.cnf
cl2-node1 # clresource disable mys-rs
cl2-node1 # clresource enable mys-rs
```

3 **Create the replication user on both databases.**

a. **On each cluster, pick the node where the MySQL database is active, and connect as an administrative user who can at least create users.**

b. **Create the replication user, and stay connected.**

Note – Be sure to create the replication with permissions to connect from any node.

- The following example assumes that the MySQL database on the primary cluster listens to the socket `/tmp/nyc.sock`:

```
cl1-node1:/ # /usr/local/mysql/bin/mysql -S /tmp/nyc.sock -uroot -proot
mysql> use mysql
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl1-node1' identified by 'repl';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl1-node2' identified by 'repl';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl2-node3' identified by 'repl';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl2-node4' identified by 'repl';
```

- The following example assumes that the MySQL database on the secondary cluster listens to the socket `/tmp/sfo.sock`:

```
cl2-node3:/ # /usr/local/mysql/bin/mysql -S /tmp/sfo.sock -uroot -proot
mysql> use mysql
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl1-node1' identified by 'repl';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl1-node2' identified by 'repl';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl2-node3' identified by 'repl';
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'cl2-node4' identified by 'repl';
```

4 **Establish the replication between the secondary and primary clusters.**

a. **On the primary cluster, issue the following on the MySQL client:**

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> show master status;
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB  |

| bin-log.000002 |     1424 |              | sc3_test_database |

1 row in set (0.03 sec)

mysql> unlock tables;
```

Note the values for file and position. In the preceding example, they are `bin-log.000002` and 1424, respectively.

b. **On the MySQL client on the slave, issue the following commands:**

```
mysql> change master to master_host='nyc',

    -> master_user='repl',

    -> master_password='repl',

    -> master_log_file='bin-log.000002',

    -> master_log_pos=1424;

Query OK, 0 rows affected (0.04 sec)

mysql> start slave;

Query OK, 0 rows affected (0.03 sec)
```

c. **Check the slave status.**

```
mysql> show slave status;
...
    Check for the following messages:
            Slave_IO_State: Waiting for master to send event
            Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
...
```

d. **Stop the slave.**

```
mysql> stop slave;
```

5 **Configure the reverse replication to prepare the two clusters for a role swap.**

a. **On the secondary cluster, issue the following commands:**

```
mysql> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.01 sec)
mysql> show master status;

| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB  |

| bin-log.000020 |     1162 |              | sc3_test_database |

1 row in set (0.00 sec)

mysql> unlock tables;
```

Note the values for file and position. In the preceding example, they are `bin-log.000020` and `1162`, respectively.

b. **On the MySQL client on the primary cluster, issue the following commands:**

```
mysql> change master to master_host='sfo',

    -> master_user='repl',
    -> master_password='repl',

    -> master_log_file='bin-log.000020',

    -> master_log_pos=1162;

mysql> start slave;

Query OK, 0 rows affected (0.03 sec)
```

c. **Check the slave status.**

```
mysql> show slave status;
...
    Check for the following messages:
            Slave_IO_State: Waiting for master to send event
            Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
...
```

d. **Stop the slave, and exit the MySQL client.**

```
mysql> stop slave;
mysql> exit;
```

e. **On the MySQL client on the secondary cluster, start the slave, and exit the client.**

```
mysql> start slave;
mysql> exit;
```

Configuring the MySQL Application Resource Group

At a minimum, you must create a resource group that contains a logical-host resource. You must leave the resource group in an unmanaged state.

For example, assume the following configuration:

- nyc-rg is the resource group on cluster nyc that contains a MySQL resource.
- sfo-rg is the resource group on cluster sfo that contains a MySQL resource.
- usa-rg is the application resource group you are adding to the MySQL protection group.

On cluster nyc, you would issue the following commands:

```
cl1-node1 # clresourcegroup create usa-rg
cl1-node1 # clresourcegroup set -p Auto_start_on_new_cluster=false  usa-rg
cl1-node1 # clresourcegroup set -p RG_Affinities=+++nyc-rg  usa-rg
cl1-node1 # clreslogicalhostname create -g usa-rg usa
cl1-node1 # clresource enable usa
```

On cluster sfo, you would issue the following commands:

```
cl2-node1 # clresourcegroup create usa-rg
cl2-node1 # clresourcegroup set -p Auto_start_on_new_cluster=false  usa-rg
cl2-node1 # clresourcegroup set -p RG_Affinities=+++sfo-rg  usa-rg
cl2-node1 # clreslogicalhostname create -g usa-rg usa
cl2-node1 # clresource enable usa
```

Administering MySQL Protection Groups

A MySQL protection group must cover at least one MySQL database per cluster. It cannot protect anything other than MySQL databases. So, if your partnership contains additional replication protocols, you must create separate protection groups for them.

Planning for Your MySQL Protection Group

A MySQL database resource group can belong to only one protection group.

The MySQL geographic replication was developed with the script-based plug-in module of Geographic Edition, so it must comply with all rules of the script-based plug-in. For each protection group, you must provide a script-based plug-in configuration file on each node. In addition, the MySQL geographic replication brings in its own configuration file, which is needed only at registration.

The MySQL geographic replication creation is an automated process that takes the MySQL geographic configuration file as input and performs the necessary actions. The essential content of this file consists of `key=`*value* pairs.

Key	Explanation of Value
`PS`	Name of the partnership.
`PG`	Name of the protection group to create.
`REPCOMP`	Name of the replicated component to create in this protection group.
`REPRS`	Name of the replication resource.
`REPRG`	Name of the replication resource group.
`DESC`	Description for the protection group.
`CONFIGFILE`	Absolute path for the script-based plug-in configuration file.
`REALMYSRG`	Resource group names that contain the MySQL database resource on the clusters. If the resource group names on the clusters differ, provide a comma-separated list.

REALMYSRS	Resource names configured as the master and slave MySQL database resources. If the resource names on the clusters differ, provide a comma-separated list.
READONLY	Switch for setting the read-only variable on the MySQL slave. If the read-only variable should not be set, leave this value undefined. Any entry here triggers the read-only variable to be set.
AAPRG	Application resource group, which is unmanaged and contains at least the logical host for client access.
LONGPING	Timeout for the extensive `ping` test. The default is 60 seconds if this variable is unset. This timeout is used at `check_takeover` where it must be verified as the remote site is unavailable.
SHORTPING	Timeout for the short `ping` test. The default is 10 seconds if this variable is unset. The short `ping` timeout is used whenever a connection should succeed but is not required to succeed.

Creating, Modifying, Validating, and Deleting a MySQL Protection Group

Protection group names are unique in the global Geographic Edition namespace. You cannot use the same protection group name in two partnerships on the same system. In addition, you can replicate the existing configuration of a protection group from a remote cluster to a local cluster. For more information, see "Replicating a MySQL Protection Group Configuration to a Partner Cluster" on page 167.

This section contains the following information:

- "How to Create the MySQL Configuration" on page 154
- "Modifying a MySQL Protection Group" on page 156
- "Validating a MySQL Protection Group" on page 157
- "Data Replication Layer Process for Validating the Application Resource Groups and Data Replication Entities" on page 157
- "How to Delete a MySQL Protection Group" on page 158

▼ How to Create the MySQL Configuration

Perform this procedure from a node of the primary cluster.

Before You Begin Ensure that the following conditions are met:

- The local cluster is a member of a partnership.
- The protection group you are creating does not already exist, if you are going to create the MySQL configuration.
- The protection group exists if you want to do anything other than create the MySQL configuration.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 **Create the content for the `mysql_geo_config` file.**

3 **Copy the `mysql_geo_config` file to a different location.**

For example:

```
cl1-phys-node1 # cp /opt/SUNWscmys/geocontrol/util/mysql_geo_config /temp
```

4 **Specify the following variables in `/temp/mysql_geo_config`. This list uses sample values.**

```
PS=mysql-ps
PG=mysql-pg
REPCOMP=mysql.sbp
REPRS=mysql-rep-rs
REPRG=mysql-rep-rg
DESC="mysql replication pg"
CONFIGFILE=/geo-config/sbpconfig
REALMYSRG=nyc-rg,sfo-rg
REALMYSRS=nyc-mys-rs,sfo-mys-rs
READONLY=
APPRG=usa-rg
LONGPING=
SHORTPING=
```

5 **Create the script-based plug-in configuration file on all nodes of all clusters.**

For example, assuming that the nodes of cluster one are `cl1-phys-node1` and `cl1-phys-node2`, on each node of cluster one, you would issue the following commands:

```
cl1-phys-node1 # mkdir /geo-config
cl1-phys-node1 # echo "mysql.sbp|any|cl1-phys-node1,cl1-phys-node2">/geo-config/sbpconfig
```

Assuming that the nodes of cluster two are `cl2-phys-node3` and `cl2-phys-node4`, on each node of cluster two, you would issue the following commands:

```
cl2-phys-node1 # mkdir /geo-config
cl2-phys-node1 # echo "mysql.sbp|any|cl2-phys-node3,cl2-phys-node4">/geo-config/sbpconfig
```

6 **Execute the `mysql_geo_register` script on the primary cluster.**

For example:

```
cl1-phys-node1 # ksh  /opt/SUNWscmys/geocontrol/util/mysql_geo_register -f  /temp/mysql_geo_config
```

7 **Replicate the protection group to the partner cluster.**

The final messages of the registration script outline the required `geopg get` command. You must log in to one node of the partner cluster and execute that exact command.

For example:

```
cl2-phys-node3 # geopg get --partnership mysql-ps mysql-pg
```

Next Steps Go to "Activating a MySQL Protection Group" on page 169.

Modifying a MySQL Protection Group

If the partner cluster contains a protection group with the same name, the `geopg set-prop`command also propagates the new configuration information to the partner cluster.

Use the following command to modify a MySQL protection group:

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

In this syntax, `-p` *property* specifies the properties of the protection group, and *protection-group* specifies the name of the protection group.

For more information about the properties you can set, see "Property Descriptions for Script-Based Plug-Ins" on page 113.

The `geopg set-prop`command revalidates the protection group with the new configuration information. If the validation is unsuccessful on the local cluster, the configuration of the protection group is not modified. Otherwise, the configuration status is set to OK on the local cluster.

If the configuration status is OK on the local cluster, but the validation is unsuccessful on the partner cluster, the configuration status is set to Error on the partner cluster.

For information about the names and values that are supported, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geopg` command, refer to the `geopg`(1M) man page.

EXAMPLE F–1 Modifying the Timeout Property of a Protection Group

The following example shows how to modify the timeout property of a protection group.

```
# geopg set-prop -p Timeout=300 mysql-pg
```

Validating a MySQL Protection Group

When the configuration status of a protection group is displayed as Error in the `geoadm` status output, you can validate the configuration by using the `geopg validate`command. This command checks the current status of the protection group and its entities.

If the protection group and its entities are valid, then the configuration status of the protection groups is set to OK. If the `geopg validate`command finds an error in the configuration files, then the command displays an error message, and the configuration remains in the Error state. In such a case, you can fix the error in the configuration, and rerun the `geopg validate`command.

The `geopg validate`command validates the configuration of the protection group on the local cluster only. To validate the protection group configuration on the partner cluster, rerun the command on the partner cluster.

Before validating the configuration of a protection group, ensure that the protection group you want to validate exists locally and that the common agent container is online on all nodes of both clusters in the partnership.

1. Validate the configuration of the protection group.

   Use the following command to validate the configuration of a protection group on the local cluster only:

   ```
    # geopg validate protection-group
   ```

   In this syntax, *protection-group* specifies a unique name that identifies a single protection group.

   In the following example, the configuration of a protection group is validated:

   ```
   # geopg validate mysql-pg
   ```

Data Replication Layer Process for Validating the Application Resource Groups and Data Replication Entities

During protection group validation, the MySQL data replication layer validates the application resource groups and the data replication entities by verifying that an application resource group in the protection group has its `Auto_start_on_new_cluster` property set to `false`.

When you bring a protection group online on the primary cluster, bring the application resources groups participating in that protection group online only on the same primary cluster. Setting the `Auto_start_on_new_cluster` property to `false` prevents the Oracle Solaris Cluster resource group manager from automatically starting the application resource groups. In this case, the startup of resource groups is reserved for the Geographic Edition software.

Application resource groups should be online only on the primary cluster when the protection group is activated.

The MySQL `geocontrol` module supplies a script that is used by the script-based plug-in module. The script entry points require the same set of arguments. These arguments are validated for semantics and completeness. The following validation checks are performed:

- Are all of the mandatory arguments defined?
- Is the configured MySQL database resource defined?
- Is the specified replication resource configured with a correct start command, if the resource exists already?
- Are the long and short `ping` intervals numeric?

When the validation is complete, the Geographic Edition software creates and brings online the replication resource group and its resources if they don't already exist. If a resource group or resource of the same name already exists, the Geographic Edition software might modify its properties. The software cannot create a new resource group or a resource of the same name if one already exists. After creating the necessary resources, the software adds the application resource group to the protection group.

▼ How to Delete a MySQL Protection Group

Perform this procedure from a node on the cluster where you want to delete the protection group, for example, `cluster-nyc`. The `cluster-nyc` cluster is the primary cluster. For a sample cluster configuration, see "Example Geographic Edition Cluster Configuration" on page 33.

To delete a protection group on all clusters, run the `geopg delete` command on each cluster where the protection group exists.

Before You Begin Before deleting a protection group, ensure that the following conditions are met:

- The protection group exists locally.
- The protection group is offline on the local cluster.

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

---

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the /var/cluster/geo ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

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
```

In this syntax, *protection-group* specifies the name of the protection group.

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

Example F–2 Deleting a Protection Group

In the following example, a protection group is deleted from both partner clusters:

```
 # rlogin cluster-nyc -l root
cluster-nyc# geopg delete mysql-pg

 # rlogin cluster-sfo -l root
cluster-sfo# geopg delete mysql-pg
```

Example F–3 Deleting a Protection Group While Keeping Application Resource Groups Online

In the following example, two application resource groups (apprg1 and apprg2) are kept online while the protection group that they share, mysql-pg, is deleted. First, the application resource groups are removed from the protection group. Then, the protection group is deleted.

```
 # geopg remove-resource-group apprg1,apprg2 mysql-pg
 # geopg stop -e global mysql-pg
 # geopg delete mysql-pg
```

Administering MySQL Application Resource Groups

To make an application highly available, the application must be managed as a resource in an application resource group.

The initial registration of the protection group is performed with the `mysql_geo_register` script. This section explains how to manage the application resource groups on their own.

All of the entities that you configure for the application resource group on the primary cluster, such as application data resources, application configuration files, and resource groups, must be replicated manually on the secondary cluster. The resource group names must be identical on both clusters. Also, the data that the application resource uses must be replicated on the secondary cluster.

This section describes the following tasks:

- "How to Add an Application Resource Group to a MySQL Protection Group" on page 160
- "How to Delete an Application Resource Group From a MySQL Protection Group" on page 162

▼ How to Add an Application Resource Group to a MySQL Protection Group

You can add an existing application resource group to the list of application resource groups for a protection group.

Before You Begin Before you add an application resource group to a protection group, ensure that the following conditions are met:

- The protection group is defined.
- The resource group to be added already exists on both clusters and is in an appropriate state.
- The `Auto_start_on_new_cluster` property of the resource group is set to `false`. You can view this property by using the `clresourcegroup` show command. For example:

  ```
  # clresourcegroup show -p Auto_start_on_new_cluster apprg1
  ```

  You can set the `Auto_start_on_new_cluster` property to `false` as follows:

  ```
  # clresourcegroup set -p Auto_start_on_new_cluster=false apprg1
  ```

  Setting the `Auto_start_on_new_cluster` property to `false` prevents the Oracle Solaris Cluster resource group manager from automatically starting the resource groups in the protection group. Once the Geographic Edition software restarts and communicates with the remote cluster to ensure that the remote cluster is running and that the remote cluster is the secondary cluster for that resource group, the software will not automatically start the resource group on the primary cluster.

Application resource groups should be online only on the primary cluster when the protection group is activated.

- The `Nodelist` property of the failover application resource group that has affinities with a replicated component defined by the resource must contain the same entries in identical order as the `Nodelist` property of the protection group.
- The application resource group must not have dependencies on resource groups and resources outside of this protection group. To add several application resource groups that share dependencies, you must add all the application resource groups that share dependencies to the protection group in a single operation. If you add the application resource groups separately, the operation will fail.

  The protection group can be activated or deactivated, and the resource group can be either online or unmanaged. If the resource group is unmanaged and the protection group is activated after the configuration of the protection group has changed, then the local state of the protection group becomes Error. If the resource group to add is online, and the protection group is deactivated, the request is rejected. You must activate the protection group before adding an online resource group.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2 **Add an application resource group to the protection group.**

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

In this syntax, *resource-group-list* specifies the name of the application resource group. You can specify more than one resource group in a comma-separated list. Also, *protection-group* specifies the name of the protection group.

This command adds an application resource group to a protection group on the local cluster. Then, the command propagates the new configuration information to the partner cluster if the partner cluster contains a protection group with the same name.

For information about the names and values that are supported by Geographic Edition software, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

If the add operation is unsuccessful on the local cluster, the configuration of the protection group is not modified. Otherwise, the configuration status is set to OK on the local cluster. If the configuration status is OK on the local cluster, but the add operation is unsuccessful on the partner cluster, the configuration status is set to Error on the partner cluster.

After the application resource group is added to the protection group, the application resource group is managed as an entity of the protection group. Then, the application resource group is affected by protection group operations such as start, stop, switchover, and takeover.

Example F–4 Adding an Application Resource Group to a MySQL Protection Group

In the following example, two application resource groups, `apprg1` and `apprg2`, are added to `mysql-pg`:

```
# geopg add-resource-group apprg1,apprg2 mysql-pg
```

▼ How to Delete an Application Resource Group From a MySQL Protection Group

You can remove an application resource group from a protection group without altering the state or contents of the application resource group.

Before You Begin Ensure that the following conditions are met:

- The protection group is defined on the local cluster.
- The resource group to be removed is part of the application resource groups of the protection group.

1. **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

   For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

   Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

2. **Remove the application resource group from the protection group.**

   The following command removes an application resource group from a protection group on the local cluster. If the partner cluster contains a protection group with the same name, the application resource group is also removed from the protection group on the partner cluster.

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

In this syntax, *resource-group-list* specifies the name of the application resource group. You can specify more than one resource group in a comma-separated list. In addition, *protection-group* specifies the name of the protection group.

If the resource group that is being removed shares dependencies with other resource groups in the protection group, then you must also remove all other resource groups that share dependencies with the resource group that is being removed.

If the remove operation fails on the local cluster, the configuration of the protection group is not modified. Otherwise, the configuration status is set to OK on the local cluster. If the configuration status is OK on the local cluster but the remove operation is unsuccessful on the partner cluster, the configuration status is set to Error on the partner cluster.

Example F–5 Deleting an Application Resource Group From a Protection Group

In the following example, two application resource groups, `apprg1` and `apprg2`, are removed from `mysql-pg`.

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

Administering MySQL Data-Replicated Components

This section describes the following tasks for administering data-replicated components in a MySQL protection group:

- "How to Add a Data-Replicated Component to a MySQL Protection Group" on page 163
- "Data Replication Subsystem Process for Verifying the Replicated Component" on page 165
- "How to Modify a MySQL Data-Replicated Component " on page 166
- "How to Delete a Data-Replicated Component From a MySQL Protection Group" on page 166

For details about configuring a MySQL protection group, see "How to Create the MySQL Configuration" on page 154.

▼ How to Add a Data-Replicated Component to a MySQL Protection Group

A protection group is the container for the application resource groups, which contain data for services that are protected from disaster. The Geographic Edition software protects the data by replicating it from the primary cluster to the secondary cluster. By adding a data-replicated component to a protection group, the software monitors the replication status of a MySQL database. The software also controls the role and state of the database during protection group operations such as start, stop, switchover, and takeover.

Before You Begin Before you add a replication component to a protection group, 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 underlying MySQL database resources exist on both the local cluster and the partner cluster.

1 Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 Add a data-replicated component to the protection group.

The following command adds a replication component to a protection group on the local cluster and propagates the new configuration to the partner cluster if the partner cluster contains a protection group with the same name.

```
# geopg add-replication-component -p property [-p...]  MySQL-replicated-component protection-group
```

In this syntax, `-p` *property* specifies the properties of the data-replicated component group. You can specify the following script-based plug-in properties:

- `switchover_args` — Specifies the command-line arguments for the switchover script
- `takeover_args` — Specifies the command-line arguments for the takeover script
- `start_replication_args` — Specifies the command-line arguments for the `start_replication` script
- `remove_config_args` — Specifies the command-line arguments for the `remove_configuration` script
- `create_config_args` — Specifies the command-line arguments for the `create_configuration` script
- `stop_replication_args` — Specifies the command-line arguments for the `stop_replication` script

Caution – Make sure that the command-line arguments are the same for all scripts.

For more information about the properties you can set, see "Property Descriptions for Script-Based Plug-Ins" on page 113.

Also in this syntax, *MySQL-replicated-component* specifies the name of the new data-replicated component, and `protection-group` specifies the name of the protection group that will contain the new data-replicated component.

For information about the names and values that are supported, see Appendix B, "Legal Names and Values of Geographic Edition Entities."

For more information about the `geopg` command, refer to the `geopg`(1M) man page.

Note – Because the add operation for the replication component is performed during the scripted registration, an example is not provided here.

Data Replication Subsystem Process for Verifying the Replicated Component

During protection group validation, the MySQL data replication layer validates the application resource groups and the data replication entities by verifying that an application resource group in the protection group has its `Auto_start_on_new_cluster` property set to `false`.

When you bring a protection group online on the primary cluster, bring the application resources groups participating in that protection group online only on the same primary cluster. Setting the `Auto_start_on_new_cluster` property to `false` prevents the Oracle Solaris Cluster resource group manager from automatically starting the application resource groups. In this case, the startup of resource groups is reserved for the Geographic Edition software.

Application resource groups should be online only on the primary cluster when the protection group is activated.

The `Mysql geocontrol` module supplies a script that is used by the script-based plug-in module. The script entry points require the same set of arguments. These arguments are validated for semantics and completeness. The following validation checks are performed:

- Are all of the mandatory arguments defined?
- Is the configured MySQL database resource defined?
- Is the specified replication resource configured with a correct start command, if the resource exists already?
- Are the long and short `ping` intervals numeric?

When the validation is complete, the Geographic Edition software adds the application resource group to the protection group.

Note – Every entry point of the underlying script-based plug-in has a validation method. In the case of the MySQL replication, all the validation methods are the same.

▼ How to Modify a MySQL Data-Replicated Component

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 **Modify the replication component.**

The following command modifies the properties of a device group in a protection group on the local cluster. Then, the command 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...] MySQL-replicated-component protection-group
```

In this syntax, `-p` *property* specifies the properties of the data-replicated component.

For more information about the properties you can set, see "Property Descriptions for Script-Based Plug-Ins" on page 113.

Also in this syntax, *MySQL-replicated-component* specifies the name of the data-replicated component, and *protectiongroupname* specifies the name of the protection group that will contain the new data-replicated component.

▼ How to Delete a Data-Replicated Component From a MySQL Protection Group

You might need to delete a data-replicated component from a protection group if you previously added a data-replicated component to that protection group. Normally, after an application is configured to connect to the database, you would not change the database.

Before You Begin Before you delete a data-replicated component, 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 device group is managed by the protection group.

For information about deleting protection groups, refer to "How to Delete a MySQL Protection Group" on page 158.

1 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

2 **Remove the replicated component.**

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

In this syntax, *MySQL-replicated-component* specifies the name of the data-replicated component, and *protection-group* specifies the name of the protection group.

Example F–6 Deleting a Replicated Component From a MySQL Protection Group

In the following example, a data-replicated device group is deleted from a `MySQLprotection` group:

```
# geopg remove-replication-component mysql-dg mysql-pg
```

Replicating a MySQL Protection Group Configuration to a Partner Cluster

Before you replicate the configuration of a MySQL protection group to a partner cluster, ensure that the following conditions are met:

- The protection group is defined on the remote cluster, not on the local cluster.
- The MySQL database resources in the protection group on the remote cluster exist on the local cluster.

- The application resource groups in the protection group on the remote cluster exist on the local cluster.
- The `Auto_start_on_new_cluster` property of the resource groups is set to `false`. You can view this property by using the `clresourcegroup` show command, as follows:

  ```
  # clresourcegroup show -p auto_start_on_new_cluster apprg
  ```

  Then, set the `Auto_start_on_new_cluster` property to `false` as follows:

  ```
   # clresourcegroup set -p Auto_start_on_new_cluster=false apprg1
  ```

  Setting this property to `false` prevents the Oracle Solaris Cluster resource group manager from automatically starting the resource groups in the protection group. Once the Geographic Edition software restarts and communicates with the remote cluster to ensure that the remote cluster is running and that the remote cluster is the secondary cluster for the resource group, the software does not automatically start the resource group on the primary cluster.

  Application resource groups should be online only on the primary cluster when the protection group is activated.

1. Replicate the protection group configuration to the partner cluster.

   Use the following command to retrieve the configuration information of the protection group from the remote cluster and creates the protection group on the local cluster.

   ```
   # geopg get -s partnershipname  MySQL-Protection-group
   ```

   In this syntax, `-s` *partnershipname* specifies the name of the partnership from which the protection group configuration information is retrieved. In addition, *MySQL-Protection-group* specifies the name of the protection group.

   Note – The `geopg get`command replicates Geographic Edition related entities. For more information, see "Replicating and Upgrading Configuration Data for Resource Groups, Resource Types, and Resources" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

EXAMPLE F–7 Replicating a MySQL Protection Group to a Partner Cluster

In the following example, the configuration of `mysql-pg` is replicated to `cluster-sfo`.

The configuration of the protection group is retrieved from the remote cluster (in this example, `cluster-nyc`) and then validated by the data replication subsystem on the local cluster, `cluster-sfo`.

If the validation is successful, the configuration status is set to OK, and the protection group is created on the local cluster. This protection group contains a replicated component and an application group that are configured almost identically to the replicated component and application group on the remote cluster. If the validation fails, the protection group is not created on the local cluster. Fix the cause of the error, and replicate it again.

EXAMPLE F–7 Replicating a MySQL Protection Group to a Partner Cluster (Continued)

```
# rlogin cl2-phys-1 -l root
cl2-phys-1# geopg get -s nyc-sfo-ps mysql-pg
```

Activating and Deactivating a MySQL Protection Group

This section describes the following tasks:

- "Activating a MySQL Protection Group" on page 169
- "Deactivating a MySQL Protection Group" on page 170

When you activate a protection group, it assumes the role that you assigned to it during configuration. For more information about configuring protection groups, see "How to Create the MySQL Configuration" on page 154.

Activating a MySQL Protection Group

You can activate a protection group in the following ways:

- Globally, which activates a protection group on both clusters where the protection group has been configured
- On the primary cluster only
- On the secondary cluster only

1. Activate the protection group.

   Use the following command to activate 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.

   ```
   # geopg start -e scope [-n] MySQL-Protection-group
   ```

   In this syntax, `-e` *scope* specifies the scope of the command. If the scope is local, 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, such as global and local, are not case sensitive.

   Also in this syntax, `-n` prevents the start of data replication at protection group startup. 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 non-superusers if this option is configured
- Prepares the `my.cnf` file to start the database with modifications prevented for non-superusers if this option is configured

Also in this syntax, `MySQL-Protection-group` specifies the name of the protection group.

The `geopg start`command uses the `clresourcegroup online -M` command to bring resource groups and resources online. For more information, see the clrg(1M) man page.

The `geopg start`command performs the following actions if the role of the protection group is primary on the local cluster:

- 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 to obtain the status of your system.

  If the configuration status is set to Error, revalidate the protection group by using the procedure described in "Validating a MySQL Protection Group" on page 157.

EXAMPLE F–8 Globally Activating a MySQL Protection Group

In the following example, a protection group is globally activated:

```
# geopg start -e global mysql-pg
```

EXAMPLE F–9 Locally Activating a MySQL Protection Group

In the following example, a protection group is activated on a local cluster only. This local cluster might be a primary cluster or a secondary cluster, depending on the role of the cluster.

```
# geopg start -e local mysql-pg
```

Deactivating a MySQL Protection Group

You can deactivate a protection group in the following ways:

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

1. Deactivate the protection group.

Use the following command to deactivate 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.

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

In this syntax, `-e` *scope* specifies the scope of the command. If the scope is local, 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.

Also in this syntax, `-D` specifies that only data replication should be stopped and that the protection group should be online. If you omit this option, the data 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.

In addition, `protection-group` specifies the name of the protection group.

If the `geopg stop`command fails, run the `geoadm status`command to obtain the status of each 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 procedure described in "Validating a MySQL Protection Group" on page 157.

EXAMPLE F–10 Deactivating a MySQL Protection Group on All Clusters

In the following example, a protection group is deactivated on all clusters:

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

EXAMPLE F–11 Deactivating a MySQL Protection Group on a Local Cluster

In the following example, a protection group is deactivated on the local cluster:

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

EXAMPLE F–12 Stopping MySQL Data Replication While Leaving the Protection Group Online

In the following example, data replication is stopped on the local cluster only:

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

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

EXAMPLE F–12 Stopping MySQL Data Replication While Leaving the Protection Group Online *(Continued)*

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

EXAMPLE F–13 Deactivating a MySQL Protection Group While Keeping Application Resource Groups Online

In the following example, two application resource groups, `apprg1` and `apprg2`, are kept online while their protection group, `mysql-pg`, is deactivated. First, the application resource groups are removed from the protection group. Then, the protection group is deactivated.

```
# geopg remove-resource-group apprg1,apprg2 mysql-pg
 # geopg stop -e global mysql-pg
```

Resynchronizing a MySQL Protection Group

You can resynchronize the configuration information for 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 is Error.

For example, you might need to resynchronize protection groups after booting the cluster. For more information, see "Booting a Cluster" on page 42.

Resynchronizing a protection group only updates entities that are related to Geographic Edition. For more information, see "Replicating and Upgrading Configuration Data for Resource Groups, Resource Types, and Resources" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*

The protection group must be deactivated on the cluster where you intend to run the `geopg update`command.

Use the following command to resynchronize the protection group.

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

In this syntax, *protection-group* specifies the name of the protection group.

EXAMPLE F–14 Resynchronizing a MySQL Protection Group

In the following example, a protection group is resynchronized:

```
# geopg update mysql-pg
```

Recovery Strategy After a Takeover of a MySQL Protection Group

When an old primary cluster is restarting for the first time after a successful takeover, the MySQL database does not detect that the cluster should no longer act as a master and the Geographic Edition software still keeps the primary role, but leaves it deactivated. The goal for the recovery is to configure the old master to run as a slave and to update the Geographic Edition software configuration to reflect this role change.

You can check for the status with the following command:

```
# geoadm status
```

The recovery strategy after a takeover involves the following:

1. Configuring the old master to run as a slave
2. Manually starting the slave threads on the old master
3. Resynchronizing the protection group to switch the role

▼ How to Recover After a Takeover

1 **Allow the MySQL slave threads to be started if the database resource performs a restart or similar action.**

Remove the `skip-slave-start` keyword from the appropriate `my.cnf` file.

2 **Become superuser or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see "Geographic Edition Software and RBAC" on page 47.

Note – If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, become superuser 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 RBAC rights profile and Oracle Data Guard.

---

3 **Log in to MySQL as superuser, then start the slave.**

```
mysql> start slave;
```

4 **Verify that the slave is running, and wait until it is synchronized with the master.**

```
mysql> show slave status\G
```

If the slave status shows that at least one slave thread is not running, fix the root cause, and retry the operation. As a last resort, you could take a backup from the current master and perform a fresh slave setup.

Connect to a node of the old primary cluster and update the protection group to change the role from a deactivated primary cluster to a secondary cluster.

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

6 **Resynchronize the protection group.**

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

7 **Start the protection group locally.**

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

For more information, see "Resynchronizing a MySQL Protection Group" on page 172 and "Activating and Deactivating a MySQL Protection Group" on page 169.

◆ ◆ ◆ APPENDIX G

Error Return Codes for Script-Based Plug-Ins

Error Return Codes for 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 non-zero 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 `rglist=`*comma separated resource groups*
127	`E_SBP_NO_SUCH_FILE`	An attempt was made to execute a null or non-existent 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.
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 node list.
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.
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