



Sun Cluster 3.1 Data Service Planning and Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-1526
May 2003 Revision A

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



030314@5533



Contents

Preface	7
1 Planning for Sun Cluster Data Services	13
Sun Cluster Data Services Installation and Configuration Tasks	13
Configuration Guidelines for Sun Cluster Data Services	14
Identifying Data Service Special Requirements	15
Determining the Location of the Application Binaries	15
Verifying the <code>nsswitch.conf</code> File Contents	16
Planning the Cluster File System Configuration	16
Relationship Between Resource Groups and Disk Device Groups	16
Understanding <code>HASStorage</code> and <code>HASStoragePlus</code>	17
Determining Whether Your Data Service Requires <code>HASStorage</code> or <code>HASStoragePlus</code>	18
Choosing Between <code>HASStorage</code> and <code>HASStoragePlus</code>	19
Considerations	19
Node List Properties	19
Overview of the Installation and Configuration Process	20
Installation and Configuration Task Flow	21
Example	21
Tools for Data Service Resource Administration	22
The SunPlex Manager Graphical User Interface (GUI)	22
The Sun Cluster Module for the Sun Management Center GUI	23
The <code>scsetup</code> Utility	23
The <code>scrgadm</code> Command	23
Data Service Resource Administration Tasks	23
Sun Cluster Data Service Fault Monitors	24

Fault Monitor Invocation 24

2	Administering Data Service Resources	27
	Administering Data Service Resources	28
	Configuring and Administering Sun Cluster Data Services	30
	Registering a Resource Type	31
	▼ How to Register a Resource Type	31
	Upgrading a Resource Type	32
	▼ How to Install and Register an Upgrade of a Resource Type	32
	▼ How to Migrate Existing Resources to a New Version of the Resource Type	33
	Creating a Resource Group	37
	▼ How to Create a Failover Resource Group	37
	▼ How to Create a Scalable Resource Group	38
	Adding Resources to Resource Groups	40
	▼ How to Add a Logical Hostname Resource to a Resource Group	41
	▼ How to Add a Shared Address Resource to a Resource Group	42
	▼ How to Add a Failover Application Resource to a Resource Group	44
	▼ How to Add a Scalable Application Resource to a Resource Group	46
	Bringing Resource Groups Online	48
	▼ How to Bring a Resource Group Online	48
	Disabling and Enabling Resource Monitors	49
	▼ How to Disable a Resource Fault Monitor	50
	▼ How to Enable a Resource Fault Monitor	50
	Removing Resource Types	51
	▼ How to Remove a Resource Type	51
	Removing Resource Groups	52
	▼ How to Remove a Resource Group	52
	Removing Resources	54
	▼ How to Remove a Resource	54
	Switching the Current Primary of a Resource Group	55
	▼ How to Switch the Current Primary of a Resource Group	55
	Disabling Resources and Moving Their Resource Group Into the UNMANAGED State	56
	▼ How to Disable a Resource and Move Its Resource Group Into the UNMANAGED State	57
	Displaying Resource Type, Resource Group, and Resource Configuration Information	58

▼ How to Display Resource Type, Resource Group, and Resource Configuration Information	59
Changing Resource Type, Resource Group, and Resource Properties	59
▼ How to Change Resource Type Properties	60
▼ How to Change Resource Group Properties	61
▼ How to Change Resource Properties	62
Clearing the STOP_FAILED Error Flag on Resources	63
▼ How to Clear the STOP_FAILED Error Flag on Resources	63
Re-registering Preregistered Resource Types	65
▼ How to Re-register Preregistered Resource Types	65
Adding or Removing a Node to or From a Resource Group	66
▼ How to Add a Node to a Resource Group	66
▼ How to Remove a Node From a Resource Group	69
Synchronizing the Startups Between Resource Groups and Disk Device Groups	74
▼ How to Set Up HAStorage Resource Type for New Resources	74
▼ How to Set Up HAStorage Resource Type for Existing Resources	76
Upgrading from HAStorage to HAStoragePlus	77
How to Upgrade from HAStorage to HAStoragePlus When Using Device Groups or CFS	77
How to Upgrade from HAStorage With CFS to HAStoragePlus With Failover Filesystem	78
Enabling Highly Available Local File Systems	79
▼ How to Set Up HAStoragePlus Resource Type	80
Freeing Node Resources by Offloading Non-critical Resource Groups	82
▼ How to Set Up an RGOffload Resource	82
Configuring RGOffload Extension Properties	84
Fault Monitor	86
A Standard Properties	89
Resource Type Properties	89
Resource Properties	96
Resource Group Properties	109
Resource Property Attributes	116
B Legal RGM Names and Values	119
RGM Legal Names	119
RGM Values	120

Index 121

Preface

The Sun Cluster 3.1 Data Service Planning and Administration Guide contains procedures to install and configure the Sun Cluster data services.

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

The instructions in this document assume knowledge of the Solaris™ operating environment and expertise with the volume manager software used with Sun Cluster.

UNIX Commands

This document contains information on commands specific to installing and configuring Sun Cluster data services. It might not contain information on basic UNIX® commands and procedures, such as shutting down the system, booting the system, and configuring devices. For that information, see one or more of the following:

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

Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type <code>rm filename</code> .

Shell Prompts

Shell	Prompt
C shell	<i>machine_name</i> %
C shell superuser	<i>machine_name</i> #
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Related Documentation

Application	Title	Part Number
Installation	<i>Sun Cluster 3.1 Software Installation Guide</i>	816-3388
Data Services Installation and Configuration	<i>Sun Cluster 3.1 Data Service Planning and Administration Guide</i>	817-1526
	<i>Sun Cluster 3.1 Data Service for Apache</i>	817-1532
	<i>Sun Cluster 3.1 Data Service for BroadVision One-To-One Enterprise</i>	817-1542
	<i>Sun Cluster 3.1 Data Service for Domain Name Service (DNS)</i>	817-1533
	<i>Sun Cluster 3.1 Data Service for Netbackup</i>	817-1538
	<i>Sun Cluster 3.1 Data Service for Network File System (NFS)</i>	817-1534
	<i>Sun Cluster 3.1 Data Service for Oracle</i>	817-1527
	<i>Sun Cluster 3.1 Data Service for Oracle Parallel Server/Real Application Clusters</i>	817-1535
	<i>Sun Cluster 3.1 Data Service for SAP</i>	817-1536
	<i>Sun Cluster 3.1 Data Service for SAP liveCache</i>	817-1539
	<i>Sun Cluster 3.1 Data Service for Siebel</i>	817-1540
	<i>Sun Cluster 3.1 Data Service for Sun ONE Application Server</i>	817-1530
	<i>Sun Cluster 3.1 Data Service for Sun ONE Directory Server</i>	817-1529
	<i>Sun Cluster 3.1 Data Service for Sun ONE Message Queue</i>	817-1531
	<i>Sun Cluster 3.1 Data Service for Sun ONE Web Server</i>	817-1528
	<i>Sun Cluster 3.1 Data Service for Sybase ASE</i>	817-1537
<i>Sun Cluster 3.1 Data Service for DHCP</i>	817-1716	
<i>Sun Cluster 3.1 Data Service for Samba</i>	817-1715	
<i>Sun Cluster 3.1 Data Service for WebSphere MQ</i>	817-1714	
<i>Sun Cluster 3.1 Data Service for WebSphere MQ Integrator</i>	817-1713	

Application	Title	Part Number
Hardware	<i>Sun Cluster 3.x Hardware Administration Manual</i>	817-0168
	Sun Cluster 3.x Data Service Collection at http://docs.sun.com/db/coll/1024.1	
API development	<i>Sun Cluster 3.1 Data Services Developer's Guide</i>	816-3385
Administration	<i>Sun Cluster 3.1 5/03 System Administration Guide</i>	816-3384
Cluster concepts	<i>Sun Cluster 3.1 Concepts Guide</i>	816-3383
Error Messages	<i>Sun Cluster 3.1 Error Messages Guide</i>	816-3382
Man Pages	<i>Sun Cluster 3.1 Reference Manual</i>	816-5251
Release notes	<i>Sun Cluster 3.1 Data Service 5/03 Release Notes</i>	817-1790
	<i>Sun Cluster 3.1 Release Notes</i>	816-5317
	<i>Sun Cluster 3.1 Release Notes Supplement</i>	816-3381

Sun Documentation Online

The docs.sun.comSM Web site enables you to access Sun technical documentation on the Web. You can browse the docs.sun.com archive or search for a specific book title or subject at <http://docs.sun.com>.

Help

If you have problems installing or using Sun Cluster, contact your service provider and provide the following information:

- Your name and E-mail address (if available)
- Your company name, address, and phone number
- The model and serial numbers of your systems
- The release number of the operating environment (for example, Solaris 8)
- The release number of Sun Cluster (for example, Sun Cluster 3.0)

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

Command	Function
<code>prtconf -v</code>	Displays the size of the system memory and reports information about peripheral devices.
<code>psrinfo -v</code>	Displays information about processors.
<code>showrev -p</code>	Reports which patches are installed.
<code>prtdiag -v</code>	Displays system diagnostic information.
<code>scinstall -pv</code>	Displays Sun Cluster release and package version information.

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

Planning for Sun Cluster Data Services

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

- “Sun Cluster Data Services Installation and Configuration Tasks” on page 13
- “Configuration Guidelines for Sun Cluster Data Services” on page 14
- “Relationship Between Resource Groups and Disk Device Groups” on page 16
- “Understanding HAStorage and HAStoragePlus” on page 17
- “Considerations” on page 19
- “Node List Properties” on page 19
- “Overview of the Installation and Configuration Process” on page 20
- “Tools for Data Service Resource Administration” on page 22
- “Sun Cluster Data Service Fault Monitors” on page 24

See the *Sun Cluster 3.1 Concepts Guide* document for conceptual information about data services, resource types, resources, and resource groups.

If your applications are not currently offered as Sun Cluster data services, see the *Sun Cluster 3.1 Data Services Developer’s Guide* for information on how to develop other applications to become highly available data services.

Sun Cluster Data Services Installation and Configuration Tasks

The following table lists the books that describe the installation and configuration of Sun Cluster data services.

TABLE 1-1 Task Map: Installing and Configuring Sun Cluster Data Services

Task	For Instructions, Go To ...
Install and configure Sun Cluster HA for Oracle	<i>Sun Cluster 3.1 Data Service for Oracle</i>
Install and configure Sun Cluster HA for Sun Open Net Environment (Sun ONE) Web Server	<i>Sun Cluster 3.1 Data Service for Sun ONE Web Server</i>
Install and configure Sun Cluster HA for Sun ONE Directory Server	<i>Sun Cluster 3.1 Data Service for Sun ONE Directory Server</i>
Install and configure Sun Cluster HA for Apache	<i>Sun Cluster 3.1 Data Service for Apache</i>
Install and configure Sun Cluster HA for DNS	<i>Sun Cluster 3.1 Data Service for Domain Name Service (DNS)</i>
Install and configure Sun Cluster HA for NFS	<i>Sun Cluster 3.1 Data Service for Network File System (NFS)</i>
Install and configure Sun Cluster Support for Oracle Parallel Server/Real Application Clusters	<i>Sun Cluster 3.1 Data Service for Oracle Parallel Server/Real Application Clusters</i>
Install and configure Sun Cluster HA for SAP	<i>Sun Cluster 3.1 Data Service for SAP</i>
Install and configure Sun Cluster HA for Sybase ASE	<i>Sun Cluster 3.1 Data Service for Sybase ASE</i>
Install and configure Sun Cluster HA for BroadVision One-To-One Enterprise	<i>Sun Cluster 3.1 Data Service for BroadVision One-To-One Enterprise</i>
Install and configure Sun Cluster HA for NetBackup	<i>Sun Cluster 3.1 Data Service for NetBackup</i>
Install and configure Sun Cluster HA for SAP liveCache	<i>Sun Cluster 3.1 Data Service for SAP liveCache</i>
Install and configure Sun Cluster HA for Siebel	<i>Sun Cluster 3.1 Data Service for Siebel</i>

Configuration Guidelines for Sun Cluster Data Services

This section provides configuration guidelines for Sun Cluster data services.

Identifying Data Service Special Requirements

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

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

Determining the Location of the Application Binaries

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

- **The local disks of each cluster node** – Placing the software and configuration files on the individual cluster nodes provides the following advantage. You can upgrade application software later without shutting down the service.
The disadvantage is that you then have several copies of the software and configuration files to maintain and administer.
- **The cluster file system** – If you put the application binaries on the cluster file system, you have only one copy to maintain and manage. However, you must shut down the data service in the entire cluster to upgrade the application software. If you can spare a small amount of downtime for upgrades, place a single copy of the application and configuration files on the cluster file system.
See the planning chapter of the *Sun Cluster 3.1 Software Installation Guide* for information on how to create cluster file systems.
- **Highly available local file system** - Using `HASStoragePlus`, you can integrate your local file system into the Sun Cluster environment making the local file system highly available. `HASStoragePlus` provides additional file system capabilities such as checks, mounts, and unmounts enabling Sun Cluster to fail over local file systems. In order to failover, the local file system must reside on global disk groups with affinity switchovers enabled.
See the individual data service chapters or “Enabling Highly Available Local File Systems” on page 79 for information on how to use the `HASStoragePlus` resource type.

Verifying the `nsswitch.conf` File Contents

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

- which databases within the Solaris environment to use for name-service lookups
- in what order to consult the databases

Some data services require that you direct “group” lookups to “files” first. For these data services, change the “group” line in the `nsswitch.conf` file so that the “files” entry is listed first. See the chapter for the data service that you plan to configure to determine whether you need to change the “group” line.

See the planning chapter in the *Sun Cluster 3.1 Software Installation Guide* for additional information on how to configure the `nsswitch.conf` file for the Sun Cluster environment.

Planning the Cluster File System Configuration

Depending on the data service, you might need to configure the cluster file system to meet Sun Cluster requirements. See the chapter for the data service that you plan to configure to determine whether any special considerations apply.

The resource type `HASStoragePlus` enables you to use a highly available local file system in a Sun Cluster environment configured for failover. See “Enabling Highly Available Local File Systems” on page 79 for information on setting up the `HASStoragePlus` resource type.

See the planning chapter of the *Sun Cluster 3.1 Software Installation Guide* for information on how to create cluster file systems.

Relationship Between Resource Groups and Disk Device Groups

Sun Cluster uses the concept of **node lists** for disk device groups and resource groups. Node lists are ordered lists of primary nodes, which are potential masters of the disk device group or resource group. Sun Cluster uses a **failback policy** to determine what happens when a node has been down and then rejoins the cluster, and the rejoining node appears earlier in the node list than the current primary node. If failback is set to `True`, the device group or resource group will be switched off of the current primary and switched onto the rejoining node, making the rejoining node the new primary.

To ensure high availability of a failover resource group, make the resource group's node list match the node list of associated disk device groups. For a scalable resource group, the resource group's node list cannot always match the device group's node list because, currently, a device group's node list must contain exactly two nodes. For a greater-than-two-node cluster, the node list for the scalable resource group can have more than two nodes.

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

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

See the *Sun Cluster 3.1 Software Installation Guide* for information on how to set up disk device groups. See the *Sun Cluster 3.1 Concepts Guide* document for more details on the relationship between disk device groups and resource groups.

Understanding HAStorage and HAStoragePlus

The `HAStorage` and the `HAStoragePlus` resource types can be used to configure the following options.

- Coordinate the boot order of disk devices and resource groups by causing the `START` methods of the other resources in the same resource group that contains the `HAStorage` or `HAStoragePlus` resource to wait until the disk device resources become available.
- With `AffinityOn` set to `True`, enforce colocation of resource groups and disk device groups on the same node, thus enhancing the performance of disk-intensive data services.

In addition, `HAStoragePlus` is capable of mounting any global file system found to be in an unmounted state. See "Planning the Cluster File System Configuration" on page 16 for more information.

Note – If the device group is switched to another node while the `HASStorage` or `HASStoragePlus` resource is online, `AffinityOn` has no effect and the resource group does **not** migrate along with the device group. On the other hand, if the resource group is switched to another node, `AffinityOn` being set to `True` causes the device group to follow the resource group to the new node.

See “Synchronizing the Startups Between Resource Groups and Disk Device Groups” on page 74 for information about the relationship between disk device groups and resource groups. The `SUNW.HASStorage(5)` and `SUNW.HASStoragePlus(5)` man pages provides additional details.

See “Enabling Highly Available Local File Systems” on page 79 for procedures for mounting of file systems such as VxFS in a local mode. The `SUNW.HASStoragePlus(5)` man page provides additional details.

Determining Whether Your Data Service Requires `HASStorage` or `HASStoragePlus`

- In cases where a data service resource group has a node list in which some of the nodes are not directly connected to the storage, you must configure `HASStorage` or `HASStoragePlus` resources in the resource group and set the dependency of the other data service resources to the `HASStorage` or `HASStoragePlus` resource. This requirement coordinates the boot order between the storage and the data services.
- If your data service is disk intensive, such as Sun Cluster HA for Oracle and Sun Cluster HA for NFS, ensure that you perform the following tasks.
 - Add a `HASStorage` or `HASStoragePlus` resource to your data service resource group.
 - Switch the `HASStorage` or `HASStoragePlus` resource online.
 - Set the dependency of your data service resources to the `HASStorage` or `HASStoragePlus` resource.
 - Set `AffinityOn` to `True`.

When you perform these tasks, the resource groups and disk device groups are collocated on the same node.
 - The failback settings must be identical for both the resource group and device group(s).
- If your data service is **not** disk intensive—such as one that reads all of its files at startup (for example, Sun Cluster HA for DNS)—configuring the `HASStorage` or `HASStoragePlus` resource type is optional.

Choosing Between HAStorage and HAStoragePlus

To determine whether to create HAStorage or HAStoragePlus resources within a data service resource group, consider the following criteria.

- Use HAStorage if you are using Sun Cluster 3.0 12/01 or earlier.
- Use HAStoragePlus if you are using Sun Cluster 3.0 5/02 or Sun Cluster 3.1. (If you want to integrate any file system locally into a Sun Cluster configured for failover, you must upgrade to Sun Cluster 3.0 5/02 or Sun Cluster 3.1 and use the HAStoragePlus resource type. See “Planning the Cluster File System Configuration” on page 16 for more information.)

Considerations

Use the information in this section to plan the installation and configuration of any data service. The information in this section encourages you to think about the impact your decisions have on the installation and configuration of any data service. For specific considerations for your data service, see the chapter in Sun Cluster 3.1 Data Service Planning and Administration Guide that applies to your specific data service.

- When using data services that are I/O intensive and that have a large number of disks configured in the cluster, the application may experience delays due to retries within the I/O subsystem during disk failures. An I/O subsystem may take several minutes to retry and recover from a disk failure. This delay can result in Sun Cluster failing over the application to another node, even though the disk may have eventually recovered on its own. To avoid failover during these instances, consider increasing the default probe timeout of the data service. If you need more information or help with increasing data service timeouts, contact your local support engineer.
- For better performance, Install and configure your data service on the cluster nodes with direct connection to the storage.

Node List Properties

You can specify the following three node lists when configuring data services.

1. `installed_nodes` – A property of the resource type. This property is a list of the cluster node names on which the resource type is installed and enabled to run.

2. `nodelist` – A property of a resource group that specifies a list of cluster node names where the group can be brought online, in order of preference. These nodes are known as the potential primaries or masters of the resource group. For failover services, configure only one resource group node list. For scalable services, configure two resource groups and thus two node lists. One resource group and its node list identifies the nodes on which the shared addresses are hosted. This list is a failover resource group on which the scalable resources depend. The other resource group and its list identifies nodes on which the application resources are hosted. The application resources depend on the shared addresses. Therefore, the node list for the resource group that contains the shared addresses must be a superset of the node list for the application resources.
3. `auxnodelist` – A property of a shared address resource. This property is a list of physical node IDs that identify cluster nodes that can host the shared address but never serve as primary in the case of failover. These nodes are mutually exclusive with the nodes identified in the node list of the resource group. This list pertains to scalable services only. See the `scrgadm(1M)` man page for details.

Overview of the Installation and Configuration Process

Use the following procedures to install and configure data services.

- Install the data service packages from the Sun Cluster Agents CD-ROM.
- Install and configure the application to run in the cluster environment.
- Configure the resources and resource groups that the data service uses. When you configure a data service, specify the resource types, resources, and resource groups that the Resource Group Manager (RGM) will manage. The chapters for the individual data services describe these procedures.

Before you install and configure data services, see the *Sun Cluster 3.1 Software Installation Guide*, which includes procedures on how to install the data service software packages and how to configure Internet Protocol Network Multipathing (IP Networking Multipathing) groups that the network resources use.

Note – You can use SunPlex Manager to install and configure the following data services: Sun Cluster HA for Oracle, Sun Cluster HA for Sun ONE Web Server, Sun Cluster HA for Sun ONE Directory Server, Sun Cluster HA for Apache, Sun Cluster HA for DNS, and Sun Cluster HA for NFS. See the SunPlex Manager online help for more information.

Installation and Configuration Task Flow

The following table shows a task map of the procedures to install and configure a Sun Cluster failover data service.

TABLE 1-2 Task Map: Sun Cluster Data Service Installation and Configuration

Task	For Instructions, Go to
Install the Solaris and Sun Cluster software	<i>Sun Cluster 3.1 Software Installation Guide</i>
Set up IP Networking Multipathing groups	<i>Sun Cluster 3.1 Software Installation Guide</i>
Set up multihost disks	<i>Sun Cluster 3.1 Software Installation Guide</i>
Plan resources and resource groups	<i>Sun Cluster 3.1 Release Notes</i>
Decide the location for application binaries, and configure the <code>nsswitch.conf</code> file	Chapter 1
Install and configure the application software	The chapter for each data service in this book
Install the data service software packages	<i>Sun Cluster 3.1 Software Installation Guide</i> or the chapter for each data service in this book
Register and configure the data service	The chapter for each data service in this book

Example

The example in this section shows how you might set up the resource types, resources, and resource groups for an Oracle application that has been instrumented to be a highly available failover data service.

The main difference between this example and an example of a scalable data service is that, in addition to the failover resource group that contains the network resources, a scalable data service requires a separate resource group (called a scalable resource group) for the application resources.

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

Because this example is a failover data service, the example uses logical hostname network resources, which are the IP addresses that fail over from a primary node to a secondary node. Place the logical hostname resources into a failover resource group, and then place the Oracle server resources and listener resources into the same resource group. This ordering enables all of the resources to fail over as a group.

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

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

Tools for Data Service Resource Administration

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

The SunPlex Manager Graphical User Interface (GUI)

SunPlex Manager is a web-based tool that enables you to perform the following tasks.

- Install a cluster.
- Administer a cluster.
- Create and configure resources and resource groups.
- Configure data services with the Sun Cluster software.

See the *Sun Cluster 3.1 Software Installation Guide* for instructions on how to use SunPlex Manager to install cluster software. SunPlex Manager provides online help for most administrative tasks.

The Sun Cluster Module for the Sun Management Center GUI

The Sun Cluster module enables you to monitor clusters and to perform some operations on resources and resource groups from the Sun Management Center GUI. See the *Sun Cluster 3.1 Software Installation Guide* for information about installation requirements and procedures for the Sun Cluster module. Go to <http://docs.sun.com> to access the Sun Management Center software documentation set, which provides additional information about Sun Management Center.

The `scsetup` Utility

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

The `scrgadm` Command

You can use the `scrgadm` command to register and configure data service resources. See the procedure on how to register and configure your data service in the applicable chapter of this book. If, for example, you use Sun Cluster HA for Oracle, see “Installing and Configuring Sun Cluster HA for Oracle” in *Sun Cluster 3.1 Data Service for Oracle*. Chapter 2 also contains information on how to use the `scrgadm` command to administer data service resources. Finally, see the `scrgadm(1M)` man page for additional information.

Data Service Resource Administration Tasks

The following table lists which tool you can use in addition to the command line for different data service resource administration tasks. See Chapter 2 for more information about these tasks and for details on how to use the command line to complete related procedures.

TABLE 1-3 Tools You Can Use for Data Service Resource Administration Tasks

Task	SunPlex Manager	Sun Management Center	The <code>scsetup</code> Utility
Register a resource type	Yes	No	Yes

TABLE 1-3 Tools You Can Use for Data Service Resource Administration Tasks
(Continued)

Task	SunPlex Manager	Sun Management Center	The <code>scsetup</code> Utility
Create a resource group	Yes	No	Yes
Add a resource to a resource group	Yes	No	Yes
Bring a resource group online	Yes	Yes	No
Remove a resource group	Yes	Yes	No
Remove a resource	Yes	Yes	No
Switch the current primary of a resource group	Yes	No	No
Disable a resource	Yes	Yes	No
Move the resource group of a disabled resource into the unmanaged state	Yes	No	No
Display resource type, resource group, and resource configuration information	Yes	Yes	No
Change resource properties	Yes	No	No
Clear the <code>STOP_FAILED</code> error flag on resources	Yes	No	No
Add a node to a resource group	Yes	No	No

Sun Cluster Data Service Fault Monitors

This section provides general information about data service fault monitors. The Sun-supplied data services contain fault monitors that are built into the package. The fault monitor (or fault probe) is a process that probes the health of the data service.

Fault Monitor Invocation

The RGM invokes the fault monitor when you bring a resource group and its resources online. This invocation causes the RGM to internally call the `MONITOR_START` method for the data service.

The fault monitor performs the following two functions.

- monitors the abnormal exit of the data service server process or processes
- checks the health of the data service

Monitoring of the Abnormal Exit of the Server Process

The Process Monitor Facility (PMF) monitors the data service processes.

The data service fault probe runs in an infinite loop and sleeps for an adjustable amount of time that the resource property `Thorough_probe_interval` sets. While sleeping, the probe checks with the PMF to see if the process has exited. If the process has exited, the probe updates the status of the data service as “Service daemon not running” and takes action. The action can involve restarting the data service locally or failing over the data service to a secondary cluster node. To decide whether to restart or to fail over the data service, the probe checks the value set in the resource properties `Retry_count` and `Retry_interval` for the data service application resource.

Checking the Health of the Data Service

Typically, communication between the probe and the data service occurs through a dedicated command or a successful connection to the specified data service port.

The logic that the probe uses is roughly as follows.

1. Sleep (`Thorough_probe_interval`).
2. Perform health checks under a time-out property `Probe_timeout`.
`Probe_timeout` is a resource extension property of each data service that you can set.
3. If Step 2 is a success, that is, the service is healthy, update the success/failure history. To update the success/failure history, purge any history records that are older than the value that is set for the resource property `Retry_interval`. The probe sets the status message for the resource as “Service is online” and returns to Step 1.

If Step 2 resulted in a failure, the probe updates the failure history. The probe then computes the total number of times that the health check failed.

The result of the health check can range from a complete failure to success. The interpretation of the result depends on the specific data service. Consider a scenario where the probe can successfully connect to the server and send a handshake message to the server, but the probe receives only a partial response before it times out. This scenario is most likely a result of system overload. If some action is taken (such as restarting the service), the clients reconnect to the service, thus further overloading the system. If this event occurs, a data service fault monitor can decide not to treat this “partial” failure as fatal. Instead, the monitor can track this failure as a nonfatal probe of the service. These partial failures are still accumulated over the interval that the `Retry_interval` property specifies.

However, if the probe cannot connect to the server at all, the failure can be considered fatal. Partial failures lead to incrementing the failure count by a fractional amount. Every time the failure count reaches total failure (either by a fatal failure or by accumulation of partial failures), the probe restarts or fails over the data service in an attempt to correct the situation.

4. If the result of the computation in Step 3 (the number of failures in the history interval) is less than the value of the resource property `Retry_count`, the probe attempts to correct the situation locally (for example, by restarting the service). The probe sets the status message of the resource as "Service is degraded" and returns to Step 1.
5. If the number of failures in `Retry_interval` exceeds `Retry_count`, the probe calls `scha_control` with the "giveover" option. This option requests failover of the service. If this request succeeds, the fault probe stops on this node. The probe sets the status message for the resource as, "Service has failed."
6. The Sun Cluster framework can deny the `scha_control` request issued in the previous step for various reasons. The return code of `scha_control` identifies the reason. The probe checks the return code. If the `scha_control` is denied, the probe resets the failure/success history and starts afresh. This probe resets the history because the number of failures is already above `Retry_count`, and the fault probe would attempt to issue `scha_control` in each subsequent iteration (which would be denied again). This request would place additional load on the system and would increase the likelihood of further service failures.

The probe then returns to Step 1.

Administering Data Service Resources

This chapter describes how to use the `scrgadm(1M)` command to manage resources, resource groups, and resource types within the cluster. See “Tools for Data Service Resource Administration” on page 22 to determine if you can use other tools to complete a procedure.

This chapter contains the following procedures.

- “How to Register a Resource Type” on page 31
- “How to Install and Register an Upgrade of a Resource Type” on page 32
- “How to Migrate Existing Resources to a New Version of the Resource Type” on page 33
- “How to Create a Failover Resource Group” on page 37
- “How to Create a Scalable Resource Group” on page 38
- “How to Add a Logical Hostname Resource to a Resource Group” on page 41
- “How to Add a Shared Address Resource to a Resource Group” on page 42
- “How to Add a Failover Application Resource to a Resource Group” on page 44
- “How to Add a Scalable Application Resource to a Resource Group” on page 46
- “How to Bring a Resource Group Online” on page 48
- “How to Disable a Resource Fault Monitor” on page 50
- “How to Enable a Resource Fault Monitor” on page 50
- “How to Remove a Resource Type” on page 51
- “How to Remove a Resource Group” on page 52
- “How to Remove a Resource” on page 54
- “How to Switch the Current Primary of a Resource Group” on page 55
- “How to Disable a Resource and Move Its Resource Group Into the UNMANAGED State” on page 57
- “How to Display Resource Type, Resource Group, and Resource Configuration Information” on page 59
- “How to Change Resource Type Properties” on page 60
- “How to Change Resource Group Properties” on page 61
- “How to Change Resource Properties” on page 62
- “How to Clear the STOP_FAILED Error Flag on Resources” on page 63
- “How to Re-register Preregistered Resource Types” on page 65
- “How to Add a Node to a Resource Group” on page 66

- “How to Remove a Node From a Resource Group” on page 69
- “How to Set Up HAStorage Resource Type for New Resources” on page 74
- “How to Set Up HAStorage Resource Type for Existing Resources” on page 76
- “How to Set Up HAStoragePlus Resource Type” on page 80
- “How to Set Up an RGOffload Resource” on page 82

See Chapter 1 and the *Sun Cluster 3.1 Concepts Guide* document for overview information about resource types, resource groups, and resources.

Administering Data Service Resources

Table 2–1 lists the sections that describe the administration tasks for data service resources.

TABLE 2–1 Task Map: Data Service Administration

Task	For Instructions, Go To ...
Register a resource type	“How to Register a Resource Type” on page 31
Upgrade a resource type	“How to Migrate Existing Resources to a New Version of the Resource Type” on page 33 “How to Install and Register an Upgrade of a Resource Type” on page 32
Create failover or scalable resource groups	“How to Create a Failover Resource Group” on page 37 “How to Create a Scalable Resource Group” on page 38
Add logical hostnames or shared addresses and data service resources to resource groups	“How to Add a Logical Hostname Resource to a Resource Group” on page 41 “How to Add a Shared Address Resource to a Resource Group” on page 42 “How to Add a Failover Application Resource to a Resource Group” on page 44 “How to Add a Scalable Application Resource to a Resource Group” on page 46
Enable resources and resource monitors, manage the resource group, and bring the resource group and its associated resources online	“How to Bring a Resource Group Online” on page 48

TABLE 2-1 Task Map: Data Service Administration (Continued)

Task	For Instructions, Go To ...
Disable and enable resource monitors independent of the resource	<p>“How to Disable a Resource Fault Monitor” on page 50</p> <p>“How to Enable a Resource Fault Monitor” on page 50</p>
Remove resource types from the cluster	<p>“How to Remove a Resource Type” on page 51</p>
Remove resource groups from the cluster	<p>“How to Remove a Resource Group” on page 52</p>
Remove resources from resource groups	<p>“How to Remove a Resource” on page 54</p>
Switch the primary for a resource group	<p>“How to Switch the Current Primary of a Resource Group” on page 55</p>
Disable resources and move their resource group into the UNMANAGED state	<p>“How to Disable a Resource and Move Its Resource Group Into the UNMANAGED State” on page 57</p>
Display resource type, resource group, and resource configuration information	<p>“How to Display Resource Type, Resource Group, and Resource Configuration Information” on page 59</p>
Change resource type, resource group, and resource properties	<p>“How to Change Resource Type Properties” on page 60</p> <p>“How to Change Resource Group Properties” on page 61</p> <p>“How to Change Resource Properties” on page 62</p>
Clear error flags for failed Resource Group Manager (RGM) processes	<p>“How to Clear the STOP_FAILED Error Flag on Resources” on page 63</p>
Re-register the built-in resource types LogicalHostname and SharedAddress	<p>“How to Re-register Preregistered Resource Types” on page 65</p>
Update the network interface ID list for the network resources, and update the node list for the resource group	<p>“How to Add a Node to a Resource Group” on page 66</p>
Remove a node from a resource group	<p>“How to Remove a Node From a Resource Group” on page 69</p>
Set up HASTorage or HASToragePlus for resource groups so as to synchronize the startups between those resource groups and disk device groups	<p>“How to Set Up HASTorage Resource Type for New Resources” on page 74</p>

TABLE 2-1 Task Map: Data Service Administration (Continued)

Task	For Instructions, Go To ...
Set up <code>HASStoragePlus</code> to enable highly available local file systems for failover data services with high I/O disk intensity	"How to Set Up <code>HASStoragePlus</code> Resource Type" on page 80
Configure a resource type to automatically free up a node for a critical data service.	"How to Set Up an <code>RGOffload</code> Resource" on page 82

Note – The procedures in this chapter describe how to use the `scrgadm(1M)` command to complete these tasks. Other tools also enable you to administer your resources. See "Tools for Data Service Resource Administration" on page 22 for details about these options.

Configuring and Administering Sun Cluster Data Services

Configuring a Sun Cluster data service is a single task composed of several procedures. These procedures enable you to perform the following tasks.

- Register a resource type.
- Upgrade a resource type.
- Create resource groups.
- Add resources into the resource groups.
- Bring the resources online.

Use the procedures in this chapter to update your data service configuration after the initial configuration. For example, to change resource type, resource group, and resource properties, go to "Changing Resource Type, Resource Group, and Resource Properties" on page 59.

Registering a Resource Type

A resource type provides specification of common properties and callback methods that apply to all of the resources of the given type. You must register a resource type before you create a resource of that type. See Chapter 1 for details about resource types.

▼ How to Register a Resource Type

To complete this procedure, you must supply the name for the resource type that you plan to register. The resource type name is an abbreviation for the data service name. This name maps to the name shown on your data service license certificate. See the *Sun Cluster 3.1 Release Notes* for details on the mapping between the names and the license certificate names.

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. Become superuser on a cluster member.

2. Register the resource type.

```
# scrgadm -a -t resource-type
```

-a Adds the specified resource type.

-t *resource-type* Specifies name of the resource type to add. See the *Sun Cluster 3.1 Release Notes* to determine the predefined name to supply.

3. Verify that the resource type has been registered.

```
# scrgadm -pv -t resource-type
```

Example – Registering Resource Types

The following example registers Sun Cluster HA for Sun ONE Web Server (internal name `iws`).

```
# scrgadm -a -t SUNW.iws
# scrgadm -pv -t SUNW.iws
Res Type name:                SUNW.iws
(SUNW.iws) Res Type description:  None registered
```

```
(SUNW.iws) Res Type base directory:      /opt/SUNWschtt/bin
(SUNW.iws) Res Type single instance:    False
(SUNW.iws) Res Type init nodes:        All potential masters
(SUNW.iws) Res Type failover:          False
(SUNW.iws) Res Type version:            1.0
(SUNW.iws) Res Type API version:        2
(SUNW.iws) Res Type installed on nodes: All
(SUNW.iws) Res Type packages:          SUNWschtt
```

Where to Go From Here

After registering resource types, you can create resource groups and add resources to the resource group. See “Creating a Resource Group” on page 37 for details.

Upgrading a Resource Type

As newer versions of resource types are released, you will want to install and register the upgraded resource type. You may also want to upgrade your existing resources to the newer resource type versions. This section provides the following procedures for installing and registering an upgraded resource type and for upgrading an existing resource to a new resource type version.

- “How to Install and Register an Upgrade of a Resource Type” on page 32
- “How to Migrate Existing Resources to a New Version of the Resource Type” on page 33

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

This procedure can also be performed using the Resource Group option of `scsetup`. For information on `scsetup`, see the `scsetup(1M)` man page.

1. **Install the resource type upgrade package on all cluster nodes.**

Note – If the resource type package is not installed on all of the nodes, then an additional step will be required (Step 3).

The upgrade documentation will indicate whether it is necessary to boot a node in non-cluster mode to install the resource type upgrade package. To avoid down time, add the new package in a rolling upgrade fashion on one node at a time, while the node is booted in non-cluster mode and the other nodes are in cluster mode.

2. Register the new resource type version.

```
scrgadm -a -t resource_type -f path_to_new_RTR_file
```

The new resource type will have a name in the following format.

```
vendor_id.raname:version
```

Use `scrgadm -p` or `scrgadm -pv` (verbose) to display the newly registered resource type.

3. If the new resource type is not installed on all of the nodes, set the `Installed_nodes` property to the nodes on which it is actually installed.

```
scrgadm -c -t resource_type -h installed_node_list
```

A new version of a resource type may differ from a previous version in the following ways.

- The settings of the resource type properties may change.
- The set of declared resource properties, including standard and extension properties, may change.
- The attributes of resource properties, such as default, min, max, arraymin, arraymax, or tunability may change.
- The set of declared methods may differ.
- The implementation of methods and/or monitor may change.

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

This procedure can also be performed using the Resource Group option of `scsetup`. For information on `scsetup`, see the `scsetup (1M)` man page.

The existing resource type version and the changes in the new version determine how to migrate to the new version type. The resource type upgrade documentation will tell you whether the migration can occur. If a migration is not supported, consider deleting the resource and replacing it with a new resource of the upgraded version or leaving the resource at the old version of the resource type.

When you migrate the existing resource, the following values may change.

Default property values	If an upgraded version of the resource type declares a new default value for a defaulted
-------------------------	--

property, the new default value will be inherited by existing resources.

The new resource type version's `VALIDATE` method checks to make sure that existing property settings are appropriate. If the settings are not appropriate, edit the properties of the existing resource to appropriate values. To edit the properties, see Step 3.

Resource type name

The RTR file contains the following properties that are used to form the fully qualified name of the resource type.

- `Vendor_id`
- `Resource_type`
- `RT_Version`

When you register the upgraded version of the resource type, its name will be stored as `vendor_id.raname:version`. A resource that has been migrated to a new version will have a new `Type` property, composed of the properties listed above.

Resource `type_version` property

The standard resource property `Type_version` stores the `RT_Version` property of a resource's type. The `Type_Version` property does not appear in the RTR file. Edit the `Type_Version` property using the following command.

```
scrgadm -c -j resource -y Type_version=new_version
```

1. Before migrating an existing resource to a new version of the resource type, read the upgrade documentation accompanying the new resource type to determine whether the migration can take place.

The documentation will specify when the migration must take place.

- Any time
- When the resource is unmonitored
- When the resource is offline
- When the resource is disabled
- When the resource group is unmanaged

If the migration is not supported, you must delete the resource and replace it with a new resource of the upgraded version, or leave the resource at the old version of the resource type.

2. For each resource of the resource type that is to be migrated, change the state of the resource or its resource group to the appropriate state as dictated by the upgrade documentation.

For example, if the resource needs to be unmonitored

```
scswitch -M -n -j resource
```

If the resource needs to be offline

```
scswitch -n -j resource
```

If the resource needs to be disabled

```
scswitch -n -j resource
```

If the resource group needs to be unmanaged

```
scswitch -n -j resource-group
```

```
scswitch -F -g resource_group
```

```
scswitch -u -g resource_group
```

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

```
scrgadm -c -j resource -y Type_version=new_version \  
-x extension_property=new_value -y extension_property=new_value
```

If necessary, edit other properties of the same resource to appropriate values in the same command by adding additional `—x` and/or `—y` options on the command line..

4. Restore the previous state of the resource or resource group by reversing the command typed in Step 2.

For example, to make the resource monitored again

```
scswitch -M -e -j resource
```

To re-enable the resource

```
scswitch -e -j resource
```

To make the resource group managed and online

```
scswitch -o -g resource_group
```

```
scswitch -Z -g resource_group
```

Example 1 – Migrating an Existing Resource to a New Resource Type Version

This example shows the migration of an existing resource to a new resource type version. Note that the new resource type package contains methods located in new paths. Because the methods will not be overwritten during the installation, the resource does not need to be disabled until after the upgraded resource type is installed.

This examples assumes the following.

- New resource type version is 2.0
- Tunability from previous version is “when_offline”
- Resource name is “myresource”
- Resource type name is “myrt”
- New RTR file is in /opt/XYZmyrt/etc/XYZ.myrt
- There are no dependencies on the resource to be migrated
- The resource to be migrated can be taken offline while leaving the containing resource group online

```
(Install the new package on all nodes according to vendor's directions.)
# scrgadm -a -t myrt -f /opt/XYZmyrt/etc/XYZ.myrt
# scswitch -n -j myresource
# scrgadm -c -j myresource -y Type_version=2.0
# scswitch -e -j myresource
```

Example 2 – Migrating an Existing Resource to a New Resource Type Version

This example shows the migration of an existing resource to a new resource type version. Note that the new resource type package contains only the monitor and RTR file. Because the monitor will be overwritten during installation, the resource must be disabled before the upgraded resource type is installed.

This example assumes the following.

- New resource type version is 2.0
- Tunability from previous version is “when_unmonitored”
- Resource name is “myresource”
- Resource type name is “myrt”
- New RTR file is in /opt/XYZmyrt/etc/XYZ.myrt

```
# scswitch -M -n -j myresource
(Install the new package according to vendor's directions.)
# scrgadm -a -t myrt -f /opt/XYZmyrt/etc/XYZ.myrt
# scrgadm -c -j myresource -y Type_version=2.0
# scswitch -M -e -j myresource
```

Creating a Resource Group

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

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

The following procedure describes how to use the `scrgadm(1M)` command to register and configure your data service.

See Chapter 1 and the *Sun Cluster 3.1 Concepts Guide* document for conceptual information on resource groups.

▼ How to Create a Failover Resource Group

A failover resource group contains network addresses, such as the built-in resource types `LogicalHostname` and `SharedAddress`, as well as failover resources, such as the data service application resources for a failover data service. The network resources, along with their dependent data service resources, move between cluster nodes when data services fail over or are switched over.

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. **Become superuser on a cluster member.**
2. **Create the failover resource group.**

```
# scrgadm -a -g resource-group [-h nodelist]
```

<code>-a</code>	Adds the specified resource group.
<code>-g resource-group</code>	Specifies your choice of the name of the failover resource group to add. This name must begin with an ASCII character.
<code>-h nodelist</code>	Specifies an optional, ordered list of nodes that can master this resource group. If you do not specify this list, it defaults to all of the nodes in the cluster.

3. Verify that the resource group has been created.

```
# scrgadm -pv -g resource-group
```

Example – Creating a Failover Resource Group

This example shows the addition of a failover resource group (`resource-group-1`) that two nodes (`phys-schost-1` and `phys-schost-2`) can master.

```
# scrgadm -a -g resource-group-1 -h phys-schost1,phys-schost-2
# scrgadm -pv -g resource-group-1
Res Group name:                resource-group-1
(resource-group-1) Res Group RG description:    <NULL>
(resource-group-1) Res Group management state:  Unmanaged
(resource-group-1) Res Group Failback:         False
(resource-group-1) Res Group Nodelist:         phys-schost-1
                                                phys-schost-2
(resource-group-1) Res Group Maximum primaries: 1
(resource-group-1) Res Group Desired primaries: 1
(resource-group-1) Res Group RG dependencies:   <NULL>
(resource-group-1) Res Group mode:             Failover
(resource-group-1) Res Group network dependencies: True
(resource-group-1) Res Group Global_resources_used: All
(resource-group-1) Res Group Pathprefix:
```

Where to Go From Here

After you create a failover resource group, you can add application resources to this resource group. See “Adding Resources to Resource Groups” on page 40 for the procedure.

▼ How to Create a Scalable Resource Group

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

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. Become superuser on a cluster member.
2. Create the failover resource group that holds the shared addresses that the scalable resource will use.
3. Create the scalable resource group.

```
# scrgadm -a -g resource-group \
-y Maximum primaries=m \
-y Desired primaries=n \
-y RG_dependencies=depend-resource-group \
-h nodelist]
```

-a

Adds a scalable resource group.

-g *resource-group*

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

-y *Maximum primaries=m*

Specifies the maximum number of active primaries for this resource group.

-y *Desired primaries=n*

Specifies the number of active primaries on which the resource group should attempt to start.

-y *RG_dependencies=depend-resource-group*

Identifies the resource group that contains the shared address resource on which the resource group that is being created depends.

-h *nodelist*

Specifies an optional list of nodes on which this resource group is to be available. If you do not specify this list, the value defaults to all of the nodes.

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

```
# scrgadm -pv -g resource-group
```

Example – Creating a Scalable Resource Group

This example shows the addition of a scalable resource group (*resource-group-1*) to be hosted on two nodes (*phys-schost-1*, *phys-schost-2*). The scalable resource group depends on the failover resource group (*resource-group-2*) that contains the shared addresses.

```
# scrgadm -a -g resource-group-1 \
-y Maximum primaries=2 \
-y Desired primaries=2 \
-y RG_dependencies=resource-group-2 \
-h phys-schost-1,phys-schost-2
```

```
# scrgadm -pv -g resource-group-1
```

```
Res Group name:                resource-group-1
(resource-group-1) Res Group RG_description:    <NULL>
```

```
(resource-group-1) Res Group management state:      Unmanaged
(resource-group-1) Res Group Failback:             False
(resource-group-1) Res Group Nodelist:             phys-schost-1
                                                    phys-schost-2
(resource-group-1) Res Group Maximum primaries:    2
(resource-group-1) Res Group Desired primaries:    2
(resource-group-1) Res Group RG_dependencies:      resource-group-2
(resource-group-1) Res Group mode:                 Scalable
(resource-group-1) Res Group network dependencies: True
(resource-group-1) Res Group Global_resources_used: All
(resource-group-1) Res Group Pathprefix:
```

Where to Go From Here

After you have created a scalable resource group, you can add scalable application resources to the resource group. See “How to Add a Scalable Application Resource to a Resource Group” on page 46 for details.

Adding Resources to Resource Groups

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

- logical hostname resources
- shared address resources
- data service (application) resources

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

See the *Sun Cluster 3.1 Concepts Guide* document and Chapter 1 for more information on resources.

▼ How to Add a Logical Hostname Resource to a Resource Group

To complete this procedure, you must supply the following information.

- the name of the failover resource group into which you are adding the resource
- the hostnames that you plan to add to the resource group

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. Become superuser on a cluster member.

2. Add the logical hostname resource to the resource group.

```
# scrgadm -a -L [-j resource] -g resource-group -l hostnamelist, ... [-n netiflist]
```

<code>-a</code>	Adds a logical hostname resource.
<code>-L</code>	Specifies the logical hostname resource form of the command.
<code>-j resource</code>	Specifies an optional resource name of your choice. If you do not specify this option, the name defaults to the first hostname that is specified with the <code>-l</code> option.
<code>-g resource-group</code>	Specifies the name of the resource group in which this resource resides.
<code>-l hostnamelist, ...</code>	Specifies a comma-separated list of UNIX hostnames (logical hostnames) by which clients communicate with services in the resource group.
<code>-n netiflist</code>	Specifies an optional, comma-separated list that identifies the IP Networking Multipathing groups that are on each node. Each element in <code>netiflist</code> must be in the form of <code>netif@node</code> . <code>netif</code> can be given as an IP Networking Multipathing group name, such as <code>sc_ipmp0</code> . The node can be identified by the node name or node ID, such as <code>sc_ipmp0@1</code> or <code>sc_ipmp@phys-schost-1</code> .

Note – Sun Cluster does not currently support using the adapter name for `netif`.

3. Verify that the logical hostname resource has been added.

```
# scrgadm -pv -j resource
```

Adding the resource causes the Sun Cluster software to validate the resource. If the validation succeeds, you can enable the resource, and you can move the resource group into the state where the RGM manages it. If the validation fails, the `scrgadm` command produces an error message and exits. If the validation fails, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `scrgadm` command.

Example – Adding a Logical Hostname Resource to a Resource Group

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

```
# scrgadm -a -L -j resource-1 -g resource-group-1 -l schost-1
# scrgadm -pv -j resource-1
Res Group name: resource-group-1
(resource-group-1) Res name: resource-1
(resource-group-1:resource-1) Res R_description:
(resource-group-1:resource-1) Res resource type: SUNW.LogicalHostname
(resource-group-1:resource-1) Res resource group name: resource-group-1
(resource-group-1:resource-1) Res enabled: False
(resource-group-1:resource-1) Res monitor enabled: True
```

Where to Go From Here

After you add logical hostname resources, use the procedure “How to Bring a Resource Group Online” on page 48 to bring them online.

▼ How to Add a Shared Address Resource to a Resource Group

To complete this procedure, you must supply the following information.

- The name of the resource group into which you are adding the resource. This group must be a failover resource group that you created previously.
- The hostnames that you plan to add to the resource group.

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. Become superuser on a cluster member.

2. Add the shared address resource to the resource group.

```
# scrgadm -a -S [-j resource] -g resource-group -l hostnamelist, ... \  
[-X auxnodelist] [-n netiflist]
```

-a	Adds shared address resources.
-S	Specifies the shared address resource form of the command.
-j <i>resource</i>	Specifies an optional resource name of your choice. If you do not specify this option, the name defaults to the first hostname that is specified with the -l option.
-g <i>resource-group</i>	Specifies the resource group name.
-l <i>hostnamelist, ...</i>	Specifies a comma-separated list of shared address hostnames.
-X <i>auxnodelist</i>	Specifies a comma-separated list of physical node names or IDs that identify the cluster nodes that can host the shared address but never serve as primary if failover occurs. These nodes are mutually exclusive, with the nodes identified as potential masters in the resource group's node list.
-n <i>netiflist</i>	Specifies an optional, comma-separated list that identifies the IP Networking Multipathing groups that are on each node. Each element in <i>netiflist</i> must be in the form of <i>netif@node</i> . <i>netif</i> can be given as an IP Networking Multipathing group name, such as <i>sc_ipmp0</i> . The node can be identified by the node name or node ID, such as <i>sc_ipmp0@1</i> or <i>sc_ipmp@phys-schost-1</i> .

Note – Sun Cluster does not currently support using the adapter name for *netif*.

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

```
# scrgadm -pv -j resource
```

Adding the resource causes the Sun Cluster software to validate the resource. If the validation succeeds, you can enable the resource, and you can move the resource group into the state where the RGM manages it. If the validation fails, the `scrgadm` command produces an error message and exits. If the validation fails, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `scrgadm` command.

Example – Adding a Shared Address Resource to a Resource Group

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

```
# scrgadm -a -S -j resource-1 -g resource-group-1 -l schost-1
# scrgadm -pv -j resource-1
(resource-group-1) Res name:                               resource-1
(resource-group-1:resource-1) Res R_description:
(resource-group-1:resource-1) Res resource type:         SUNW.SharedAddress
(resource-group-1:resource-1) Res resource group name:   resource-group-1
(resource-group-1:resource-1) Res enabled:               False
(resource-group-1:resource-1) Res monitor enabled:       True
```

Where to Go From Here

After you add a shared resource, use the procedure “How to Bring a Resource Group Online” on page 48 to enable the resource.

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

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

To complete this procedure, you must supply the following information.

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

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. **Become superuser on a cluster member.**
2. **Add a failover application resource to the resource group.**

```
# scrgadm -a -j resource -g resource-group -t resource-type \
[-x Extension_property=value, ...] [-y Standard_property=value, ...]
```

-a	Adds a resource.
-j <i>resource</i>	Specifies your choice of the name of the resource to add.
-g <i>resource-group</i>	Specifies the name of the failover resource group created previously.
-t <i>resource-type</i>	Specifies the name of the resource type for the resource.
-x <i>Extension_property=value, ...</i>	Specifies a comma-separated list of extension properties that depend on the particular data service. See the chapter for each data service to determine whether the data service requires this property.
-y <i>Standard_property=value, ...</i>	Specifies a comma-separated list of standard properties that depends on the particular data service. See the chapter for each data service and Appendix A to determine whether the data service requires this property.

Note – You can set additional properties. See Appendix A and the chapter in this book on how to install and configure your failover data service for details.

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

```
# scrgadm -pv -j resource
```

Adding the resource causes the Sun Cluster software to validate the resource. If the validation succeeds, you can enable the resource, and you can move the resource group into the state where the RGM manages it. If the validation fails, the `scrgadm` command produces an error message and exits. If the validation fails, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `scrgadm` command.

Example – Adding a Failover Application Resource to a Resource Group

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

```
# scrgadm -a -j resource-1 -g resource-group-1 -t resource-type-1 \
-y Network_resources_used=schost-1,schost2 \
# scrgadm -pv -j resource-1
(resource-group-1) Res name: resource-1
(resource-group-1:resource-1) Res R_description:
(resource-group-1:resource-1) Res resource type: resource-type-1
(resource-group-1:resource-1) Res resource group name: resource-group-1
(resource-group-1:resource-1) Res enabled: False
(resource-group-1:resource-1) Res monitor enabled: True
```

Where to Go From Here

After you add a failover application resource, use the procedure “How to Bring a Resource Group Online” on page 48 to enable the resource.

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

A scalable application resource is an application resource that uses shared addresses in a failover resource group.

To complete this procedure, you must supply the following information:

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

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. **Become superuser on a cluster member.**
2. **Add a scalable application resource to the resource group.**

```
# scrgadm -a -j resource -g resource-group -t resource-type \
-y Network_resources_used=network-resource[,network-resource...] \
-y Scalable=True
[-x Extension_property=value, ...] [-y Standard_property=value, ...]
-a
Adds a resource.
```

- j *resource*
Specifies your choice of the name of the resource to add.
- g *resource-group*
Specifies the name of a scalable service resource group that you previously created.
- t *resource-type*
Specifies the name of the resource type for this resource.
- y `Network_resources_used= network-resource[,network-resource...]`
Specifies the list of network resources (shared addresses) on which this resource depends.
- y `Scalable=True`
Specifies that this resource is scalable.
- x `Extension_property=value, ...`
Specifies a comma-separated list of extension properties that depend on the particular data service. See the chapter for each data service to determine whether the data service requires this property.
- y `Standard_property=value, ...`
Specifies a comma-separated list of standard properties that depends on the particular data service. See the chapter for each data service and Appendix A to determine whether the data service requires this property.
- y `Standard_property=value, ...`
Specifies a comma-separated list of standard properties that depends on the particular data service. See the chapter for each data service and Appendix A to determine whether the data service requires this property.

Note – You can set additional properties. See Appendix A and the chapter in this book on how to install and configure your scalable data service for information on other configurable properties. Specifically for scalable services, you typically set the `Port_list`, `Load_balancing_weights`, and `Load_balancing_policy` properties, which Appendix A describes.

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

```
# scrgadm -pv -j resource
```

Adding the resource causes the Sun Cluster software to validate the resource. If the validation succeeds, you can enable the resource, and you can move the resource group into the state where the RGM manages it. If the validation fails, the `scrgadm` command produces an error message and exits. If the validation fails, check the `syslog` on each node for an error message. The message appears on the node that performed the validation, not necessarily the node on which you ran the `scrgadm` command.

Example – Adding a Scalable Application Resource to a Resource Group

This example shows the addition of a resource (`resource-1`) to a resource group (`resource-group-1`). Note that `resource-group-1` depends on the failover resource group that contains the network addresses that are in use (`schost-1` and `schost-2` in the following example). The resource depends on shared address resources (`schost-1`, `schost-2`), which must reside in one or more failover resource groups that you defined previously.

```
# scrgadm -a -j resource-1 -g resource-group-1 -t resource-type-1 \  
-y Network_resources_used=schost-1,schost-2 \  
-y Scalable=True  
# scrgadm -pv -j resource-1  
(resource-group-1) Res name: resource-1  
(resource-group-1:resource-1) Res R_description:  
(resource-group-1:resource-1) Res resource type: resource-type-1  
(resource-group-1:resource-1) Res resource group name: resource-group-1  
(resource-group-1:resource-1) Res enabled: False  
(resource-group-1:resource-1) Res monitor enabled: True
```

Where to Go From Here

After you add a scalable application resource, follow the procedure “How to Bring a Resource Group Online” on page 48 to enable the resource.

Bringing Resource Groups Online

To enable resources to begin providing HA services, you must enable the resources in the resource group, enable the resource monitors, make the resource group managed, and bring the resource group online. You can perform these tasks individually or by using the following one-step procedure. See the `scswitch(1M)` man page for details.

Note – Perform this procedure from any cluster node.

▼ How to Bring a Resource Group Online

1. **Become superuser on a cluster member.**
2. **Enable the resource, and bring the resource group online.**
If the resource monitor was disabled, it will be enabled also.


```
# scswitch -Z -g resource-group
-Z           Brings a resource group online by first enabling its resources and
             fault monitors.
-g resource-group Specifies the name of the resource group to bring online. The
             group must be an existing resource group.
```

3. Verify that the resource is online.

Run the following command on any cluster node, and check the resource group state field to verify that the resource group is online on the nodes that are specified in the node list.

```
# scstat -g
```

Example – Bring a Resource Group Online

This example shows how to bring a resource group (*resource-group-1*) online and verify its status.

```
# scswitch -Z -g resource-group-1
# scstat -g
```

Where to Go From Here

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

Disabling and Enabling Resource Monitors

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

See the `scswitch(1M)` man page for additional information.

Note – Run this procedure from any cluster node.

▼ How to Disable a Resource Fault Monitor

1. **Become superuser on a cluster member.**

2. **Disable the resource fault monitor.**

```
# scswitch -n -M -j resource
-n          Disable a resource or resource monitor.
-M          Disable the fault monitor for the specified resource.
-j resource The name of the resource.
```

3. **Verify that the resource fault monitor has been disabled.**

Run the following command on each cluster node, and check for monitored fields (RS Monitored).

```
# scrgadm -pv
```

Example—Disabling a Resource Fault Monitor

This example shows how to disable a resource fault monitor.

```
# scswitch -n -M -j resource-1
# scrgadm -pv
...
RS Monitored: no...
```

▼ How to Enable a Resource Fault Monitor

1. **Become superuser on a cluster member.**

2. **Enable the resource fault monitor.**

```
# scswitch -e -M -j resource
-e          Enables a resource or resource monitor.
-M          Enables the fault monitor for the specified resource.
-j resource Specifies the name of the resource.
```

3. Verify that the resource fault monitor has been enabled.

Run the following command on each cluster node, and check for monitored fields (RS Monitored).

```
# scrgadm -pv
```

Example—Enabling a Resource Fault Monitor

This example shows how to enable a resource fault monitor.

```
# scswitch -e -M -j resource-1
# scrgadm -pv
...
RS Monitored: yes...
```

Removing Resource Types

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

See the `scrgadm(1M)` and `scswitch(1M)` man pages for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Remove a Resource Type

Before you remove a resource type, you must disable and remove all of the resources of that type in all of the resource groups that are in the cluster. Use the `scrgadm -pv` command to identify the resources and resource groups that are in the cluster.

- 1. Become superuser on a cluster member.**
- 2. Disable each resource of the resource type that you will remove.**

```
# scswitch -n -j resource
```

-n Disables the resource.

-j *resource* Specifies the name of the resource to disable.
- 3. Remove each resource of the resource type that you will remove.**

```
# scrgadm -r -j resource
-r                Removes the specified resource.
-j                Specifies the name of the resource to remove.
```

4. Remove the resource type.

```
# scrgadm -r -t resource-type
-r                Removes the specified resource type.
-t resource-type Specifies the name of the resource type to remove.
```

5. Verify that the resource type has been removed.

```
# scrgadm -p
```

Example – Removing a Resource Type

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

```
# scswitch -n -j resource-1
# scrgadm -r -j resource-1
# scrgadm -r -t resource-type-1
```

Removing Resource Groups

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

See the `scrgadm(1M)` and `scswitch(1M)` man pages for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Remove a Resource Group

1. Become superuser on a cluster member.
2. Run the following command to switch the resource group offline.

```
# scswitch -F -g resource-group
-F                Switches a resource group offline.
-g resource-group Specifies the name of the resource group to take offline.
```

3. Disable all of the resources that are part of the resource group.

You can use the `scrgadm -pv` command to view the resources in the resource group. Disable all of the resources in the resource group that you will remove.

```
# scswitch -n -j resource
-n                Disables the resource.
-j resource       Specifies the name of the resource to disable.
```

If any dependent data service resources exist in a resource group, you cannot disable the resource until you have disabled all of the resources that depend on it.

4. Remove all of the resources from the resource group.

Use the `scrgadm` command to perform the following tasks.

- Remove the resources.
- Remove the resource group.

```
# scrgadm -r -j resource
# scrgadm -r -g resource-group
-r                Removes the specified resource or resource group.
-j resource       Specifies the name of the resource to be removed.
-g resource-group Specifies the name of the resource group to be removed.
```

5. Verify that the resource group has been removed.

```
# scrgadm -p
```

Example – Removing a Resource Group

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

```
# scswitch -F -g resource-group-1
# scrgadm -r -j resource-1
# scrgadm -r -g resource-group-1
```

Removing Resources

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

See the `scrgadm(1M)` and `scswitch(1M)` man pages for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Remove a Resource

1. **Become superuser on a cluster member.**
2. **Disable the resource that you want to remove.**

```
# scswitch -n -j resource
-n           Disables the resource.
-j resource  Specifies the name of the resource to disable.
```

3. **Remove the resource.**

```
# scrgadm -r -j resource
-r           Removes the specified resource.
-j resource  Specifies the name of the resource to remove.
```

4. **Verify that the resource has been removed.**

```
# scrgadm -p
```

Example – Removing a Resource

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

```
# scswitch -n -j resource-1
# scrgadm -r -j resource-1
```

Switching the Current Primary of a Resource Group

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

See the `scrgadm(1M)` and `scswitch(1M)` man pages for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Switch the Current Primary of a Resource Group

To complete this procedure, you must supply the following information.

- The name of the resource group that you will switch over.
- The names of the nodes on which you want the resource group to be brought online or to remain online. These nodes must be cluster nodes that have been set up to be potential masters of the resource group that you will switch. To see a list of potential primaries for the resource group, use the `scrgadm -pv` command.

1. Become superuser on a cluster member.

2. Switch the primary to a potential primary.

```
# scswitch -z -g resource-group -h nodelist
```

`-z` Switches the specified resource group online.

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

`-h nodelist` Specifies the node or nodes on which the resource group is to be brought online or is to remain online. This resource group is then switched offline on all of the other nodes.

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

Run the following command, and check the output for the state of the resource group that has been switched over.

```
# scstat -g
```

Example – Switching the Resource Group to a New Primary

This example shows how to switch a resource group (`resource-group-1`) from its current primary (`phys-schost-1`) to the potential primary (`phys-schost-2`). First, verify that the resource group is online on `phys-schost-1`. Next, perform the switch. Finally, verify that the group is switched to be online on `phys-schost-2`.

```
phys-schost-1# scstat -g
...
Resource Group Name:      resource-group-1
  Status
  Node Name:              phys-schost-1
  Status:                 Online

  Node Name:              phys-schost-2
  Status:                 Offline
...
phys-schost-1# scswitch -z -g resource-group-1 -h phys-schost-2
phys-schost-1# scstat -g
...
Resource Group Name:      resource-group-1
  Status
  Node Name:              phys-schost-2
  Status:                 Online

  Node Name:              phys-schost-1
  Status:                 Offline
...
```

Disabling Resources and Moving Their Resource Group Into the UNMANAGED State

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

See the `scrgadm(1M)` and `scswitch(1M)` man pages for additional information.

Note – Perform this procedure from any cluster node.

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

To complete this procedure, you must supply the following information.

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

To determine the resource and resource group names that you need for this procedure, use the `scrgadm -pv` command.

1. Become superuser on a cluster member.

2. Disable the resource.

Repeat this step for all of the resources in the resource group.

```
# scswitch -n -j resource
```

`-n` Disables the resource.

`-j resource` Specifies the name of the resource to disable.

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

```
# scswitch -F -g resource-group
```

`-F` Switches a resource group offline.

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

4. Move the resource group into the UNMANAGED state.

```
# scswitch -u -g resource-group
```

`-u` Moves the specified resource group in the UNMANAGED state.

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

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

```
# scrgadm -pv -g resource-group
```

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

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

```

# scswitch -n -j resource-1
# scswitch -F -g resource-group-1
# scswitch -u -g resource-group-1
# scrgadm -pv -g resource-group-1
Res Group name:                resource-group-1
(resource-group-1) Res Group RG_description:    <NULL>
(resource-group-1) Res Group management state:  Unmanaged
(resource-group-1) Res Group Failback:         False
(resource-group-1) Res Group Nodelist:         phys-schost-1
                                                phys-schost-2

(resource-group-1) Res Group Maximum primaries: 2
(resource-group-1) Res Group Desired primaries: 2
(resource-group-1) Res Group RG_dependencies:  <NULL>
(resource-group-1) Res Group mode:             Failover
(resource-group-1) Res Group network dependencies: True
(resource-group-1) Res Group Global_resources_used: All
(resource-group-1) Res Group Pathprefix:

(resource-group-1) Res name:                resource-1
(resource-group-1:resource-1) Res R_description:
(resource-group-1:resource-1) Res resource type:  SUNW.apache
(resource-group-1:resource-1) Res resource group name: resource-group-1
(resource-group-1:resource-1) Res enabled:        True
(resource-group-1:resource-1) Res monitor enabled: False
(resource-group-1:resource-1) Res detached:      False

```

Displaying Resource Type, Resource Group, and Resource Configuration Information

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

See the `scrgadm(1M)` and `scswitch(1M)` man pages for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Display Resource Type, Resource Group, and Resource Configuration Information

The `scrgadm` command provides the following three levels of configuration status information.

- With the `-p` option, the output shows a very limited set of property values for resource types, resource groups, and resources.
- With the `-pv` option, the output shows more details on other resource type, resource group, and resource properties.
- With the `-pvv` option, the output provides a detailed view, including resource type methods, extension properties, and all of the resource and resource group properties.

You can also use the `-t`, `-g`, and `-j` (resource type, resource group, and resource, respectively) options, followed by the name of the object that you want to view, to check status information on specific resource types, resource groups, and resources. For example, the following command specifies that you want to view specific information on the resource `apache-1` only.

```
# scrgadm -p[v[v]] -j apache-1
```

See the `scrgadm(1M)` man page for details.

Changing Resource Type, Resource Group, and Resource Properties

Resource groups and resources have standard configuration properties that you can change. The following procedures describe how to change these properties.

Resources also have extension properties—some of which the data service developer predefines—that you cannot change. See the individual data service chapters in this document for a list of the extension properties for each data service.

See the `scrgadm(1M)` man page for information on the standard configuration properties for resource groups and resources.

▼ How to Change Resource Type Properties

To complete this procedure, you must supply the following information.

- The name of the resource type to change.
- The name of the resource type property to change. For resource types, you can change only one property—the list of nodes on which resources of this type can be instantiated.

Note – Perform this procedure from any cluster node.

1. **Become superuser on a cluster member.**
2. **Run the `scrgadm` command to determine the name of the resource type that you need for this procedure.**

```
# scrgadm -pv
```

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

The only property that can be changed for a resource type is the `Installed_node_list` property.

```
# scrgadm -c -t resource-type -h installed-node-list
```

<code>-c</code>	Changes the specified resource type property.
<code>-t resource-type</code>	Specifies the name of the resource type.
<code>-h installed-node-list</code>	Specifies the names of nodes on which this resource type is installed.

4. **Verify that the resource type property has been changed.**

```
# scrgadm -pv -t resource-type
```

Example – Changing a Resource Type Property

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

```
# scrgadm -c -t SUNW.apache -h phys-schost-1,phys-schost-2
# scrgadm -pv -t SUNW.apache
Res Type name:                SUNW.apache
(SUNW.apache) Res Type description:  Apache Resource Type
(SUNW.apache) Res Type base directory: /opt/SUNWscapc/bin
(SUNW.apache) Res Type single instance:  False
(SUNW.apache) Res Type init nodes:      All potential masters
(SUNW.apache) Res Type failover:       False
```

```
(SUNW.apache) Res Type version:          1.0
(SUNW.apache) Res Type API version:      2
(SUNW.apache) Res Type installed on nodes: phys-schost1 phys-schost-2
(SUNW.apache) Res Type packages:        SUNWscapc
```

▼ How to Change Resource Group Properties

To complete this procedure, you must supply the following information.

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

This procedure describes the steps to change resource group properties. See Appendix A for a complete list of resource group properties.

Note – Perform this procedure from any cluster node.

1. Become superuser on a cluster member.

2. Change the resource group property.

```
# scrgadm -c -g resource-group -y property=new_value
-c          Changes the specified property.
-g resource-group  Specifies the name of the resource group.
-y property  Specifies the name of the property to change.
```

3. Verify that the resource group property has been changed.

```
# scrgadm -pv -g resource-group
```

Example – Changing a Resource Group Property

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

```
# scrgadm -c -g resource-group-1 -y Failback=True
# scrgadm -pv -g resource-group-1
```

▼ How to Change Resource Properties

To complete this procedure, you must supply the following information.

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

This procedure describes the steps to change resource properties. See Appendix A for a complete list of resource group properties.

Note – Perform this procedure from any cluster node.

1. **Become superuser on a cluster member.**
2. **Run the `scrgadm -pvv` command to view the current resource property settings.**

```
# scrgadm -pvv -j resource
```

3. **Change the resource property.**

```
# scrgadm -c -j resource -y property=new_value | -x extension_property=new_value
```

<code>-c</code>	Changes the specified property.
<code>-j resource</code>	Specifies the name of the resource.
<code>-y property=new_value</code>	Specifies the name of the standard property to change.
<code>-x extension_property=new_value</code>	Specifies the name of the extension property to change. For data services that Sun supplies, see the extension properties that are documented in the chapters on how to install and configure the individual data services.

4. **Verify that the resource property has been changed.**

```
# scrgadm pvv -j resource
```

Example – Changing a Standard Resource Property

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

```
# scrgadm -c -j resource-1 -y start_timeout=30
# scrgadm -pvv -j resource-1
```

Example – Changing an Extension Resource Property

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

```
# scrgadm -c -j resource-1 -x Log_level=3
# scrgadm -pvv -j resource-1
```

Clearing the STOP_FAILED Error Flag on Resources

When the `Failover_mode` resource property is set to `NONE` or `SOFT` and the `STOP` of a resource fails, the individual resource goes into the `STOP_FAILED` state, and the resource group goes into the `ERROR_STOP_FAILED` state. You cannot bring a resource group in this state on any node online, nor can you edit the resource group (create or delete resources, or change resource group or resource properties).

▼ How to Clear the STOP_FAILED Error Flag on Resources

To complete this procedure, you must supply the following information.

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

See the `scswitch(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

1. **Become superuser on a cluster member.**
2. **Identify which resources have gone into the `STOP_FAILED` state and on which nodes.**

```
# scstat -g
```
3. **Manually stop the resources and their monitors on the nodes on which they are in `STOP_FAILED` state.**

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

4. Manually set the state of these resources to OFFLINE on all of the nodes on which you manually stopped the resources.

```
# scswitch -c -h nodelist -j resource -f STOP_FAILED
```

- c Clears the flag.
- h *nodelist* Specifies the node names on which the resource was running.
- j *resource* Specifies the name of the resource to switch offline.
- f **STOP_FAILED** Specifies the flag name.

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

The resource group state should now be OFFLINE or ONLINE.

```
# scstat -g
```

The command `scstat -g` indicates whether the resource group remains in the `ERROR_STOP_FAILED` state. If the resource group is still in the `ERROR_STOP_FAILED` state, then run the following `scswitch` command to switch the resource group offline on the appropriate nodes.

```
# scswitch -F -g resource-group
```

- F Switches the resource group offline on all of the nodes that can master the group.
- g *resource-group* Specifies the name of the resource group to switch offline.

This situation can occur if the resource group was being switched offline when the STOP method failure occurred and the resource that failed to stop had a dependency on other resources in the resource group. Otherwise, the resource group reverts to the ONLINE or OFFLINE state automatically after you have run the command in Step 4 on all of the `STOP_FAILED` resources.

Now you can switch the resource group to the ONLINE state.

Re-registering Preregistered Resource Types

Two preregistered resource types are `SUNW.LogicalHostname` and `SUNW.SharedAddress`. All of the logical hostname and shared address resources use these resource types. You never need to register these two resource types, but you might accidentally delete them. If you have deleted resource types inadvertently, use the following procedure to re-register them.

See the `scrgadm(1M)` man page for additional information.

Note – Perform this procedure from any cluster node.

▼ How to Re-register Preregistered Resource Types

- **Re-register the resource type.**

```
# scrgadm -a -t SUNW.resource-type
```

-a Adds a resource type.

-t `SUNW.resource-type` Specifies the resource type to add (re-register). The resource type can be either `SUNW.LogicalHostname` or `SUNW.SharedAddress`.

Example – Re-registering a Preregistered Resource Type

This example shows how to re-register the `SUNW.LogicalHostname` resource type.

```
# scrgadm -a -t SUNW.LogicalHostname
```

Adding or Removing a Node to or From a Resource Group

The procedures in this section enable you to perform the following tasks.

- Configure a cluster node to be an additional master of a resource group.
- Remove a node from a resource group.

The procedures are slightly different, depending on whether you plan to add or remove the node to or from a failover or scalable resource group.

Failover resource groups contain network resources that both failover and scalable services use. Each IP subnetwork connected to the cluster has its own network resource that is specified and included in a failover resource group. The network resource is either a logical hostname or a shared address resource. Each network resource includes a list of IP Networking Multipathing groups that it uses. For failover resource groups, you must update the complete list of IP Networking Multipathing groups for each network resource that the resource group includes (the `netiflist` resource property).

For scalable resource groups, in addition to changing the scalable group to be mastered on the new set of hosts, you must repeat the procedure for failover groups that contain the network resources that the scalable resource uses.

See the `scrgadm(1M)` man page for additional information.

Note – Run either of these procedures from any cluster node.

▼ How to Add a Node to a Resource Group

This procedure contains the following sections.

- “How to Add a Node to a Scalable Resource Group” on page 67
- “How to Add a Node to a Failover Resource Group” on page 67

You must supply the following information to complete this procedure.

- the names and node IDs of all of the cluster nodes
- the names of the resource groups to which you are adding the node
- the name of the IP Networking Multipathing group that will host the network resources that are used by the resource group on all of the nodes

Also, be sure to verify that the new node is already a cluster member.

▼ How to Add a Node to a Scalable Resource Group

1. For each network resource that a scalable resource in the resource group uses, make the resource group where the network resource is located run on the new node. See Step 1 through Step 4 in the following procedure for details.

2. Add the new node to the list of nodes that can master the scalable resource group (the `nodelist` resource group property).

This step overwrites the previous value of `nodelist`, and therefore you must include all of the nodes that can master the resource group here.

```
# scrgadm -c -g resource-group -h nodelist
```

-c Changes a resource group.

-g *resource-group* Specifies the name of the resource group to which the node is being added.

-h *nodelist* Specifies a comma-separated list of nodes that can master the resource group.

3. (Optional) Update the `Load_balancing_weights` property of the scalable resource to assign a weight to the node that you want to add to the resource group. Otherwise, the weight defaults to 1. See the `scrgadm(1M)` man page for more information.

▼ How to Add a Node to a Failover Resource Group

1. Display the current node list and the current list of IP Networking Multipathing groups that are configured for each resource in the resource group.

```
# scrgadm -pvv -g resource-group | grep -i nodelist  
# scrgadm -pvv -g resource-group | grep -i netiflist
```

Note – The output of the command line for `nodelist` and `netiflist` identifies the nodes by node name. To identify node IDs, run the command `scconf -pv | grep -i node_id`.

2. Update `netiflist` for the network resources that the node addition affects. This step overwrites the previous value of `netiflist`, and therefore you must include all of the IP Networking Multipathing groups here..

```
# scrgadm -c -j network-resource -x netiflist=netiflist
```

-c Changes a network resource.

-j *network-resource* Specifies the name of the network resource (logical hostname or shared address) that is being hosted on the *netiflist* entries.

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

3. Update the node list to include all of the nodes that can now master this resource group.

This step overwrites the previous value of *nodelist*, and therefore you must include all of the nodes that can master the resource group here.

```
# scrgadm -c -g resource-group -h nodelist
```

-c Changes a resource group.

-g *resource-group* Specifies the name of the resource group to which the node is being added.

-h *nodelist* Specifies a comma-separated list of nodes that can master the resource group.

4. Verify the updated information.

```
# scrgadm -pvv -g resource-group | grep -i nodelist
# scrgadm -pvv -g resource-group | grep -i netiflist
```

Example – Adding a Node to a Resource Group

This example shows how to add a node (*phys-schost-2*) to a resource group (*resource-group-1*) that contains a logical hostname resource (*schost-2*).

```
# scrgadm -pvv -g resource-group-1 | grep -i nodelist
(resource-group-1) Res Group Nodelist:   phys-schost-1 phys-schost-3
# scrgadm -pvv -g resource-group-1 | grep -i netiflist
(resource-group-1:schost-2) Res property name: NetIfList
(resource-group-1:schost-2:NetIfList) Res property class: extension
(resource-group-1:schost-2:NetIfList) List of IP Networking Multipathing
interfaces on each node
(resource-group-1:schost-2:NetIfList) Res property type: stringarray
(resource-group-1:schost-2:NetIfList) Res property value: sc_ipmp0@1 sc_ipmp0@3
```

(Only nodes 1 and 3 have been assigned IP Networking Multipathing groups.)

You must add a IP Networking Multipathing group for node 2.)

```
# scrgadm -c -j schost-2 -x netiflist=sc_ipmp0@1,sc_ipmp0@2,sc_ipmp0@3
# scrgadm -c -g resource-group-1 -h phys-schost-1,phys-schost-2,phys-schost-3
# scrgadm -pvv -g resource-group-1 | grep -i nodelist
(resource-group-1) Res Group Nodelist:      phys-schost-1 phys-schost-2
                                           phys-schost-3
# scrgadm -pvv -g resource-group-1 | grep -i netiflist
(resource-group-1:schost-2:NetIfList) Res property value: sc_ipmp0@1 sc_ipmp0@2
                                                         sc_ipmp0@3
```

▼ How to Remove a Node From a Resource Group

This procedure contains the following sections.

- “How to Remove a Node From a Scalable Resource Group” on page 70
- “How to Remove a Node From a Failover Resource Group” on page 71
- “How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources” on page 72
- “Example – Removing a Node From a Resource Group” on page 73

To complete these procedures, you must supply the following information.

- node names and node IDs of all of the cluster nodes

```
# scconf -pv | grep "Node ID"
```
- name(s) of the resource group or groups from which you plan to remove the node

```
# scrgadm -pv | grep "Res Group Nodelist"
```
- names of the IP Networking Multipathing groups that will host the network resources that are used by the resource group(s) on all of the nodes

```
# scrgadm -pvv | grep "NetIfList.*value"
```

Additionally, be sure to verify that the resource group **is not mastered** on the node that you will remove. If the resource group **is mastered** on the node that you will remove, run the `scswitch` command to switch the resource group offline from that node. The following `scswitch` command will bring the resource group offline from a given node, provided that `new-masters` does not contain that node.

```
# scswitch -z -g resource-group -h new-masters
```

<code>-g resource-group</code>	Specifies the name of the resource group (mastered on the node that you will remove) that you are switching offline.
<code>-h new-masters</code>	Specifies the node(s) that will now master the resource group.

See the `scswitch(1M)` man page for additional information.



Caution – If you plan to remove a node from all of the resource groups, and you use a scalable services configuration, first remove the node from the scalable resource group(s). Then, remove the node from the failover group(s).

▼ How to Remove a Node From a Scalable Resource Group

A scalable service is configured as two resource groups, as follows.

- One resource group is a scalable group that contains the scalable service resource.
- One resource group is a failover group that contains the shared address resources that the scalable service resource uses.

Additionally, the `RG_dependencies` property of the scalable resource group is set to configure the scalable group with a dependency on the failover resource group. See Appendix A for details on this property.

See the *Sun Cluster 3.1 Concepts Guide* document for details about scalable service configuration.

Removing a node from the scalable resource group causes the scalable service to no longer be brought online on that node. To remove a node from the scalable resource group, perform the following steps.

1. **Remove the node from the list of nodes that can master the scalable resource group (the `nodelist` resource group property).**

```
# scrgadm -c -g scalable-resource-group -h nodelist
```

- | | |
|---|--|
| <code>-c</code> | Changes a resource group. |
| <code>-g scalable-resource-group</code> | Specifies the name of the resource group from which the node is being removed. |
| <code>-h nodelist</code> | Specifies a comma-separated list of nodes that can master this resource group. |

2. **(Optional) Remove the node from the failover resource group that contains the shared address resource.**

See “How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources” on page 72 for details.

3. **(Optional) Update the `Load_balancing_weights` property of the scalable resource to remove the weight of the node that you want to remove from the resource group.**

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

▼ How to Remove a Node From a Failover Resource Group

Perform the following steps to remove a node from a failover resource group.



Caution – If you plan to remove a node from all of the resource groups, and you use a scalable services configuration, first remove the node from the scalable resource group(s). Then, use this procedure to remove the node from the failover group(s).

Note – If the failover resource group contains shared address resources that scalable services use, see “How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources” on page 72.

1. Update the node list to include all of the nodes that can now master this resource group.

This step removes the node and overwrites the previous value of the node list. Be sure to include all of the nodes that can master the resource group here.

```
# scrgadm -c -g failover-resource-group -h nodelist
```

-c Changes a resource group.

-g *failover-resource-group* Specifies the name of the resource group from which the node is being removed.

-h *nodelist* Specifies a comma-separated list of nodes that can master this resource group.

2. Display the current list of IP Networking Multipathing groups that are configured for each resource in the resource group.

```
# scrgadm -pvv -g failover-resource-group | grep -i netiflist
```

3. Update `netiflist` for network resources that the removal of the node affects.

This step overwrites the previous value of `netiflist`. Be sure to include all of the IP Networking Multipathing groups here.

```
# scrgadm -c -j network-resource -x netiflist=netiflist
```

Note – The output of the preceding command line identifies the nodes by node name. Run the command line `scconf -pv | grep "Node ID"` to find the node ID.

-c Changes a network resource.

<code>-j network-resource</code>	Specifies the name of the network resource that is hosted on the <code>netiflist</code> entries.
<code>-x netiflist=netiflist</code>	Specifies a comma-separated list that identifies the IP Networking Multipathing groups that are on each node. Each element in <code>netiflist</code> must be in the form of <code>netif@node</code> . <code>netif</code> can be given as an IP Networking Multipathing group name, such as <code>sc_ipmp0</code> . The node can be identified by the node name or node ID, such as <code>sc_ipmp0@1</code> or <code>sc_ipmp@phys-schost-1</code> .

Note – Sun Cluster does not currently support using the adapter name for `netif`.

4. Verify the updated information.

```
# scrgadm -pvv -g failover-resource-group | grep -i nodelist
# scrgadm -pvv -g failover-resource-group | grep -i netiflist
```

▼ How to Remove a Node From a Failover Resource Group That Contains Shared Address Resources

In a failover resource group that contains shared address resources that scalable services use, a node can appear in the following locations.

- the node list of the failover resource group
- the `auxnodelist` of the shared address resource

To remove the node from the node list of the failover resource group, follow the procedure “How to Remove a Node From a Failover Resource Group” on page 71.

To modify the `auxnodelist` of the shared address resource, you must remove and recreate the shared address resource.

If you remove the node from the failover group’s node list, you can continue to use the shared address resource on that node to provide scalable services. To do so, you must add the node to the `auxnodelist` of the shared address resource. To add the node to the `auxnodelist`, perform the following steps.

Note – You can also use the following procedure to **remove** the node from the `auxnodelist` of the shared address resource. To remove the node from the `auxnodelist`, you must delete and recreate the shared address resource.

1. Switch the scalable service resource offline.

2. Remove the shared address resource from the failover resource group.

3. Create the shared address resource.

Add the node ID or node name of the node that you removed from the failover resource group to the `auxnodelist`.

```
# scrgadm -a -S -g failover-resource-group\  
-l shared-address -x new-auxnodelist
```

failover-resource-group The name of the failover resource group that used to contain the shared address resource.

shared-address The name of the shared address.

new-auxnodelist The new, modified `auxnodelist` with the desired node added or removed.

Example – Removing a Node From a Resource Group

This example shows how to remove a node (`phys-schost-3`) from a resource group (`resource-group-1`), which contains a logical hostname resource (`schost-1`).

```
# scrgadm -pvv -g resource-group-1 | grep -i nodelist  
(resource-group-1) Res Group Nodelist:      phys-schost-1 phys-schost-2  
                                           phys-schost-3  
  
# scrgadm -c -g resource-group-1 -h phys-schost-1,phys-schost-2  
# scrgadm -pvv -g resource-group-1 | grep -i netiflist  
(resource-group-1:schost-1) Res property name: NetIfList  
(resource-group-1:schost-1:NetIfList) Res property class: extension  
(resource-group-1:schost-1:NetIfList) List of IP Networking Multipathing  
interfaces on each node  
(resource-group-1:schost-1:NetIfList) Res property type: stringarray  
(resource-group-1:schost-1:NetIfList) Res property value: sc_ipmp0@1 sc_ipmp0@2  
                                                         sc_ipmp0@3  
  
(sc_ipmp0@3 is the IP Networking Multipathing group to be removed.)  
  
# scrgadm -c -j schost-1 -x netiflist=sc_ipmp0@1,sc_ipmp0@2  
# scrgadm -pvv -g resource-group-1 | grep -i nodelist  
(resource-group-1) Res Group Nodelist:      phys-schost-1 phys-schost-2  
# scrgadm -pvv -g resource-group-1 | grep -i netiflist  
(resource-group-1:schost-1:NetIfList) Res property value: sc_ipmp0@1 sc_ipmp0@2
```

Synchronizing the Startups Between Resource Groups and Disk Device Groups

After a cluster boots up or services fail over to another node, global devices and cluster file systems might require time to become available. However, a data service can run its *START* method before global devices and cluster file systems—on which the data service depends—come online. In this instance, the *START* method times out, and you must reset the state of the resource groups that the data service uses and restart the data service manually. The resource types *HASStorage* and *HASStoragePlus* monitor the global devices and cluster file systems and cause the *START* method of the other resources in the same resource group to wait until they become available. (To determine which resource type to create, see “Choosing Between *HASStorage* and *HASStoragePlus*” on page 19.) To avoid additional administrative tasks, set up *HASStorage* or *HASStoragePlus* for all of the resource groups whose data service resources depend on global devices or cluster file systems.

To create a *HASStorage* resource type, see “How to Set Up *HASStorage* Resource Type for New Resources” on page 74.

To create a *HASStoragePlus* resource type, see “How to Set Up *HASStoragePlus* Resource Type” on page 80.

▼ How to Set Up *HASStorage* Resource Type for New Resources

HASStorage might not be supported in a future release of Sun Cluster. Equivalent functionality is supported by *HASStoragePlus*. To upgrade from *HASStorage* to *HASStoragePlus*, see xxxx.

In the following example, the resource group `resource-group-1` contains three data services.

- Sun ONE Web Server, which depends on `/global/resource-group-1`
- Oracle, which depends on `/dev/global/dsk/d5s2`
- NFS, which depends on `dsk/d6`

To create a *HASStorage* resource `hastorage-1` for new resources in `resource-group-1`, read “Synchronizing the Startups Between Resource Groups and Disk Device Groups” on page 74 and then perform the following steps.

To create a HAStoragePlus resource type, see “Enabling Highly Available Local File Systems” on page 79.

1. Become superuser on a cluster member.

2. Create the resource group `resource-group-1`.

```
# scrgadm -a -g resource-group-1
```

3. Determine whether the resource type is registered.

The following command prints a list of registered resource types.

```
# scrgadm -p | egrep Type
```

4. If you need to, register the resource type.

```
# scrgadm -a -t SUNW.HAStorage
```

5. Create the HAStorage resource `hastorage-1`, and define the service paths.

```
# scrgadm -a -j hastorage-1 -g resource-group-1 -t SUNW.HAStorage \  
-x ServicePaths=/global/resource-group-1,/dev/global/dsk/d5s2,dsk/d6
```

ServicePaths can contain the following values.

- global device group names, such as `nfs-dg`
- paths to global devices, such as `/dev/global/dsk/d5s2` or `dsk/d6`
- cluster file system mount points, such as `/global/nfs`

Note – Global device groups might not be collocated with the resource groups that correspond to them if `ServicePaths` contains cluster file system paths.

6. Enable the `hastorage-1` resource.

```
# scswitch -e -j hastorage-1
```

7. Add the resources (Sun ONE Web Server, Oracle, and NFS) to `resource-group-1`, and set their dependency to `hastorage-1`.

For example, for Sun ONE Web Server, run the following command.

```
# scrgadm -a -j resource -g resource-group-1 -t SUNW.iws \  
-x Confdir_list=/global/iws/schost-1 -y Scalable=False \  
-y Network_resources_used=schost-1 -y Port_list=80/tcp \  
-y Resource_dependencies=hastorage-1
```

8. Verify that you have correctly configured the resource dependencies.

```
# scrgadm -pvv -j resource | egrep strong
```

9. Set `resource-group-1` to the MANAGED state, and bring `resource-group-1` online.

```
# scswitch -Z -g resource-group-1
```

The `HASStorage` resource type contains another extension property, `AffinityOn`, which is a Boolean that specifies whether `HASStorage` must perform an affinity switchover for the global devices and cluster file systems that are defined in `ServicePaths`. See the `SUNW.HASStorage(5)` man page for details.

Note – `HASStorage` and `HASStoragePlus` do not permit `AffinityOn` to be set to `TRUE` if the resource group is scalable. `HASStorage` and `HASStoragePlus` checks the `AffinityOn` value and internally resets the value to `FALSE` for a scalable resource group.

▼ How to Set Up `HASStorage` Resource Type for Existing Resources

`HASStorage` might not be supported in a future release of Sun Cluster. Equivalent functionality is supported by `HASStoragePlus`. To upgrade from `HASStorage` to `HASStoragePlus`, see xxxx.

To create a `HASStorage` resource for existing resources, read “Synchronizing the Startups Between Resource Groups and Disk Device Groups” on page 74, and then perform the following steps.

1. Determine whether the resource type is registered.

The following command prints a list of registered resource types.

```
# scrgadm -p | egrep Type
```

2. If you need to, register the resource type.

```
# scrgadm -a -t SUNW.HASStorage
```

3. Create the `HASStorage` resource `hastorage-1`.

```
# scrgadm -a -g resource-group -j hastorage-1 -t SUNW.HASStorage \  
-x ServicePaths= ... -x AffinityOn=True
```

4. Enable the `hastorage-1` resource.

```
# scswitch -e -j hastorage-1
```

5. Set up the dependency for each of the existing resources, as required.

```
# scrgadm -c -j resource -y Resource_Dependencies=hastorage-1
```

6. Verify that you have correctly configured the resource dependencies.

```
# scrgadm -pvv -j resource | egrep strong
```

Upgrading from HAStorage to HAStoragePlus

HAStorage might not be supported in a future release of Sun Cluster. Equivalent functionality is supported by HAStoragePlus. To upgrade from HAStorage to HAStorage, see the following sections.

How to Upgrade from HAStorage to HAStoragePlus When Using Device Groups or CFS

HAStorage might not be supported in a future release of Sun Cluster. Equivalent functionality is supported by HAStoragePlus. To upgrade from HAStorage to HAStoragePlus when using device groups or CFS, complete the following steps.

The following example uses a simple HA-NFS resource active with HAStorage. The ServicePaths are the diskgroup `nf_sdg` and the `AffinityOn` property is `TRUE`. Furthermore, the HA-NFS resource has `Resource_Dependencies` set to the HAStorage resource.

1. Remove the dependencies the application resources has on HAStorage.

```
# scrgadm -c -j nfserver-rs -y Resource_Dependencies=""
```

2. Disable the HAStorage resource.

```
# scswitch -n -j nfs1storage-rs
```

3. Remove the HAStorage resource from the application resource group.

```
# scrgadm -r -j nfs1storage-rs
```

4. Unregister the HAStorage resource type.

```
# scrgadm -r -t SUNW.HAStorage
```

5. Register the HAStoragePlus resource type.

```
# scrgadm -a -t SUNW.HAStoragePlus
```

6. Create the HAStoragePlus resource.

To specify a filesystem mount point, input the following text.

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HAStoragePlus -x FilesystemMountPoints=/global/nfsdata -x \  
AffinityOn=True
```

To specify global device paths, input the following text.

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HAStoragePlus -x GlobalDevicePaths=nfsdg -x AffinityOn=True
```

Note – Instead of using the ServicePaths property for HAStorage, you must use the GlobalDevicePaths or FilesystemMountPoints property for HAStoragePlus. The FilesystemMountPoints extension property must match the sequence specified in /etc/vfstab.

7. Enable the HAStoragePlus resource.

```
# scswitch -e -j nfs1-hastp-rs
```

8. Set up the dependencies between the application server and HAStoragePlus.

```
# scrgadm -c -j nfsserver-rs -y \  
Resource_Dependencies=nfs1-hastp-rs
```

How to Upgrade from HAStorage With CFS to HAStoragePlus With Failover Filesystem

HAStorage might not be supported in a future release of Sun Cluster. Equivalent functionality is supported by HAStoragePlus. To upgrade from HAStorage with CFS to HAStoragePlus with Failover Filesystem (FFS), complete the following steps.

The following example uses a simple HA-NFS resource active with HAStorage. The ServicePaths are the diskgroup nfsdg and the AffinityOn property is TRUE. Furthermore, the HA-NFS resource has Resource_Dependencies set to HAStorage resource.

1. Remove the dependencies the application resource has on HAStorage resource.

```
# scrgadm -c -j nfsserver-rs -y Resource_Dependencies=""
```

2. Disable the HAStorage resource.

```
# scswitch -n -j nfs1storage-rs
```

3. Remove the HAStorage resource from the application resource group.

```
# scrgadm -r -j nfs1storage-rs
```

4. Unregister the HAStorage resource type.

```
# scrgadm -r -t SUNW.HAStorage
```

5. Modify `/etc/vfstab` to remove the global flag and change “mount at boot” to “no”.

6. Create the `HASStoragePlus` resource.

To specify a filesystem mount point, input the following text.

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HASStoragePlus -x FilesystemMountPoints=/global/nfsdata -x \  
AffinityOn=True
```

To specify global device paths, input the following text.

```
# scrgadm -a -j nfs1-hastp-rs -g nfs1-rg -t \  
SUNW.HASStoragePlus -x GlobalDevicePaths=nfsdg -x AffinityOn=True
```

Note – Instead of using the `ServicePaths` property for `HASStorage`, you must use the `GlobalDevicePaths` or `FilesystemMountPoints` property for `HASStoragePlus`. The `FilesystemMountPoints` extension property must match the sequence specified in `/etc/vfstab`.

7. Enable the `HASStoragePlus` resource.

```
# scswitch -e -j nfs1-hastp-rs
```

8. Set up the dependencies between the application server and `HASStoragePlus`.

```
# scrgadm -c -j nfserver-rs -y \  
Resource_Dependencies=nfs1=hastp-rs
```

Enabling Highly Available Local File Systems

The `HASStoragePlus` resource type can be used to make a local file system highly available within a Sun Cluster environment. The local file system partitions must reside on global disk groups with affinity switchovers enabled and the Sun Cluster environment must be configured for failover. This enables the user to make any file system on multi-host disks accessible from any host directly connected to those multi-host disks. (You cannot use `HASStoragePlus` to make a root file system highly available.) The failback settings must be identical for both the resource group and device group(s).

Using a highly available local file system is strongly recommended for some I/O intensive data services, and a procedure on how to configure the `HASStoragePlus` resource type has been added to the Registration and Configuration procedures for these data services. For procedures on how to set up the `HASStoragePlus` resource type for these data services, see the following sections.

- “Registering and Configuring Sun Cluster HA for Oracle” in *Sun Cluster 3.1 Data Service for Oracle*
- “Registering and Configuring Sun Cluster HA for Sybase ASE” in *Sun Cluster 3.1 Data Service for Sybase ASE*

For the procedure to set up `HASStoragePlus` resource type for other data services, see “How to Set Up `HASStoragePlus` Resource Type” on page 80.

▼ How to Set Up `HASStoragePlus` Resource Type

The `HASStoragePlus` resource type was introduced in Sun Cluster 3.0 5/02. This new resource type performs the same functions as `HASStorage`, and synchronizes the startups between resource groups and disk device groups. The `HASStoragePlus` resource type has an additional feature to make a local file system highly available. (For background information on making a local file system highly available, see “Enabling Highly Available Local File Systems” on page 79.) To use both of these features, set up the `HASStoragePlus` resource type.

To set up `HASStoragePlus`, the local file system partitions must reside on global disk groups with affinity switchovers enabled and the Sun Cluster environment must be configured for failover.

The following example uses a simple NFS service that shares out home directory data from a locally mounted directory `/global/local-fs/nfs/export/home`. The example assumes the following:

- The mount point `/global/local-fs/nfs` will be used to mount a UFS local file system on a Sun Cluster global device partition.
- The `/etc/vfstab` entry for the `/global/local-fs/nfs` file system should specify that it is a local file system and the mount boot flag is `no`.
- The `PathPrefix` directory (the directory used by HA-NFS to maintain administrative and status information) is on the root directory of the same file system to be mounted (for example, `/global/local-fs/nfs`).

1. Become superuser on a cluster member.

2. Determine whether the resource type is registered.

The following command prints a list of registered resource types.

```
# scrgadm -p | egrep Type
```


3. If you need to, register the resource type.

```
# scrgadm -a -t SUNW.nfs
```

4. Create the failover resource group nfs-r

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/local-fs/nfs
```

5. Create a logical host resource of type SUNW.LogicalHostname.

```
# scrgadm -a -j nfs-lh-rs -g nfs-rg -L -l log-nfs
```

6. Register the HASStoragePlus resource type with the cluster.

```
# scrgadm -a -t SUNW.HASStoragePlus
```

7. Create the resource nfs-hastp-rs of type HASStoragePlus.

```
# scrgadm -a -j nfs-hastp-rs -g nfs-rg -t SUNW.HASStoragePlus\  
-x FilesystemMountPoints=/global/local-fs/nfs \  
-x AffinityOn=TRUE
```

Note – The FilesystemMountPoints extension property can be used to specify a list of one or more file system mount points. This list can consist of both local and global file system mount points. The mount at boot flag is ignored by HASStoragePlus for global file systems.

8. Bring the resource group nfs-rg online on a cluster node.

This node will become the primary node for the /global/local-fs/nfs file system's underlying global device partition. The file system /global/local-fs/nfs will then be locally mounted on this node

```
# scswitch -Z -g nfs-rg
```

9. Register the SUNW.nfs resource type with the cluster. Create the resource nfs-rs of type SUNW.nfs and specify its resource dependency on the resource nfs-hastp-rs.

dfstab.nfs-rs will be present in /global/local-fs/nfs/SUNW.nfs.

```
# scrgadm -a -t SUNW.nfs  
# scrgadm -a -g nfs-rg -j nfs-rs -t SUNW.nfs \  
-y Resource_dependencies=nfs-hastp-rs
```

Note – The nfs-hastp-rs resource must be online before you can set the dependency in the nfs resource.

10. Bring the resource nfs-rs online.

```
# scswitch -Z -g nfs-rg
```



Caution – Be sure to switch only at the resource group level. Switching at the device group level will confuse the resource group causing it to failover.

Now whenever the service is migrated to a new node, the primary I/O path for `/global/local-fs/nfs` will always be online and collocated with the NFS servers. The file system `/global/local-fs/nfs` will be locally mounted before starting the NFS server.

Freeing Node Resources by Offloading Non-critical Resource Groups

Prioritized Service Management (RGOffload) allows your cluster to automatically free a node's resources for critical data services. RGOffload is used when the startup of a critical failover data service requires a Non-Critical, scalable or failover data service to be brought offline. RGOffload is used to offload resource groups containing non-critical data services.

Note – The critical data service must be a failover data service. The data service to be offloaded can be a failover or scalable data service.

▼ How to Set Up an RGOffload Resource

1. **Become superuser on a cluster member.**
2. **Determine whether the RGOffload resource type is registered.**

The following command prints a list of resource types.

```
# scrgadm -p | egrep SUNW.RGOffload
```

3. **If needed, register the resource type**

```
# scrgadm -a -t SUNW.RGOffload
```

.

4. Set the `Desired primaries` to zero in each resource group to be offloaded by the RGOffload resource.

```
# scrgadm -c -g offload-rg -y Desired_primaries=0
```

5. Add the RGOffload resource to the critical failover resource group and set the extension properties.

Do not place a resource group on more than one resource's `rg_to_offload` list. Placing a resource group on multiple `rg_to_offload` lists may cause the resource group to be taken offline and brought back online repeatedly.

See "Configuring RGOffload Extension Properties" on page 84 for extension property descriptions.

```
# scrgadm -aj rgoffload-resource\  
-t SUNW.RGOffload -g critical-rg \  
-x rg_to_offload=offload-rg-1,offload-rg-2,...\  
-x continue_to_offload=TRUE \  
-x max_offload_retry=15
```

Note – Extension properties other than `rg_to_offload` are shown with default values here. `rg_to_offload` is a comma-separated list of resource groups that are not dependent on each other. This list cannot include the resource group to which the RGOffload resource is being added.

6. Enable the RGOffload resource.

```
# scswitch -ej rgoffload-resource
```

7. Set the dependency of the critical failover resource on the RGOffload resource.

```
# scrgadm -c -j critical-resource \  
-y Resource_dependencies=rgoffload-resource
```

`Resource_dependencies_weak` may also be used. Using `Resource_dependencies_weak` on the RGOffload resource type will allow the critical failover resource to start up even if errors are encountered during offload of `offload-rg`.

8. Bring the resource group to be offloaded online.

```
# scswitch -z -g offload-rg,offload-rg-2,... -h [nodelist]
```

The resource group remains online on all nodes where the critical resource group is offline. The fault monitor prevents the resource group from running on the node where the critical resource group is online.

Because `Desired primaries` for resource groups to be offloaded is set to 0 (see Step 4), the "-Z" option will not bring these resource groups online.

9. If the critical failover resource group is not online, bring it online.

```
# scswitch -Z -g critical-rg
```

Example – Configuring an RGOffload Resource

This example describes how to configure an RGOffload resource (`rgofl`), the critical resource group that contains the RGOffload resource (`oracle_rg`), and scalable resource groups that are offloaded when the critical resource group comes online (`IWS-SC`, `IWS-SC-2`). The critical resource in this example is `oracle-server-rs`.

In this example, `oracle_rg`, `IWS-SC`, and `IWS-SC-2` can be mastered on any node of cluster "triped": `phys-triped-1`, `phys-triped-2`, `phys-triped-3`.

[Determine whether the `SUNW.RGOffload` resource type is registered.]

```
# scrgadm -p | egrep SUNW.RGOffload
```

[If needed, register the resource type.]

```
# scrgadm -a -t SUNW.RGOffload
```

[Set the `Desired primaries` to zero in each resource group to be offloaded by the RGOffload resource.]

```
# scrgadm -c -g IWS-SC-2 -y Desired primaries=0
```

```
# scrgadm -c -g IWS-SC -y Desired primaries=0
```

[Add the RGOffload resource to the critical resource group and set the extension properties.]

```
# scrgadm -aj rgofl -t SUNW.RGOffload -g oracle_rg \  
-x rg_to_offload=IWS-SC,IWS-SC-2 -x continue_to_offload=TRUE \  
-x max_offload_retry=15
```

[Enable the RGOffload resource.]

```
# scswitch -ej rgofl
```

[Set the dependency of the critical failover resource to the RGOffload resource.]

```
# scrgadm -c -j oracle-server-rs -y Resource_dependencies=rgofl
```

[Bring the resource groups to be offloaded online on all nodes.]

```
# scswitch -z -g IWS-SC,IWS-SC-2 -h phys-triped-1,phys-triped-2,phys-triped-3
```

[If the critical failover resource group is not online, bring it online.]

```
# scswitch -Z -g oracle_rg
```

Configuring RGOffload Extension Properties

Typically, you use the command line `scrgadm -x parameter=value` to configure extension properties when you create the RGOffload resource. See Appendix A for details on all of the Sun Cluster standard properties.

Table 2–2 describes extension properties that you can configure for RGOffload. The Tunable entries indicate when you can update the property.

TABLE 2–2 RGOffload Extension Properties

Name/Data Type	Default
<code>rg_to_offload</code> (string)	<p>A comma-separated list of resource groups that need to be offloaded on a node when a critical failover resource group starts up on that node. This list should not contain resource groups that depend upon each other. This property has no default and must be set.</p> <p>RGOffload does not check for dependency loops in the list of resource groups set in the <code>rg_to_offload</code> extension property. For example, if resource group RG-B depends in some way on RG-A, then both RG-A and RG-B should not be included in <code>rg_to_offload</code>.</p> <p>Default: None</p> <p>Tunable: Any time</p>
<code>continue_to_offload</code> (Boolean)	<p>A Boolean to indicate whether to continue offloading the remaining resource groups in the <code>rg_to_offload</code> list after an error in offloading a resource group occurs.</p> <p>This property is only used by the START method.</p> <p>Default: True</p> <p>Tunable: Any time</p>

TABLE 2-2 RGOffload Extension Properties (Continued)

Name/Data Type	Default
max_offload_retry (integer)	<p>The number of attempts to offload a resource group during startup in case of failures due to cluster or resource group reconfiguration. There is an interval of 10 seconds between successive retries.</p> <p>Set the max_offload_retry so that (the number of resource groups to be offloaded * max_offload_retry * 10 seconds) is less than the start_timeout for the RGOffload resource. If this number is close to or more than the start_timeout number, the START method of RGOffload resource may time out before maximum offload attempts are completed.</p> <p>This property is only used by the START method.</p> <p>Default: 15</p> <p>Tunable: Any time</p>

Fault Monitor

The Fault Monitor probe for RGOffload resource is used to keep resource groups specified in the rg_to_offload extension property offline on the node mastering the critical resource. During each probe cycle, Fault Monitor verifies that resource groups to be offloaded (offload-rg) are offline on the node mastering the critical resource. If the offload-rg is online on the node mastering the critical resource, the Fault Monitor attempts to start offload-rg on nodes other than the node mastering the critical resource, thereby bringing offload-rg offline on the node mastering the critical resource.

Because desired primaries for offload-rg is set to 0, off-loaded resource groups are not restarted on nodes that become available later. Therefore, the RGOffload Fault Monitor attempts to start up offload-rg on as many primaries as possible, until maximum primaries limit is reached, while keeping offload-rg offline on the node mastering the critical resource.

RGOffload attempts to start up all offloaded resource groups unless they are in the MAINTENANCE or UNMANAGED state. To place a resource group in an UNMANAGED state, use the scswitch command.

```
# scswitch -u -g resourcegroup
```

The Fault Monitor probe cycle is invoked after every Thorough_probe_interval.

Standard Properties

This appendix describe the standard resource type, resource group, and resource properties. This appendix also describes the resource property attributes available for changing system-defined properties and creating extension properties.

This appendix contains the following sections.

- “Resource Type Properties” on page 89
- “Resource Properties” on page 96
- “Resource Group Properties” on page 109
- “Resource Property Attributes” on page 116

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

Resource Type Properties

Table A-1 describes the resource type properties defined by Sun Cluster. The property values are categorized as follows.

- **Required** — The property requires an explicit value in the Resource Type Registration (RTR) file or the object that it belongs to cannot be created. A blank or the empty string is not allowed as a value.
- **Conditional** — To exist, the property must be declared in the RTR file; otherwise, the Resource Group Manager (RGM) does not create it and it is not available to administrative utilities. A blank or the empty string is allowed. If the property is declared in the RTR file but no value is specified, the RGM supplies a default value.

- **Conditional/Explicit** — To exist, the property must be declared in the RTR file with an explicit value; otherwise, the RGM does not create it and it is not available to administrative utilities. A blank or the empty string is not allowed.
- **Optional** — The property can be declared in the RTR file;; if it isn't, the RGM creates it and supplies a default value. If the property is declared in the RTR file but no value is specified, the RGM supplies the same default value as if the property were not declared in the RTR file.

Resource type properties are not updatable by administrative utilities with the exception of `Installed_nodes`, which cannot be declared in the RTR file and must be set by the administrator.

TABLE A-1 Resource Type Properties

Property Name	Description
API_version (integer)	The version of the resource management API used by this resource type implementation. Category: Optional Default: 2 Tunable: Never
BOOT (string)	An optional callback method. The path to the program that the RGM starts on a node, which joins or rejoins the cluster when a resource of this type is already managed. This method is expected to initialize resources of this type similar to the INIT method. Category: Conditional/Explicit Default: No Default Tunable: Never
Failover (Boolean)	True indicates that resources of this type cannot be configured in any group that can be online on multiple nodes at once. Category: Optional Default: False Tunable: Never

TABLE A-1 Resource Type Properties (Continued)

Property Name	Description
FINI (string)	<p>An optional callback method. The path to the program that the RGM starts when a resource of this type is removed from RGM management.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
INIT (string)	<p>An optional callback method. The path to the program that the RGM starts when a resource of this type becomes managed by the RGM.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
Init_nodes (enum)	<p>Indicates the nodes on which the RGM is to call the INIT, FINI, BOOT and VALIDATE methods. The values can be <code>RG primaries</code> (just the nodes that can master the resource) or <code>RT installed nodes</code> (all nodes on which the resource type is installed).</p> <p>Category: Optional</p> <p>Default: <code>RG primaries</code></p> <p>Tunable: Never</p>
Installed_nodes (string array)	<p>A list of the cluster node names that the resource type is allowed to be run on. The RGM automatically creates this property. The cluster administrator can set the value. You cannot declare this property in the RTR file.</p> <p>Category: Configurable by cluster administrator</p> <p>Default: All cluster nodes</p> <p>Tunable: Any time</p>

TABLE A-1 Resource Type Properties (Continued)

Property Name	Description
Monitor_check (string)	<p>An optional callback method. The path to the program that the RGM starts before performing a monitor-requested failover of a resource of this type.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
Monitor_start (string)	<p>An optional callback method. The path to the program that the RGM activates to start a fault monitor for a resource of this type.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
Monitor_stop (string)	<p>A callback method that is required if Monitor_start is set. The path to the program that the RGM activates to stop a fault monitor for a resource of this type.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
Num_resource_restart on each cluster node (integer)	<p>This property is set by the RGM to the number of scha_control RESTART calls that have been made for this resource in this node within the past <i>n</i> seconds, where <i>n</i> is the value of the Retry_interval property of the resource. If a resource type does not declare the Retry_interval property, then the Num_resource_restarts property is not available for resources of that type.</p>
Pkglist (string array)	<p>An optional list of packages that are included in the resource type installation.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>

TABLE A-1 Resource Type Properties (Continued)

Property Name	Description
Postnet_stop (string)	<p>An optional callback method. The path to the program that the RGM starts after calling the STOP method of any network-address resources (<code>Network_resources_used</code>) that a resource of this type is dependent on. This method is expected to do STOP actions that must be done after the network interfaces are configured down.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
Prenet_start (string)	<p>An optional callback method. The path to the program that the RGM starts before calling the START method of any network-address resources (<code>Network_resources_used</code>) that a resource of this type is dependent on. This method is expected to do START actions that must be done before network interfaces are configured up.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
RT_basedir (string)	<p>The directory path that is used to complete relative paths for callback methods. This path is expected to be set to the installation location for the resource type packages. It must be a complete path, that is, it must start with a forward slash (/). This property is not required if all the method path names are absolute.</p> <p>Category: Required (unless all method path names are absolute)</p> <p>Default: No Default</p> <p>Tunable: Never</p>
RT_description (string)	<p>A brief description of the resource type.</p> <p>Category: Conditional</p> <p>Default: The empty string</p> <p>Tunable: Never</p>

TABLE A-1 Resource Type Properties (Continued)

Property Name	Description
Resource_type (string)	<p>The name of the resource type.</p> <p>To view the names of the currently registered resource types, use:</p> <pre>scrgadm -p</pre> <p>Starting in Sun Cluster 3.1, a resource type name is of the form</p> <pre>vendor_id.resource_type:version</pre> <p>The three components of the resource type name are properties specified in the RTR file as <i>Vendor_id</i>, <i>Resource_type</i>, and <i>RT_version</i>; the <code>scrgadm</code> command inserts the period and colon delimiters. The <code>RT_version</code> suffix of the resource type name is the same value as the <code>RT_version</code> property.</p> <p>To ensure that the <i>Vendor_id</i> is unique, the recommended approach is the use the stock symbol for the company creating the resource type.</p> <p>Resource type names created prior to Sun Cluster 3.1 continue to be of the form:</p> <pre>vendor_id.resource_type</pre> <p>Category: Required</p> <p>Default: The empty string</p> <p>Tunable: Never</p>
RT_version (string)	<p>Starting with Sun Cluster 3.1, a required version of this resource type implementation. The <code>RT_version</code> is the suffix component of the full resource type name.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>

TABLE A-1 Resource Type Properties (Continued)

Property Name	Description
Single_instance (Boolean)	<p>If True, indicates that only one resource of this type can exist in the cluster. Hence, the RGM allows only one resource of this type to run cluster-wide at one time.</p> <p>Category: Optional</p> <p>Default: False</p> <p>Tunable: Never</p>
START (string)	<p>A callback method. The path to the program that the RGM activates to start a resource of this type.</p> <p>Category: Required (unless the RTR file declares a PRENET_START method)</p> <p>Default: No Default</p> <p>Tunable: Never</p>
STOP (string)	<p>A callback method. The path to the program that the RGM activates to stop a resource of this type.</p> <p>Category: Required (unless the RTR file declares a POSTNET_STOP method)</p> <p>Default: No Default</p> <p>Tunable: Never</p>
UPDATE (string)	<p>An optional callback method. The path to the program that the RGM starts when properties of a running resource of this type are changed.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>
VALIDATE (string)	<p>An optional callback method. The path to the program that will be invoked to check values for properties of resources of this type.</p> <p>Category: Conditional/Explicit</p> <p>Default: No Default</p> <p>Tunable: Never</p>

TABLE A-1 Resource Type Properties (Continued)

Property Name	Description
Vendor_ID (string)	See the Resource_type property. Category: Conditional Default: No Default Tunable: Never

Resource Properties

Table A-2 describes the resource properties defined by Sun Cluster. These descriptions have been developed for data service developers. For more information about a particular data service, see that data service's man page. Resource property values are categorized as follows.

- **Required** — The administrator must specify a value when creating a resource with an administrative utility.
- **Optional** — If the administrator does not specify a value when creating a resource group, the system supplies a default value.
- **Conditional** — The RGM creates the property only if the property is declared in the RTR file. Otherwise, the property does not exist and is not available to system administrators. A conditional property declared in the RTR file is optional or required, depending on whether a default value is specified in the RTR file. For details, see the description of each conditional property.
- **Query-only** — Cannot be set directly by an administrative tool.

Table A-2 also lists whether and when resource properties are tunable, as follows.

None or False	Never.
True or Anytime	Any time.
At_creation	When the resource is added to a cluster.
When_disabled	When the resource is disabled.

TABLE A-2 Resource Properties

Property Name	Description
<p>Affinity_timeout (integer)</p>	<p>Controls the number of seconds IP affinity lasts after a client closes all its connections to sticky scalable service. Before this timeout period expires, any new connection from the client will be forwarded to the same cluster node. A 0 value means affinity lasts for as long as there is a connection between a client and the scalable service. A -1 value means affinity lasts until the cluster node to which the client is bound crashes or reboots. A -1 value means no timeout period applies.</p> <p>This property only applies to resources with <code>Load_balancing_policy=LB_STICKY</code> or <code>LB_STICKYWILD</code>.</p> <p>Category: Optional</p> <p>Default: 0</p> <p>Tunable: Any time</p>
<p>Cheap_probe_interval (integer)</p>	<p>The number of seconds between invocations of a quick fault probe of the resource. This property is only created by the RGM and available to the administrator if it is declared in the RTR file.</p> <p>This property is optional if a default value is specified in the RTR file. If the <code>Tunable</code> attribute is not specified in the resource type file, the <code>Tunable</code> value for the property is <code>When_disabled</code>.</p> <p>This property is required if the <code>Default</code> attribute is not specified in the property declaration in the RTR file.</p> <p>Category: Conditional</p> <p>Default: See above</p> <p>Tunable: When disabled</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Extension properties	<p>The developer declares the resource type properties in the initial configuration of the data service at the time the cluster administrator registers the data service with Sun Cluster. For information on the individual attributes you can set for extension properties, see Table A-4.</p> <p>Category: Conditional</p> <p>Default: No Default</p> <p>Tunable: Depends on the specific property</p>
Failover_mode (enum)	<p>Possible settings are None, Soft, and Hard. Controls whether the RGM relocates a resource group or aborts a node in response to a failure of a <i>START</i> or <i>STOP</i> method call on the resource. <i>None</i> indicates that the RGM should just set the resource state on method failure and wait for operator intervention. <i>Soft</i> indicates that failure of a <i>START</i> method should cause the RGM to relocate the resource's group to a different node while failure of a <i>STOP</i> method should cause the RGM to set the resource state and wait for operator intervention. <i>Hard</i> indicates that failure of a <i>START</i> method should cause the relocation of the group and failure of a <i>STOP</i> method should cause the forcible stop of the resource by aborting the cluster node.</p> <p>Category: Optional</p> <p>Default: No Default</p> <p>Tunable: Any time</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Load_balancing_policy (string)	<p>A string that defines the load-balancing policy in use. This property is used only for scalable services. The RGM automatically creates this property if the Scalable property is declared in the RTR file.</p> <p>Load_balancing_policy can take the following values:</p> <p>Lb_weighted (the default). The load is distributed among various nodes according to the weights set in the Load_balancing_weights property.</p> <p>Lb_sticky. A given client (identified by the client's IP address) of the scalable service is always sent to the same node of the cluster.</p> <p>Lb_sticky_wild. A given client (identified by the client's IP address), who connects to an IP address of a wildcard sticky service, is always sent to the same cluster node regardless of the port number it is coming to.</p> <p>Category: Conditional/Optional</p> <p>Default: Lb_weighted</p> <p>Tunable: At creation</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Load_balancing_weights (string array)	<p>For scalable resources only. The RGM automatically creates this property if the Scalable property is declared in the RTR file. The format is <i>weight@node,weight@node</i>, where <i>weight</i> is an integer that reflects the relative portion of load distributed to the specified <i>node</i>. The fraction of load distributed to a node is the weight for this node divided by the sum of all weights. For example, 1@1, 3@2 specifies that node 1 receives 1/4 of the load and node 2 receives 3/4. The empty string (""), the default, sets a uniform distribution. Any node that is not assigned an explicit weight, receives a default weight of 1.</p> <p>If the Tunable attribute is not specified in the resource type file, the Tunable value for the property is Anytime. Changing this property revises the distribution for new connections only.</p> <p>Category: Conditional/Optional</p> <p>Default: The empty string</p> <p>Tunable: Any time</p>
method_timeout for each callback method (integer)	<p>A time lapse, in seconds, after which the RGM concludes that an invocation of the method has failed.</p> <p>Category: Conditional/Optional</p> <p>Default: 3,600 (one hour) if the method itself is declared in the RTR file.</p> <p>Tunable: Any time</p>
Monitored_switch (enum)	<p>Set to Enabled or Disabled by the RGM if the cluster administrator enables or disables the monitor with an administrative utility. If Disabled, the monitor does not have its START method called until it is enabled again. If the resource does not have a monitor callback method, this property does not exist.</p> <p>Category: Query-only</p> <p>Default: Enabled</p> <p>Tunable: Never</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
<p>Network_resources_used (string array)</p>	<p>A comma-separated list of logical host name or shared address network resources used by the resource. For scalable services, this property must refer to shared address resources that exist in a separate resource group. For failover services, this property refers to logical host name or shared address resources that exist in the same resource group. The RGM automatically creates this property if the Scalable property is declared in the RTR file. If Scalable is not declared in the RTR file, Network_resources_used is unavailable unless it is explicitly declared in the RTR file.</p> <p>If the Tunable attribute is not specified in the RTR file, the Tunable value for the property is At_creation.</p> <p>Category: Conditional/Required</p> <p>Default: No Default</p> <p>Tunable: At creation</p>
<p>On_off_switch (enum)</p>	<p>Set to Enabled or Disabled by the RGM if the cluster administrator enables or disables the resource with an administrative utility. If disabled, a resource has no callbacks invoked until it is enabled again.</p> <p>Category: Query-only</p> <p>Default: Disabled</p> <p>Tunable: Never</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Port_list (string array)	<p>A comma-separated list of port numbers on which the server is listening. Appended to each port number is the protocol being used by that port, for example, Port_list=80/tcp. If the Scalable property is declared in the RTR file, the RGM automatically creates Port_list; otherwise, this property is unavailable unless it is explicitly declared in the RTR file.</p> <p>For specifics on setting up this property for Apache, see the Apache chapter in the <i>Sun Cluster 3.1 Data Service Planning and Administration Guide</i>.</p> <p>Category: Conditional/Required Default: No Default Tunable: At creation</p>
R_description (string)	<p>A brief description of the resource.</p> <p>Category: Optional Default: The empty string Tunable: Any time</p>
Resource_dependencies (string array)	<p>A comma-separated list of resources in the same group that must be online in order for this resource to be online. This resource cannot be started if the start of any resource in the list fails. When bringing the group offline, this resource is stopped before those in the list. Resources in the list are not allowed to be disabled unless this resource is disabled first.</p> <p>Category: Optional Default: The empty list Tunable: Any time</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Resource_dependencies_weak (string array)	<p>A list of resources in the same group that determines the order of method calls within the group. The RGM calls the start methods of the resources in this list before the start method of this resource and the stop methods of this resource before the stop methods of those in the list. The resource can still be online if those in the list fail to start or are disabled.</p> <p>Category: Optional</p> <p>Default: The empty list</p> <p>Tunable: Any time</p>
Resource_name (string)	<p>The name of the resource instance. Must be unique within the cluster configuration and cannot be changed after a resource has been created.</p> <p>Category: Required</p> <p>Default: No Default</p> <p>Tunable: Never</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Resource_project_name(string)	<p>The Solaris project name associated with the resource. Use this property to apply Solaris resource management features such as CPU shares and resource pools to cluster data services. When the RGM brings resources online, it launches the related processes under this project name. If this property is not specified, the project name will be taken from the <code>RG_project_name</code> property of the resource group that contains the resource. If neither property is specified, the RGM will use the predefined project name <code>default</code>. The specified project name must exist in the projects database (see appropriate SRM man page), and the user <code>root</code> must be configured as a member of the named project. This property is supported in Solaris 9 and later releases. For more information on Solaris project names, see <i>System Administration Guide: Resource Management and Network Services</i>.</p> <p>Category: Optional Default: see above Tunable: Any time</p>
Resource_state: on each cluster node (enum)	<p>The RGM-determined state of the resource on each cluster node. Possible states are: <code>ONLINE</code>, <code>OFFLINE</code>, <code>STOP_FAILED</code>, <code>START_FAILED</code>, <code>MONITOR_FAILED</code>, and <code>ONLINE_NOT_MONITORED</code>.</p> <p>This property is not user configurable.</p> <p>Category: Query-only Default: No Default Tunable: Never</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
<p>Retry_count (integer)</p>	<p>The number of times a monitor attempts to restart a resource if it fails. This property is created by the RGM only and available to the administrator if it is declared in the RTR file. It is optional if a default value is specified in the RTR file.</p> <p>If the Tunable attribute is not specified in the resource type file, the Tunable value for the property is When_disabled.</p> <p>This property is required if the Default attribute is not specified in the property declaration in the RTR file.</p> <p>Category: Conditional</p> <p>Default: See above</p> <p>Tunable: When disabled</p>
<p>Retry_interval (integer)</p>	<p>The number of seconds over which to count attempts to restart a failed resource. The resource monitor uses this property in conjunction with Retry_count. This property is created by the RGM only and available to the administrator if it is declared in the RTR file. It is optional if a default value is specified in the RTR file.</p> <p>If the Tunable attribute is not specified in the resource type file, the Tunable value for the property is When_disabled.</p> <p>This property is required if the Default attribute is not specified in the property declaration in the RTR file.</p> <p>Category: Conditional</p> <p>Default: See above</p> <p>Tunable: When disabled</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Scalable (Boolean)	<p>Indicates whether the resource is scalable. If this property is declared in the RTR file, the RGM automatically creates the scalable service properties <code>Network_resources_used</code>, <code>Port_list</code>, <code>Load_balancing_policy</code>, and <code>Load_balancing_weights</code> for resources of that type. These properties have their default values unless they are explicitly declared in the RTR file. The default for <code>Scalable</code>—when it is declared in the RTR file—is <code>True</code>.</p> <p>When this property is declared in RTR file, the <code>Tunable</code> attribute must be set to <code>At_creation</code> or resource creation fails.</p> <p>If this property is not declared in the RTR file, the resource is not scalable, the cluster administrator cannot tune this property, and no scalable service properties are set by the RGM. However, you can explicitly declare the <code>Network_resources_used</code> and <code>Port_list</code> properties in the RTR file, if desired, because they can be useful in a non-scalable service as well as in a scalable service.</p> <p>Category: Optional Default: See above Tunable: At creation</p>
Status: on each cluster node (enum)	<p>Set by the resource monitor. Possible values include <code>OK</code>, <code>degraded</code>, <code>faulted</code>, <code>unknown</code>, and <code>offline</code>. The RGM sets the value to <code>unknown</code> when the resource is brought online and to <code>Offline</code> when it is brought offline.</p> <p>Category: Query-only Default: No Default Tunable: Never</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
Status_msg: on each cluster node (string)	<p>Set by the resource monitor at the same time as the Status property. This property is tunable per resource per node. The RGM sets it to the empty string when the resource is brought offline.</p> <p>Category: Query-only</p> <p>Default: No Default</p> <p>Tunable: Never</p>
Thorough_probe_interval (integer)	<p>The number of seconds between invocations of a high-overhead fault probe of the resource. This property is created by the RGM only and available to the administrator if it is declared in the RTR file. It is optional if a default value is specified in the RTR file.</p> <p>If the Tunable attribute is not specified in the resource type file, the Tunable value for the property is When_disabled.</p> <p>This property is required if the Default attribute is not specified in the property declaration in the RTR file.</p> <p>Category: Conditional</p> <p>Default: No Default</p> <p>Tunable: When disabled</p>
Type (string)	<p>An instance's resource type.</p> <p>Category: Required</p> <p>Default: No Default</p> <p>Tunable: Never</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
<p>Type_version (string)</p>	<p>Specifies which version of the resource type is currently associated with this resource. The RGM automatically creates this property, which cannot be declared in the RTR file. The value of this property is equal to the RT_version property of the resource's type. When a resource is created, the Type_version property is not specified explicitly, though it may appear as a suffix of the resource type name. When a resource is edited, the Type_version may be changed to a new value.</p> <p>The RT_Version is used for upgrading from one version of a data service to another version of the data service.</p> <p>Its tunability is derived from:</p> <ul style="list-style-type: none"> ■ The current version of the resource type The #upgrade_from directive in the *RTR file
<p>Udp_affinity (Boolean)</p>	<p>If TRUE, UDP packets from a given client are forwarded to the same cluster server node the same way TCP connections are forwarded. If a client's TCP connections are "stuck" with a server node, its UDP packets are sent to that node as well. If FALSE, this behavior is not maintained, and UDP packets from a given client could end up on a different node than TCP connections from that client.</p> <p>This property only applies to resources with Load_balancing_policy=LB_STICKY or LB_STICKYWILD.</p> <p>Category: Optional</p> <p>Default: False</p> <p>Tunable: When disabled</p>

TABLE A-2 Resource Properties (Continued)

Property Name	Description
<p><code>Weak_affinity</code> (Boolean)</p>	<p>Provides IP affinity for sticky scalable services, except it is subject to disruptions under the following cases: (1) When a server listener starts up, for example, due to a fault monitor restart, resource failover and switchover, or when a node rejoins the cluster after failover. (2) When <code>load_balancing_weights</code> for the scalable resource changes due to an administrative action. Provides a low overhead alternative to the default form of IP affinity in terms of memory consumption and processor cycles. Does not use the affinity timeout value. <code>Weak_affinity</code> lasts for as long as none of the above listed disruptions occurs.</p> <p>This property only applies to resources with <code>Load_balancing_policy=LB_STICKY</code> or <code>LB_STICKYWILD</code>.</p> <p>Category: Optional Default: <code>False</code> Tunable: When disabled</p>

Resource Group Properties

Table A-3 describes the resource group properties defined by Sun Cluster.

TABLE A-3 Resource Group Properties

Property Name	Description
Auto_start_on_new_cluster (Boolean)	<p>This property controls whether the Resource Group Manager starts the resource group automatically when a new cluster is forming.</p> <p>The default is TRUE. If set to TRUE, the Resource Group Manager attempts to start the resource group automatically to achieve <code>Desired primaries</code> when all the nodes of the cluster are simultaneously rebooted. If set to FALSE, the Resource Group does not start automatically when the cluster is rebooted.</p> <p>Category: Optional</p> <p>Default: True</p> <p>Tunable: Any time</p>
Desired primaries (integer)	<p>The desired number of nodes that the group can run on simultaneously.</p> <p>The default is 1. If the <code>RG_mode</code> property is <code>Failover</code>, the value of this property must be no greater than 1. If the <code>RG_mode</code> property is <code>Scalable</code>, a value greater than 1 is allowed.</p> <p>Category: Optional</p> <p>Default: 1, see above</p> <p>Tunable: Any time</p>
Failback (Boolean)	<p>A Boolean value that indicates whether to recalculate the set of nodes where the group is online when the cluster membership changes. A recalculation can cause the RGM to bring the group offline on less preferred nodes and online on more preferred nodes.</p> <p>Category: Optional</p> <p>Default: False</p> <p>Tunable: Any time</p>

TABLE A-3 Resource Group Properties (Continued)

Property Name	Description
Global_resources_used (string array)	<p>Indicates whether cluster file systems are used by any resource in this resource group. Legal values that the administrator can specify are an asterisk (*) to indicate all global resources, and the empty string ("") to indicate no global resources.</p> <p>Category: Optional</p> <p>Default: All global resources</p> <p>Tunable: Any time</p>
Implicit_network_dependencies (Boolean)	<p>A Boolean value that indicates, when True, that the RGM should enforce implicit strong dependencies of non-network-address resources on network-address resources within the group. This means that the RGM starts all network-address resources before all other resources and stops network address resources after all other resources within the group. Network-address resources include the logical host name and shared address resource types.</p> <p>In a scalable resource group, this property has no effect because a scalable resource group does not contain any network-address resources.</p> <p>Category: Optional</p> <p>Default: True</p> <p>Tunable: When disabled</p>
Maximum primaries (integer)	<p>The maximum number of nodes where the group might be online at once.</p> <p>The default is 1. If the RG_mode property is Failover, the value of this property must be no greater than 1. If the RG_mode property is Scalable, a value greater than 1 is allowed.</p> <p>Category: Optional</p> <p>Default: 1, see above</p> <p>Tunable: Any time</p>

TABLE A-3 Resource Group Properties (Continued)

Property Name	Description
Nodelist (string array)	<p>A comma-separated list of cluster nodes where the group can be brought online in order of preference. These nodes are known as the potential primaries or masters of the resource group.</p> <p>Category: Optional</p> <p>Default: The list of all cluster nodes in arbitrary order</p> <p>Tunable: Any time</p>
Pathprefix (string)	<p>A directory in the cluster file system that resources in the group can write essential administrative files in. Some resources might require this property. Make Pathprefix unique for each resource group.</p> <p>Category: Optional</p> <p>Default: The empty string</p> <p>Tunable: Any time</p>

TABLE A-3 Resource Group Properties (Continued)

Property Name	Description
Pingpong_interval (integer)	<p>A non-negative integer value (in seconds) used by the RGM to determine where to bring the resource group online in the event of a reconfiguration or as the result of an <code>scha_control giveover</code> command or function being executed.</p> <p>In the event of a reconfiguration, if the resource group fails to come online more than once within the past <code>Pingpong_interval</code> seconds on a particular node (because the resource's <code>START</code> or <code>PRENET_START</code> method exited non-zero or timed out), that node is considered ineligible to host the resource group and the RGM looks for another master.</p> <p>If a call to a resource's <code>scha_control(1ha)</code> (<code>3ha</code>) command or function causes the resource group to be brought offline on a particular node within the past <code>Pingpong_interval</code> seconds, that node is ineligible to host the resource group as the result of a subsequent call to <code>scha_control</code> originating from another node.</p> <p>Category: Optional Default: 3,600 (one hour) Tunable: Any time</p>
Resource_list (string array)	<p>The list of resources that are contained in the group. The administrator does not set this property directly. Rather, the RGM updates this property when the administrator adds or removes resources from the resource group.</p> <p>Category: Query-only Default: The empty list Tunable: Never</p>

TABLE A-3 Resource Group Properties (Continued)

Property Name	Description
RG_dependencies (string array)	<p>A comma-separated list of resource groups that this group depends on. This list indicates a preferred order for bringing other groups online or offline on the same node. It has no effect if the groups are brought online on different nodes.</p> <p>Category: Optional</p> <p>Default: The empty list</p> <p>Tunable: Any time</p>
RG_description (string)	<p>A brief description of the resource group.</p> <p>Category: Optional</p> <p>Default: The empty string</p> <p>Tunable: Any time</p>
RG_mode (enum)	<p>Indicates whether the resource group is a failover or scalable group. If the value is <code>Failover</code>, the RGM sets the <code>Maximum primaries</code> property of the group to 1 and restricts the resource group to being mastered by a single node.</p> <p>If the value of this property is <code>Scalable</code>, the RGM allows the <code>Maximum primaries</code> property to have a value greater than 1, meaning the group can be mastered by multiple nodes simultaneously.</p> <p>Note: The RGM does not allow a resource whose <code>Failover</code> property is <code>True</code> to be added to a resource group whose <code>RG_mode</code> is <code>Scalable</code>.</p> <p>Category: Optional</p> <p>Default: <code>Failover</code> if <code>Maximum primaries</code> is 1</p> <p><code>Scalable</code> if <code>Maximum primaries</code> is greater than 1.</p> <p>Tunable: Never</p>

TABLE A-3 Resource Group Properties (Continued)

Property Name	Description
RG_name (string)	<p>The name of the resource group. This property is required and must be unique within the cluster.</p> <p>Category: Required</p> <p>Default: No Default</p> <p>Tunable: Never</p>
RG_project_name	<p>The Solaris project name associated with the resource group. Use this property to apply Solaris resource management features such as CPU shares and resource pools to cluster data services. When the RGM brings resource groups online, it launches the related processes under this project name for resources that do not have the Resource_project_name property set. The specified project name must exist in the projects database (see appropriate SRM man page), and the user root must be configured as a member of the named project. This property is supported in Solaris 9 and later releases. For more information on Solaris project names, see <i>System Administration Guide: Resource Management and Network Services</i>.</p> <p>Category: Optional</p> <p>Default: Default, see above</p> <p>Tunable: Any time</p>
RG_state: on each cluster node (enum)	<p>Set by the RGM to Online, Offline, Pending_online, Pending_offline or Error_stop_failed to describe the state of the group on each cluster node. A group can also exist in an UNMANAGED state when it is not under the control of the RGM.</p> <p>This property is not user configurable.</p> <p>Category: Query-only</p> <p>Default: Offline</p> <p>Tunable: Never</p>

Resource Property Attributes

Table A-4 describes the resource property attributes that can be used to change system-defined properties or create extension properties.



Caution – You cannot specify NULL or the empty string (“”) as the default value for boolean, enum, or int types.

TABLE A-4 Resource Property Attributes

Property	Description
Property	The name of the resource property.
Extension	If used, indicates that the RTR file entry declares an extension property defined by the resource type implementation. Otherwise, the entry is a system-defined property.
Description	A string annotation intended to be a brief description of the property. The description attribute cannot be set in the RTR file for system-defined properties.
Type of the property	Allowable types include <code>string</code> , <code>boolean</code> , <code>int</code> , <code>enum</code> , and <code>stringarray</code> . You cannot set the type attribute in an RTR file entry for system-defined properties. The type determines acceptable property values and the type-specific attributes that are allowed in the RTR file entry. An <code>enum</code> type is a set of string values.
Default	Indicates a default value for the property.
Tunable	Indicates when the cluster administrator can set the value of this property in a resource. Can be set to <code>None</code> or <code>False</code> to prevent the administrator from setting the property. Values that allow administrator tuning include <code>True</code> or <code>Anytime</code> (at any time), <code>At_creation</code> (only when the resource is created), or <code>When_disabled</code> (when the resource is offline). The default is <code>True (Anytime)</code> .
Enumlist	For an <code>enum</code> type, a set of string values permitted for the property.
Min	For an <code>int</code> type, the minimal value permitted for the property.
Max	For an <code>int</code> type, the maximum value permitted for the property.
Minlength	For <code>string</code> and <code>stringarray</code> types, the minimum string length permitted.

TABLE A-4 Resource Property Attributes (Continued)

Property	Description
Maxlength	For <code>string</code> and <code>stringarray</code> types, the maximum string length permitted.
Array_minsize	For <code>stringarray</code> type, the minimum number of array elements permitted.
Array_maxsize	For <code>stringarray</code> type, the maximum number of array elements permitted.

Legal RGM Names and Values

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

RGM Legal Names

RGM names fall into the following five categories.

- resource group names
- resource type names
- resource names
- property names
- enumeration literal names

Except for resource type names, all names must comply with the following rules.

- must be in ASCII
- must start with a letter
- can contain upper and lowercase letters, digits, dashes (-), and underscores (_)
- must not exceed 255 characters

A resource type name can be a simple name (specified by the `Resource_type` property in the RTR file) or a complete name (specified by the `Vendor_id` and `Resource_type` properties in the RTR file). When you specify both of these properties, the RGM inserts a period between the `Vendor_id` and `Resource_type` to form the complete name. For example, if `Vendor_id=SUNW` and `Resource_type=sample`, the complete name is `SUNW.sample`. This instance is the only case where a period is a legal character in an RGM name.

RGM Values

RGM values fall into two categories—property values and description values—both of which share the same rules, as follows.

- 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 any of the following characters.
 - null
 - newline
 - comma
 - semicolon

Index

A

- add
 - node to resource group, 66
 - failover, 67
 - scalable, 67
- adding
 - See* modifying
 - resources to a resource group, 40
 - failover application, 44
 - logical hostname, 41
 - scalable application, 46
 - shared address, 42
- Affinity_timeout, resource property, 97
- API_version, resource type property, 90
- application binaries, determining location, 15
- Array_maxsize, resource property
 - attribute, 116
- Array_minsize, resource property
 - attribute, 116
- Auto_start_on_new_cluster, resource group property, 110
- auxnodelist, node list property, 19

B

- BOOT, resource type property, 90
- bringing online, resource group, 48

C

- change
 - resource group properties, 61
 - resource properties, 62
 - resource type properties, 60
- changing
 - See* modifying
- Cheap_probe_interval, resource property, 97
- clear, STOP_FAILED error flag on resource, 63
- commands
 - prtconf -v, 11
 - prtdiag -v, 11
 - psrinfo -v, 11
 - showrev -p, 11
- configuration
 - overview, 20
- configuration, planning cluster file system, 16
- configuration
 - task map, 21
- configuration guidelines, 14
- configuring and administering, Sun Cluster
 - data services, 30
- considerations, 19
- Copy
 - resource group property, 110
- creating
 - resource group
 - failover, 37
 - scalable, 38

D

- data service resources, task map, 28
- data services
 - considerations, 19
 - planning, 13
 - special requirements, 15
- Default, resource property attribute, 116
- deleting
 - See* modifying
- Description
 - resource property attribute, 116
- Desired_primaries, resource group property, 110
- disabling
 - See* stopping
 - resource
 - and moving resource group to UNMANAGED state, 57
 - resource group monitors, 49
- disk device group
 - See also* node list
- disk device groups
 - relationship with resource groups, 16
 - synchronize startups with resource groups, 74
- display, resource type, resource group, and resource configuration, 59
- displaying
 - See* listing

E

- editing
 - See* modifying
- enabling, *See* starting, 7
 - resource group monitors, 49
- Enumlist, resource property attribute, 116
- executing
 - See* starting
- Extension, resource property attribute, 116
- Extension properties, resource property, 97
- extension properties
 - RGOffload
 - continue_to_offload, 86
 - max_offload_retry, 86
 - rg_to_offload, 86

F

- Failback, resource group property, 110
- Failover, resource type property, 90
- Failover_mode, resource property, 97
- fault monitor
 - overview, 24
 - process monitor facility (PMF), 25
 - RGOffload, 86
- FINI, resource type property, 90
- freeing node resources, RGOffload, 82

G

- Global_resources_uses, resource group property, 110

H

- HASStorage
 - determining whether your data service requires it, 18
 - overview, 17
 - versus HASStoragePlus, 19
- HASStoragePlus
 - determining whether your data service requires it, 18
 - overview, 17
 - versus HASStorage, 19
- highly available local file systems, setting up HASStoragePlus, 79

I

- Implicit_network_dependencies, resource group property, 110
- INIT, resource type property, 90
- Init_nodes, resource type property, 90
- installation
 - overview, 20
 - task map, 21
- installed_nodes, node list property, 19
- Installed_nodes, resource type property, 90

L

legal names, resource group manager, 119
Load_balancing_policy, resource property, 97
Load_balancing_weights, resource property, 97

M

Max, resource property attribute, 116
Maximum primaries, resource group property, 110
Maxlength, resource property attribute, 116
method_timeout, resource property, 97
migrating to a new resource type version, 33
Min, resource property attribute, 116
Minlength, resource property attribute, 116
Monitor_check, resource type property, 90
Monitor_start, resource type property, 90
Monitor_stop, resource type property, 90
Monitored_switch, resource property, 97

N

Network_resources_used, resource property, 97
node
 add
 to resource group, 66, 67
 remove
 form a resource group, 70, 71, 72
 from a resource group, 69
node list properties
 auxnodelist, 19
 installed_nodes, 19
 nodelist, 19
nodelist, node list property, 19
Nodelist, resource group property, 110
nsswitch.conf, verifying file contents, 16
Num_resources_restart, resource type property, 90

O

offloading non-critical resource groups, RGOffload, 82
On_off_switch, resource property, 97

P

Pathprefix, resource group property, 110
Pingpong_interval, resource group property, 110
Pkg_list, resource type property, 90
planning
 cluster file system configuration, 16
 data services, 13
PMF
 See process monitor facility (PMF)
Port_list, resource property, 97
Postnet_stop, resource type property, 90
Prenet_stop, resource type property, 90
printing
 See listing
process monitor facility (PMF), 25
Property, resource property attribute, 116
prtconf -v command, 11
psrinfo -v command, 11
-pv command, 11

R

R_description, resource property, 97
re-register, preregistered resource type, 65
registering, resource type, 31
remove
 node from a resource group, 69
 failover, 71
 scalable, 70
 node from a resource group containing shared address resources
 failover, 72
removing
 resource, 54
 resource group, 52
 resource type, 51
requirements, data services, 15

- resource
 - adding a failover application resource to a resource group, 44
 - adding a scalable application resource to a resource group, 46
 - adding a shared address resource to a resource group, 42
 - adding logical hostname resource to a resource group, 41
 - change properties, 62
 - change resource type properties, 60
 - clear
 - STOP_FAILED error flag, 63
 - disabling and moving resource group to UNMANAGED state, 57
 - display configuration information, 59
 - re-register resource type, 65
 - removing, 54
 - removing resource type, 51
- Resource_dependencies, resource property, 97
- Resource_dependencies_weak, resource property, 97
- resource fault monitor
 - disabling, 50
 - enabling, 50
- resource group
 - See also* node list
 - add
 - node, 66, 67
 - adding resources, 40
 - failover application, 44
 - logical hostname, 41
 - scalable application, 46
 - shared address, 42
 - bringing online, 48
 - change properties, 61
 - creating
 - failover, 37
 - scalable, 38
 - disabling
 - resource fault monitor, 50
 - disabling monitors, 49
 - display configuration information, 59
 - enabling
 - resource fault monitor, 50
 - enabling monitors, 49
- resource group (Continued)
 - remove
 - node, 69, 70, 71, 72
 - removing, 52
 - switching the current primary, 55
 - synchronize startups with disk device groups, 74
 - resource group manager
 - legal names, 119
 - values, 120
 - resource group properties
 - Auto_start_on_new_cluster, 110
 - Desired primaries, 110
 - Failback, 110
 - Global_resources_uses, 110
 - Implicit_network_dependencies, 110
 - Maximum primaries, 110
 - Nodelist, 110
 - Pathprefix, 110
 - Pingpong_interval, 110
 - Resource_list, 110
 - RG_dependencies, 110
 - RG_description, 110
 - RG_mode, 110
 - RG_name, 110
 - RG_project_name, 110
 - RG_state, 110
 - resource groups, relationship with disk device groups, 16
 - Resource_list, resource group property, 110
 - Resource_name, resource property, 97
 - Resource_project_name, resource property, 97
 - resource properties
 - Affinity_timeout, 97
 - Cheap_probe_interval, 97
 - Extension properties, 97
 - Failover_mode, 97
 - Load_balancing_policy, 97
 - Load_balancing_weights, 97
 - method_timeout, 97
 - Monitored_switch, 97
 - Network_resources_used, 97
 - On_off_switch, 97
 - Port_list, 97
 - R_description, 97
 - Resource_dependencies, 97

resource properties (Continued)

- Resource_dependencies_weak, 97
- Resource_name, 97
- Resource_project_name, 97
- Resource_state, 97
- Retry_count, 97
- Retry_interval, 97
- Scalable, 97
- Status, 97
- Status_message, 97
- Thorough_probe_interval, 97
- Type, 97
- Type_version, 97
- Udp_affinity, 97
- Weak_affinity, 97

resource property attributes

- Array_maxsize, 116
- Array_minsize, 116
- Default, 116
- Description, 116
- Enumlist, 116
- Extension, 116
- Max, 116
- Maxlength, 116
- Min, 116
- Minlength, 116
- Property, 116
- Tunable, 116
- Type of property, 116

Resource_state, resource property, 97

resource type

- change properties, 60
- display configuration information, 59
- HASStorage
 - existing resources, 76
 - new resources, 74
- migrating to a new resource type version, 33
- re-register, 65
- registering, 31
- removing, 51

Resource_type, resource type property, 90

resource type

- RGOffload, 82
- upgrading, 32

resource type properties

- API_version, 90
- BOOT, 90
- Failover, 90

resource type properties (Continued)

- FINI, 90
- INIT, 90
- Init_nodes, 90
- Installed_nodes, 90
- Monitor_check, 90
- Monitor_start, 90
- Monitor_stop, 90
- Num_resources_restart, 90
- Pkg_list, 90
- Postnet_stop, 90
- Prenet_stop, 90
- Resource_type, 90
- RT_basedir, 90
- RT_description, 90
- RT_version, 90
- Single_instance, 90
- START, 90
- STOP, 90
- UPDATE, 90
- VALIDATE, 90
- Vendor_ID, 90

resources, adding resources to a resource group, 40

- Retry_count, resource property, 97
- Retry_interval, resource property, 97
- RG_dependencies, resource group property, 110
- RG_description, resource group property, 110
- RG_mode, resource group property, 110
- RG_name, resource group property, 110
- RG_project_name, resource group property, 110
- RG_state, resource group property, 110

RGM

- See* resource group manager

RGOffload

- extension properties
 - continue_to_offload, 86
 - max_offload_retry, 86
 - rg_to_offload, 86
- fault monitor, 86

- RT_basedir, resource type property, 90
- RT_description, resource type property, 90
- RT_version, resource type property, 90

running
 See enabling

S

Scalable, resource property, 97
scrgadm command, 23
scsetup utility, 23
set up
 HAStorage
 existing resources, 76
 new resources, 74
 HAStoragePlus, 79
 RGOffload, 82
Single_instance, resource type property, 90
special requirements, identifying, 15
START, resource type property, 90
Status, resource property, 97
Status_message, resource property, 97
STOP, resource type property, 90
Sun Management Center GUI, 23
SunPlex Manager GUI, 22
switching the current primary, resource group, 55
synchronize startups, between resource groups and disk device group, 74

T

task map, data service resources, 28
Thorough_probe_interval, resource property, 97
tools
 scrgadm command, 23
 scsetup utility, 23
 Sun Management Center GUI, 23
 SunPlex Manager GUI, 22
Tunable, resource property attribute, 116
turning off
 See disabling
turning on
 See enabling
Type, resource property, 97
Type of property, resource property attribute, 116
Type_version, resource property, 97

U

Udp_affinity, resource property, 97
UPDATE, resource type property, 90
upgrading, resource type, 32
/usr/cluster/bin/scinstall -pv, 11

V

VALIDATE, resource type property, 90
values, resource group manager, 120
Vendor_ID, resource type property, 90
verifying, nsswitch.conf file contents, 16

W

Weak_affinity, resource property, 97