



Sun Cluster Data Service for MySQL Guide for Solaris OS

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

Part No: 819-1088-11
August 2005, Revision A

Copyright 2005 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 or registered trademarks 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.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

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 2005 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 États-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, de Sun Microsystems, Inc. aux États-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 États-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.



050503@11223



Contents

Preface	5
Installing and Configuring Sun Cluster HA for MySQL	11
Installing and Configuring Sun Cluster HA for MySQL	11
Sun Cluster HA for MySQL Overview	12
Planning the Sun Cluster HA for MySQL Installation and Configuration	13
Configuration Restrictions	13
Configuration Requirements	17
Installing and Configuring MySQL	21
▼ How to Install and Configure MySQL	21
Verifying the Installation and Configuration of MySQL	24
▼ How to Verify the Installation and Configuration of MySQL	24
Installing the Sun Cluster HA for MySQL Packages	25
▼ How to Install the Sun Cluster HA for MySQL Packages by Using the Web Start Program	25
▼ How to Install the Sun Cluster HA for MySQL Packages Using the <code>scsintall</code> Utility	26
Registering and Configuring Sun Cluster HA for MySQL	27
▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service	27
Verifying the Sun Cluster HA for MySQL Installation and Configuration	31
▼ How to Verify the Sun Cluster HA for MySQL Installation and Configuration	31
Understanding the Sun Cluster HA for MySQL Fault Monitor	32
Resource Properties	32
Probing Algorithm and Functionality	32
Debug Sun Cluster HA for MySQL	33

▼ How to turn on debug for Sun Cluster HA for MySQL	33
Upgrade to SC3.1 when Using Sun Cluster HA for MySQL	34
▼ Upgrade to Sun Cluster 3.1 when Using Sun Cluster HA for MySQL	34
Upgrade to MySQL 4.x.x from 3.23.54 when using Sun Cluster HA for MySQL	37
▼ Upgrade to MySQL 4.x.x from 3.23.54	38
Index	39

Preface

Sun Cluster Data Service for MySQL Guide for Solaris OS explains how to install and configure Sun™ Cluster HA for MySQL on both SPARC® and x86 based systems.

Note – In this document, the term “x86” refers to the Intel 32-bit family of microprocessor chips and compatible microprocessor chips made by AMD.

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 System and expertise with the volume manager software that is used with Sun Cluster.

Note – Sun Cluster software runs on two platforms, SPARC and x86. The information in this document pertains to both platforms unless otherwise specified in a special chapter, section, note, bulleted item, figure, table, or example.

UNIX Commands

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

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

Typographic Conventions

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

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name%</code> su Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . Perform a <i>patch analysis</i> . Do <i>not</i> save the file. [Note that some emphasized items appear bold online.]

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

Related Documentation

Information about related Sun Cluster topics is available in the documentation that is listed in the following table. All Sun Cluster documentation is available at <http://docs.sun.com>.

Topic	Documentation
Data service administration	<i>Sun Cluster Data Services Planning and Administration Guide for Solaris OS</i> Individual data service guides
Concepts	<i>Sun Cluster Concepts Guide for Solaris OS</i>
Overview	<i>Sun Cluster Overview for Solaris OS</i>
Software installation	<i>Sun Cluster Software Installation Guide for Solaris OS</i>
System administration	<i>Sun Cluster System Administration Guide for Solaris OS</i>
Hardware administration	<i>Sun Cluster 3.0-3.1 Hardware Administration Manual for Solaris OS</i> Individual hardware administration guides
Data service development	<i>Sun Cluster Data Services Developer's Guide for Solaris OS</i>
Error messages	<i>Sun Cluster Error Messages Guide for Solaris OS</i>
Command and function reference	<i>Sun Cluster Reference Manual for Solaris OS</i>

For a complete list of Sun Cluster documentation, see the release notes for your release of Sun Cluster at <http://docs.sun.com>.

Related Third-Party Web Site References

Third-party URLs that are referenced in this document provide additional related information.

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

Documentation, Support, and Training

Sun Function	URL	Description
Documentation	http://www.sun.com/documentation/	Download PDF and HTML documents, and order printed documents
Support and Training	http://www.sun.com/supporttraining/	Obtain technical support, download patches, and learn about Sun courses

Product Training

Sun Microsystems offers training in many Sun technologies through a variety of instructor-led courses and self-paced courses. For information about the training courses that Sun offers and to enroll in a class, visit Sun Microsystems Training at <http://training.sun.com/>.

Getting Help

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

- Your name and email address (if available)
- Your company name, address, and phone number
- The model and serial numbers of your systems
- The release number of the Solaris Operating System (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
SPARC: <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.

Installing and Configuring Sun Cluster HA for MySQL

This chapter explains how to install and configure Sun Cluster HA for MySQL.

This chapter contains the following sections.

- “Installing and Configuring Sun Cluster HA for MySQL” on page 11
- “Sun Cluster HA for MySQL Overview” on page 12
- “Planning the Sun Cluster HA for MySQL Installation and Configuration” on page 13
- “Installing and Configuring MySQL” on page 21
- “Verifying the Installation and Configuration of MySQL” on page 24
- “Installing the Sun Cluster HA for MySQL Packages” on page 25
- “Registering and Configuring Sun Cluster HA for MySQL” on page 27
- “Verifying the Sun Cluster HA for MySQL Installation and Configuration” on page 31
- “Understanding the Sun Cluster HA for MySQL Fault Monitor” on page 32
- “Debug Sun Cluster HA for MySQL” on page 33
- “Upgrade to SC3.1 when Using Sun Cluster HA for MySQL” on page 34
- “Upgrade to MySQL 4.x.x from 3.23.54 when using Sun Cluster HA for MySQL” on page 37

Installing and Configuring Sun Cluster HA for MySQL

Table 1 lists the tasks for installing and configuring Sun Cluster HA for MySQL. Perform these tasks in the order that they are listed.

TABLE 1 Task Map: Installing and Configuring Sun Cluster HA for MySQL

Task	For Instructions, Go To
1. Plan the installation.	“Sun Cluster HA for MySQL Overview” on page 12 “Planning the Sun Cluster HA for MySQL Installation and Configuration” on page 13
2. Install and configure MySQL.	“How to Install and Configure MySQL” on page 21
3. Verify installation and configuration.	“How to Verify the Installation and Configuration of MySQL” on page 24
4. Install Sun Cluster HA for MySQL Packages.	“Installing the Sun Cluster HA for MySQL Packages” on page 25
5. Register and Configure Sun Cluster HA for MySQL.	“How to Register and Configure Sun Cluster HA for MySQL as a Failover Service” on page 27
6. Verify Sun Cluster HA for MySQL Installation and Configuration.	“How to Verify the Sun Cluster HA for MySQL Installation and Configuration” on page 31
7. Understand Sun Cluster HA for MySQL fault monitor.	“Understanding the Sun Cluster HA for MySQL Fault Monitor” on page 32
8. Debug Sun Cluster HA for MySQL.	“Debug Sun Cluster HA for MySQL” on page 33
9. Upgrade to SC3.1 when using Sun Cluster HA for MySQL.	“Upgrade to SC3.1 when Using Sun Cluster HA for MySQL” on page 34
10. Upgrade to MySQL 4.0.15 from 3.23.54 when using Sun Cluster HA for MySQL.	“Upgrade to MySQL 4.x.x from 3.23.54 when using Sun Cluster HA for MySQL” on page 37

Sun Cluster HA for MySQL Overview

The MySQL software delivers a fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL is a trademark of MySQL AB™.

MySQL is freely available under the GNU General Public License, and you can download it from <http://www.mysql.com>.

The Sun Cluster HA for MySQL data service provides a mechanism for orderly startup and shutdown, fault monitoring and automatic failover of the MySQL service. The following MySQL components are protected by the Sun Cluster HA for MySQL data service.

TABLE 2 Protection of Components

Component	Protected by
MySQL server	Sun Cluster HA for MySQL

Planning the Sun Cluster HA for MySQL Installation and Configuration

This section contains the information you need to plan your Sun Cluster HA for MySQL installation and configuration.

Configuration Restrictions

This section provides a list of software and hardware configuration restrictions that apply to Sun Cluster HA for MySQL only.

For restrictions that apply to all data services, see the *Sun Cluster Release Notes*.



Caution – Your data service configuration might not be supported if you do not observe these restrictions.

- The Sun Cluster for MySQL can be configured only as a failover data service and not as a scalable data service.
- The MySQL configuration file (*my.cnf*) should be placed only in the MySQL Database directory. If *my.cnf* has to be placed in local file system, and then create a symlink from the MySQL Database directory. The configuration file *my.cnf* should not be placed in /etc (Global file) because it will override command line options
- **Supported configurations** – The following configurations are supported with the Sun Cluster HA for MySQL data service.
 - Single/Multiple MySQL instances in master configuration
 - Single/Multiple MySQL instances in slave configuration

Note – The master and slave instances shouldn't be on the same physical node.

- **Installing MySQL onto Cluster File Systems** —Regardless of which MySQL delivery method you have chosen, that is from <http://www.mysql.com> or from other source, the following restrictions apply.

- Each MySQL instance must have a unique *Database directory*. You can mount this *Database directory* as either a Failover File System or Global File System.

Note – It is best practice to mount Global File Systems with the */global* prefix and to mount Failover File Systems with the */local* prefix.

EXAMPLE 1 MySQL instances with Global File Systems

The following example shows MySQL installed onto a Global File System with two MySQL instances (*mysql-data-1* and *mysql-data-2*). The final output shows a subset of the */etc/vfstab* entries for MySQL deployed using Veritas Volume Manager.

```
# ls -l /usr/local
total 4
drwxrwxrwx  2 root    other      512 Oct  1 16:44 bin
lrwxrwxrwx  1 root    other      13 Oct 11 11:20 mysql -> /global/mysql
#
# ls -l /global/mysql
total 10432
drwxr-xr-x 13 mysql  mysql      512 Mar 16 00:03 .
drwxrwxrwx  7 root    other     2048 Apr 11 09:53 ..
-rw-r--r--  1 mysql  mysql     19106 Mar 15 23:29 COPYING
-rw-r--r--  1 mysql  mysql     28003 Mar 15 23:29 COPYING.LIB
-rw-r--r--  1 mysql  mysql    126466 Mar 15 16:47 ChangeLog
-rw-r--r--  1 mysql  mysql     6811 Mar 15 23:29 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql     1937 Mar 15 16:47 README
drwxr-xr-x  2 mysql  mysql     1536 Mar 16 00:03 bin
-rwxr-xr-x  1 mysql  mysql      773 Mar 16 00:03 configure
drwxr-x---  4 mysql  mysql      512 Mar 16 00:03 data
drwxr-xr-x  2 mysql  mysql     1024 Mar 16 00:03 include
drwxr-xr-x  2 mysql  mysql      512 Mar 16 00:03 lib
drwxr-xr-x  3 mysql  mysql      512 Mar 16 00:03 man
-rw-r--r--  1 mysql  mysql    2676944 Mar 15 23:23 manual.html
-rw-r--r--  1 mysql  mysql    2329252 Mar 15 23:23 manual.txt
-rw-r--r--  1 mysql  mysql     98233 Mar 15 23:23 manual_toc.html
drwxr-xr-x  6 mysql  mysql      512 Mar 16 00:03 mysql-test
drwxr-xr-x  2 mysql  mysql      512 Mar 16 00:03 scripts
drwxr-xr-x  3 mysql  mysql      512 Mar 16 00:03 share
drwxr-xr-x  7 mysql  mysql     1024 Mar 16 00:03 sql-bench
drwxr-xr-x  2 mysql  mysql      512 Mar 16 00:03 support-files
drwxr-xr-x  2 mysql  mysql      512 Mar 16 00:03 tests
# ls -l /global/mysql-data-1
Total 30
drwxrwxrwx  9 mysql  mysql      512 Apr 15 12:06 .
drwxrwxrwx 20 root    root     1024 Apr 10 12:41 ..
drwxr-xr-x  2 mysql  mysql      512 Apr 15 12:00 BDB
drwxrwxrwx  2 mysql  mysql      512 Apr 15 11:59 innodb
drwxrwxrwx  2 mysql  mysql     2048 Apr 15 14:47 logs
-rw-r--r--  1 mysql  mysql     1432 Apr 15 11:58 my.cnf
drwx----- 2 mysql  mysql      512 Apr 15 11:59 mysql
-rw-rw----  1 mysql  mysql      5 Apr 15 14:47 mysql.pid
```

EXAMPLE 1 MySQL instances with Global File Systems (Continued)

```
drwx----- 2 mysql mysql 512 Apr 15 14:53 sc3_test_database
drwx----- 2 mysql mysql 512 Apr 15 11:58 test
drwx----- 2 mysql mysql 512 Apr 15 12:00 testdb
#

# ls -l /global/mysql-data-2
total 32
drwxrwxrwx 9 mysql mysql 512 Apr 15 07:49 .
drwxrwxrwx 20 root root 1024 Apr 10 12:41 ..
drwxr-xr-x 2 mysql mysql 512 Apr 14 11:16 BDB
drwxr-xr-x 2 mysql mysql 512 Apr 14 11:14 innodb
drwxr-xr-x 2 mysql mysql 2560 Apr 15 10:15 logs
-rw-r--r-- 1 mysql mysql 1459 Apr 14 11:13 my.cnf
drwx----- 2 mysql mysql 512 Apr 14 11:14 mysql
-rw-rw---- 1 mysql mysql 5 Apr 15 10:10 mysqld.pid
drwx----- 2 mysql mysql 512 Apr 15 10:10 sc3_test_database
drwx----- 2 mysql mysql 512 Apr 14 11:14 test
drwx----- 2 mysql mysql 512 Apr 14 11:16 testdb

# more /etc/vfstab (Subset of the output)
/dev/vx/dsk/dg1/vol01 /dev/vx/rdisk/dg1/vol01 /global/mysql ufs 2
yes global,logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /global/mysql-data-1 ufs 2
yes global,logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /global/mysql-data-2 ufs 2
yes global,logging

#
```

Note – In the above example the *Database directory* for the MySQL instance 1 is /global/mysql-data-1, whereas the *Database directory* for the MySQL instance 2 is /global/mysql-data-2.

EXAMPLE 2 MySQL instances with Failover File Systems

The following example shows MySQL installed on Local File Systems and two MySQL instances (mysql-data-1 and mysql-data-2) on Failover File Systems. The final output shows a subset of the /etc/vfstab entries for MySQL deployed using Veritas Volume Manager.

```
# ls -l /usr/local/mysql
total 10432
drwxr-xr-x 13 mysql mysql 512 Mar 16 00:03 .
drwxrwxrwx 7 root other 2048 Apr 11 09:53 ..
-rw-r--r-- 1 mysql mysql 19106 Mar 15 23:29 COPYING
-rw-r--r-- 1 mysql mysql 28003 Mar 15 23:29 COPYING.LIB
-rw-r--r-- 1 mysql mysql 126466 Mar 15 16:47 ChangeLog
-rw-r--r-- 1 mysql mysql 6811 Mar 15 23:29 INSTALL-BINARY
```

EXAMPLE 2 MySQL instances with Failover File Systems (Continued)

```

-rw-r--r-- 1 mysql mysql 1937 Mar 15 16:47 README
drwxr-xr-x 2 mysql mysql 1536 Mar 16 00:03 bin
-rwxr-xr-x 1 mysql mysql 773 Mar 16 00:03 configure
drwxr-x--- 4 mysql mysql 512 Mar 16 00:03 data
drwxr-xr-x 2 mysql mysql 1024 Mar 16 00:03 include
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 lib
drwxr-xr-x 3 mysql mysql 512 Mar 16 00:03 man
-rw-r--r-- 1 mysql mysql 2676944 Mar 15 23:23 manual.html
-rw-r--r-- 1 mysql mysql 2329252 Mar 15 23:23 manual.txt
-rw-r--r-- 1 mysql mysql 98233 Mar 15 23:23 manual_toc.html
drwxr-xr-x 6 mysql mysql 512 Mar 16 00:03 mysql-test
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 scripts
drwxr-xr-x 3 mysql mysql 512 Mar 16 00:03 share
drwxr-xr-x 7 mysql mysql 1024 Mar 16 00:03 sql-bench
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 support-files
drwxr-xr-x 2 mysql mysql 512 Mar 16 00:03 tests
# ls -l /local/mysql-data-1
Total 30
drwxrwxrwx 9 mysql mysql 512 Apr 15 12:06 .
drwxrwxrwx 20 root root 1024 Apr 10 12:41 ..
drwxr-xr-x 2 mysql mysql 512 Apr 15 12:00 BDB
drwxrwxrwx 2 mysql mysql 512 Apr 15 11:59 innodb
drwxrwxrwx 2 mysql mysql 2048 Apr 15 14:47 logs
-rw-r--r-- 1 mysql mysql 1432 Apr 15 11:58 my.cnf
drwx----- 2 mysql mysql 512 Apr 15 11:59 mysql
-rw-rw---- 1 mysql mysql 5 Apr 15 14:47 mysqld.pid
drwx----- 2 mysql mysql 512 Apr 15 14:53 sc3_test_database
drwx----- 2 mysql mysql 512 Apr 15 11:58 test
drwx----- 2 mysql mysql 512 Apr 15 12:00 testdb
#

#ls -l /local/mysql-data-2
total 32
drwxrwxrwx 9 mysql mysql 512 Apr 15 07:49 .
drwxrwxrwx 20 root root 1024 Apr 10 12:41 ..
drwxr-xr-x 2 mysql mysql 512 Apr 14 11:16 BDB
drwxr-xr-x 2 mysql mysql 512 Apr 14 11:14 innodb
drwxr-xr-x 2 mysql mysql 2560 Apr 15 10:15 logs
-rw-r--r-- 1 mysql mysql 1459 Apr 14 11:13 my.cnf
drwx----- 2 mysql mysql 512 Apr 14 11:14 mysql
-rw-rw---- 1 mysql mysql 5 Apr 15 10:10 mysqld.pid
drwx----- 2 mysql mysql 512 Apr 15 10:10 sc3_test_database
drwx----- 2 mysql mysql 512 Apr 14 11:14 test
drwx----- 2 mysql mysql 512 Apr 14 11:16 testdb

# more /etc/vfstab (Subset of the output)
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /local/mysql-data-1 ufs 2
yes logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdisk/dg2/vol01 /local/mysql-data-2 ufs 2
yes logging
#

```

Note – In the above example the *Database directory* for the *MySQL* instance 1 is `/local/mysql-data-1`, whereas the *Database directory* for the *MySQL* instance 2 is `/local/mysql-data-2`.

Configuration Requirements

The requirements in this section apply to Sun Cluster HA for MySQL only. You must meet these requirements before you proceed with your Sun Cluster HA for MySQL installation and configuration.



Caution – Your data service configuration might not be supported if you do not adhere to these requirements.

Components and their Dependencies for Sun Cluster HA for MySQL

MySQL components and their dependencies —You can configure Sun Cluster HA for MySQL data service to protect a MySQL instance and its respective components. The components, and their dependencies between each other, are briefly described below.

Note – In the above example the *Database directory* for the *MySQL* instance 1 is `/local/mysql-data-1`, whereas the *Database directory* for the *MySQL* instance 2 is `/local/mysql-data-2`.

TABLE 3 Dependencies for Sun Cluster HA for MySQL

Component	Description
MySQL Server	→ <i>SUNW.LogicalHost</i> resource The <i>SUNW.LogicalHost</i> resource manages the MySQL ip address and ensures, that MySQL is not started until the ip address is up and running.
MySQL Server	→ <i>SUNW.HAStoragePlus</i> resource The <i>SUNW.HAStoragePlus</i> resource manages the MySQL File System Mount points and ensures, that MySQL is not started until these are mounted.

Registration and Configuration File for Sun Cluster HA for MySQL

The MySQL component has two configuration and registration files in `/opt/SUNWscmys/util`. These files allow you to register the MySQL component with Sun Cluster and prepare a MySQL instance to be registered.

Within these files, the appropriate dependencies have been applied.

EXAMPLE 3 MySQL configuration and registration file for Sun Cluster

```
# cd /opt/SUNWscmys
#
# ls -l util
-rwxr-xr-x  1 root    other      1208 Mar 25 13:49 ha_mysql_config
-rwxr-xr-x  1 root    other       820 Mar 25 13:55 ha_mysql_register
-rwxr-xr-x  1 root    other     1105 Apr 16 07:48 mysql_config
-rwxr-xr-x  1 root    other     2993 Apr 11 10:57 mysql_register
#
# more util/*config
:::::::::::::
util/mysql_config
:::::::::::::
#
# Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#

# Where is mysql installed (BASEDIR)
MYSQL_BASE=

# Mysql admin-user for localhost
MYSQL_USER=

# Password for mysql admin user
MYSQL_PASSWD=

# Configured logicalhost
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user
FMPASS=

# Socket name for mysqld ( Should be /tmp/<logical host>.sock )
MYSQL_SOCKET=/tmp/<logical host>.sock

# FOR SC3.1 ONLY, Specify the physical hostname for the
# physical NIC that this logicalhostname belongs to for every node in the
```

EXAMPLE 3 MySQL configuration and registration file for Sun Cluster (Continued)

```
# cluster this Resourcegroup can located on.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme1 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
MYSQL_NIC_HOSTNAME=""
::::::::::::
util/ha_mysql_config
::::::::::::
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
# RS - name of the resource for the application
# RG - name of the resource group containing RS
# BASEDIR - name of the Mysql bin directory
# DATADIR - name of the Mysql Data directory
# MYSQLUSER - name of the user Mysql should be started of
# LH - name of the LogicalHostname SC resource
# MYSQLHOST - name of the host in /etc/hosts
# FMUSER - name of the Mysql fault monitor user
# FMPASS - name of the Mysql fault monitor user password
# LOGDIR - name of the directory mysqld should store it's logfile.
# CHECK - should HA-MySQL check MyISAM index files before start YES/NO.
# HAS_RS - name of the mysql HAStoragePlus SC resource
#
# The following examples illustrate sample parameters
# for Mysql
#
# BASEDIR=/usr/local/mysql
# DATADIR=/global/mysqldata
# MYSQLUSER=mysql
# LH=mysql1h
# MYSQLHOST=mysqlipname
# FMUSER=fmuser
# FMPASS=fmuser
# LOGDIR=/global/mysqldata/logs
# CHECK=YES
#
RS=
RG=
BASEDIR=
DATADIR=
MYSQLUSER=
LH=
MYSQLHOST=
FMUSER=
```

EXAMPLE 3 MySQL configuration and registration file for Sun Cluster (Continued)

```
FMPASS=  
LOGDIR=  
CHECK=  
HAS_RS=  
#
```

Explanation of the my.cnf File

my.cnf file — The Sun Cluster HA for MySQL data service provides two sample my.cnf files, one sample file for a master configuration and one for a slave configuration. However, ensure that at least the following parameters are set.

Note – The my.cnf file is an important file within MySQL. Refer to the MySQL Documentation for complete configuration information on the parameters that follow.

MySQL my.cnf file in [mysqld] section in a master configuration

- **bind address** must be set to the defined logical hosts ip name.
- **socket** is defined as /tmp/<logical hosts ip name>.sock.
- **binlog-ignore-db** contains sc3_test_database, if log-bin option is being used.

MySQL my.cnf file in [mysqld] section in a slave configuration

- **bind address** must be set to the defined logical hosts ip name.
- **socket** is defined as /tmp/<logical hosts ip name>.sock.
- **binlog-ignore-db** contains sc3_test_database if log-bin option is being used.
- **master—host** is the hostname where the master instance resides.
- **master—user** is the username the slave will use for identification to the master.
- **master—password** is the password the slave will use for identification to the master.
- **master-info-file** is the location of the file that remembers where MySQL left off on the master during the replication process. This file must be placed on a GFS/FFS.

Installing and Configuring MySQL

This section contains the procedures you need to install and configure MySQL.

References will be made to certain directories for MySQL. The following list shows common pathnames for these references. Refer to “[Configuration Restrictions](#)” on page 13 where these examples .

- MySQL installed from <http://www.mysql.com> on a Global File System, with a MySQL instance on a Global File System ([Example 1](#))
 - *MySQL Basedirectory*— /global/mysql
 - *MySQL Database directory* — /global/mysqldata
- MySQL installed on a Local File System, with mysql instances on a Failover File System ([Example 2](#))
 - *MySQL Basedirectory*— /usr/local/mysql
 - *MySQL Database directory* — /local/mysqldata

▼ How to Install and Configure MySQL

- Steps**
1. **Determine how MySQL will be deployed in Sun Cluster –**
 - Determine how many MySQL instances will be deployed.
 - Determine which Cluster File System will be used by each MySQL instance.
 2. **Mount the MySQL Cluster File Systems –**

Note – If Failover File Systems will be used by the MySQL instance, you must mount these manually.

3. **Install MySQL onto all nodes within Sun Cluster**

It is recommended that MySQL be installed onto a Global File System. For a discussion of the advantages and disadvantages of installing the software on local versus cluster files systems, see “Determining the Location of the Application Binaries” in the *Sun Cluster Data Services Installation and Configuration Guide*.

Download MySQL from <http://www.mysql.com> — If you intend to use local disks for the MySQL software, you will need to repeat this step on all nodes within Sun Cluster.

4. **Create a mysql-user and mysql-group for MySQL on all nodes in the cluster that will run MySQL**

Create an entry in `/etc/group` on all nodes with Sun Cluster.

■

```
# groupadd -g 1000 mysql
```

Create an entry in `/etc/passwd` on all nodes within Sun Cluster. This user should have a locked password.

```
# useradd -u 1000 -g 1000 -d /global/mysql -s /bin/sh mysql
```

5. Change owner and group for MySQL binaries

If MySQL binaries are on all nodes then repeat this step on every node.

```
# chown -R mysql:mysql /global/mysql
```

6. Create your MySQL Database directory for your MySQL Instance(s).

```
# mkdir <MySQL Database directory>
```

```
#
```

Note – Refer back to “[Configuration Restrictions](#)” on page 13 for a description of the `<MySQL Database directory>` and to “[Installing and Configuring MySQL](#)” on page 21 for a list of common pathnames.

Shows one MySQL instance. MySQL has been installed from <http://www.mysql.com> in `/global/mysql` which is mounted as a Global File System. The *MySQL Database Directory* for the MySQL instance is `/global/mysql-data`.

```
# cd /global/mysql
#
# ls -l
-rw-r--r-- 1 mysql mysql 19106 Dec 10 14:52 COPYING
-rw-r--r-- 1 mysql mysql 28003 Dec 10 14:52 COPYING.LIB
-rw-r--r-- 1 mysql mysql 44577 Dec 5 10:37 ChangeLog
-rw-r--r-- 1 mysql mysql 6811 Dec 10 14:53 INSTALL-BINARY
-rw-r--r-- 1 mysql mysql 1976 Dec 5 10:37 README
drwxr-xr-x 2 mysql mysql 1024 Dec 13 18:05 bin
-rwxr-xr-x 1 mysql mysql 773 Dec 10 15:34 configure
drwxr-x-- 3 mysql mysql 512 Apr 3 12:23 data
drwxr-xr-x 2 mysql mysql 1024 Dec 10 15:35 include
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 lib
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 man
-rw-r--r-- 1 mysql mysql 2582089 Dec 10 14:47 manual.html
-rw-r--r-- 1 mysql mysql 2239278 Dec 10 14:47 manual.txt
-rw-r--r-- 1 mysql mysql 94600 Dec 10 14:47 manual_toc.html
drwxr-xr-x 6 mysql mysql 512 Dec 10 15:35 mysql-test
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 scripts
drwxr-xr-x 3 mysql mysql 512 Dec 10 15:35 share
drwxr-xr-x 7 mysql mysql 1024 Dec 10 15:35 sql-bench
drwxr-xr-x 2 mysql mysql 512 Dec 10 15:35 support-files
```

```
drwxr-xr-x  2 mysql  mysql          512 Dec 10 15:35 tests
#
```

7. **Create the MySQL my.cnf file according to your requirements — The Sun Cluster HA for MySQL data service provides two sample my.cnf file for MySQL: one sample configuration file is for a master configuration and one sample file is for a slave configuration.**

Note – If the Sun Cluster HA for MySQL package (SUNWscmys) was not installed during your initial Sun Cluster installation, proceed to [“Installing the Sun Cluster HA for MySQL Packages”](#) on page 25 to install it on SC3.1 and SC3.0U3. Return here to continue the Installation and Configuration of MySQL.

The contents of

`/opt/SUNWscmys/etc/my.cnf_sample_[master|slave]` provide a sample MySQL configuration file that you can use to create your MySQL instance `<MySQL Databasedirectory>/my.cnf`. You must still edit that file to reflect your configuration values.

```
# cp /opt/SUNWscmys/etc/my.cnf_sample_master \
  <MySQL Databasedirectory>/my.cnf
```

8. **Bootstrap MySQL instance — This will create the privilege tables `db`, `host`, `user`, `tables_priv` and `columns_priv` in the `mysql` database, as well as the `func` table.**

```
# cd <MySQL Basedirectory>
# ./scripts/mysql_install_db \
  --datadir=<MySQL Database directory>
```

9. **Create a logfile directory in `<MySQL Database Directory>`**

```
# mkdir <MySQL Database Directory>/logs
```

10. **Change owner and group for `<MySQL Database Directory>`**

```
# chown -R mysql:mysql <MySQL Database Directory>
```

11. **Change file permission for `<MySQL Database Directory>/my.cnf`**

```
# chmod 644 <MySQL Database Directory>/my.cnf
```

Verifying the Installation and Configuration of MySQL

This section contains the procedure you need to verify the installation and configuration.

▼ How to Verify the Installation and Configuration of MySQL

This procedure does not verify that your application is highly available because you have not yet installed your data service.

Note – Before verifying the installation and configuration of MySQL, ensure that the Logical Hostname for the mysql is available. You will need to complete steps 2 through 7 in “How to Register and Configure Sun Cluster HA for MySQL as a Failover Service” on page 27.

Steps 1. Start the MySQL Server for this instance.

```
#cd <MySQL Basedirectory>
# ./bin/mysqld --defaults-file=<MySQL Databasedirectry>/my.cnf \
--basedir=<MySQL Basedirectory>\
--datadir=<MySQL Databasedirectory>\
--user=mysql --pid-file=<MySQL Databasedirectory>/mysqld.pid &
```

2. Connect to the MySQL instance.

```
# <MySQL Basedirectory>/bin/mysql -h <Logical host> -uroot
```

3. Stop the MySQL server instance.

```
# kill -TERM `cat <MySQL Databasedirectory>/mysqld.pid`
```

Installing the Sun Cluster HA for MySQL Packages

If you did not install the Sun Cluster HA for MySQL packages during your initial Sun Cluster installation, perform this procedure to install the packages. Perform this procedure on each cluster node where you are installing the Sun Cluster HA for MySQL packages. To complete this procedure, you need the Sun Cluster Agents CD-ROM.

If you are installing more than one data service simultaneously, perform the procedure in “Installing the Software” in *Sun Cluster Software Installation Guide for Solaris OS*.

Install the Sun Cluster HA for MySQL packages using one of the following installation tools:

- Web Start program
- `scinstall` utility

Note – The Web Start program is not available in releases earlier than Sun Cluster 3.1 Data Services 10/03.

If you are using Solaris 10, install these packages *only* in the global zone. To ensure that these packages are not propagated to any local zones that are created after you install the packages, use the `scinstall` utility to install these packages. Do *not* use the Web Start program.

▼ How to Install the Sun Cluster HA for MySQL Packages by Using the Web Start Program

You can run the Web Start program with a command-line interface (CLI) or with a graphical user interface (GUI). The content and sequence of instructions in the CLI and the GUI are similar. For more information about the Web Start program, see the `installer(1M)` man page.

- Steps**
1. On the cluster node where you are installing the Sun Cluster HA for MySQL packages, become superuser.
 2. (Optional) If you intend to run the Web Start program with a GUI, ensure that your `DISPLAY` environment variable is set.
 3. Insert the Sun Cluster Agents CD-ROM into the CD-ROM drive.

If the Volume Management daemon `vold(1M)` is running and configured to manage CD-ROM devices, it automatically mounts the CD-ROM on the `/cdrom/cdrom0` directory.

4. Change to the Sun Cluster HA for MySQL component directory of the CD-ROM.

The Web Start program for the Sun Cluster HA for MySQL data service resides in this directory.

```
# cd /cdrom/cdrom0/components/SunCluster_HA_MySQL_3.1
```

5. Start the Web Start program.

```
# ./installer
```

6. When you are prompted, select the type of installation.

- To install only the C locale, select Typical.
- To install other locales, select Custom.

7. Follow the instructions on the screen to install the Sun Cluster HA for MySQL packages on the node.

After the installation is finished, the Web Start program provides an installation summary. This summary enables you to view logs that the Web Start program created during the installation. These logs are located in the `/var/sadm/install/logs` directory.

8. Exit the Web Start program.

9. Remove the Sun Cluster Agents CD-ROM from the CD-ROM drive.

- a. To ensure that the CD-ROM is not being used, change to a directory that does *not* reside on the CD-ROM.
- b. Eject the CD-ROM.

```
# eject cdrom
```

▼ How to Install the Sun Cluster HA for MySQL Packages Using the `scsinstall` Utility

You need the Sun Cluster Agents CD-ROM to perform this procedure. This procedure assumes that you did not install the data service packages during your initial Sun Cluster installation.

If you installed the Sun Cluster HA for MySQL packages as part of your initial Sun Cluster installation, proceed to [“Registering and Configuring Sun Cluster HA for MySQL” on page 27](#).

Perform this procedure on all nodes that can run Sun Cluster HA for MySQL data service.

- Steps**
1. **Load the Sun Cluster Agents CD-ROM into the CD-ROM drive.**
 2. **Run the `scinstall` utility with no options.**
This step starts the `scinstall` utility in interactive mode.
 3. **Choose the menu option, Add Support for New Data Service to This Cluster Node.**
The `scinstall` utility prompts you for additional information.
 4. **Provide the path to the Sun Cluster Agents CD-ROM.**
The utility refers to the CD as the “data services cd.”
 5. **Specify the data service to install.**
The `scinstall` utility lists the data service that you selected and asks you to confirm your choice.
 6. **Exit the `scinstall` utility.**
 7. **Unload the CD from the drive.**

Registering and Configuring Sun Cluster HA for MySQL

This section contains the procedures you need to configure Sun Cluster HA for MySQL.

▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service

This procedure assumes that you installed the data service packages during your initial Sun Cluster installation.

If you did not install the Sun Cluster HA for MySQL packages as part of your initial Sun Cluster installation, go to [“How to Install the Sun Cluster HA for MySQL Packages Using the `scinstall` Utility”](#) on page 26.

- Steps**
1. **Become superuser on one of the nodes in the cluster that will host MySQL.**
 2. **Register the `SUNW.gdsresource` type.**

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

3. Register the *SUNW.HAStoragePlus* resource type.

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

4. Create a failover resource group.

```
# scrgadm -a -g MySQL-failover-resource-group
```

5. Create a resource for the MySQL Disk Storage.

```
# scrgadm -a -j MySQL-has-resource \  
-g MySQL-failover-resource-group \  
-t SUNW.HAStoragePlus \  
-x FilesystemMountPoints=MySQL-instance-mount-points
```

6. Create a resource for the MySQL Logical Hostname.

```
# scrgadm -a -L -j MySQL-lh-resource \  
-g MySQL-failover-resource-group \  
-l MySQL-logical-hostname
```

7. Enable the failover resource group that now includes the MySQL Disk Storage and Logical Hostname resources.

```
# scswitch -z -g MySQL-failover-resource-group
```

Note – Make sure that the resource group is enabled on the same node as what was used when *mysql_install_db* was executed.

8. Start the MySQL Server instance manually.

```
#cd <MySQL Basedirectory>  
  
# ./bin/mysqld --defaults-file=<MySQL Databasedirectry>/my.cnf \  
--basedir=<MySQL. Basedirectory>\ \  
--datadir=<MySQL Databasedirectory>\ \  
--user=mysql\  
--pid-file=<MySQL Databasedirectory>/mysqld.pid &
```

9. Configure the admin password for the adminuser.

```
# <MySQL Databasedirectry>/bin/mysqladmin -s /tmp/<Logical host>.sock password 'admin password'
```

10. Create faultmonitor-user and a test-database for the MySQL instance.

```
# cd /opt/SUNWscmys/util  
Edit the mysql_config file and follow the comments within that file:  
  
# Where is mysql installed (BASEDIR)  
MYSQL_BASE=  
  
# Mysql admin-user for localhost (Should be root)  
MYSQL_USER=
```

```

# Password for mysql admin user
MYSQL_PASSWD=

# Configured logicalhost
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user
FMPASS=

# Socket name for mysqld ( Should be /tmp/<Logical host>.sock )
MYSQL_SOCKET=

# FOR SC3.1 ONLY, Specify the physical hostname for the
# physical NIC that this logicalhostname belongs to for every node in the
# cluster this Resourcegroup can be located on.
# IE: The logicalhost lhl belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme1 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
MYSQL_NIC_HOSTNAME=""

```

Note – The following is an example for MySQL instance on SC3.1.

```

MYSQL_BASE=/global/mysql
MYSQL_USER=root
MYSQL_PASSWD=root
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCKET=/tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME="clusterix1 clusterix2"

```

Note – If SC3.0U3 is being used, omit the MYSQL_NIC_HOSTNAME parameter.

After editing `mysql_config` you must run the `mysql_register` script.

```
# ./mysql_register
```

11. Stop the MySQL Server instance manually.

```
# kill -TERM `cat <MySQL Databasedirectory>/mysqld.pid`
```

12. Create and register MySQL as a failover data service.

```
# cd /opt/SUNWscmys/util
```

Edit the `ha_mysql_config` file and follow the comments within that file, i.e.

```
These parameters can be customized in (key=value) form
#
#     RS - name of the resource for the application
#     RG - name of the resource group containing RS
#     BASEDIR - name of the Mysql bin directory
#     DATADIR - name of the Mysql Data directory
#     MYSQLUSER - name of the user Mysql should be started with
#     LH - name of the LogicalHostname SC resource
#     MYSQLHOST - name of the host in /etc/hosts
#     FMUSER - name of the Mysql fault monitor user
#     FMPASS - name of the Mysql fault monitor user password
#     LOGDIR - name of the directory mysqld should store it's logfile.
#     CHECK - should HA-MySQL check MyISAM index files before start YES/NO
#     HAS_RS - name of the MySQL HAStoragePlus SC resource
#
```

The following is an example for MySQL instance.

```
RS=mysql-res
RG=mysql-rg
BASEDIR=/global/mysql
DATADIR=/global/mysql-data
MYSQLUSER=mysql
LH=hahostix1
MYSQLHOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
LOGDIR=/global/mysql-data/logs
CHECK=YES
HAS_RS=mysql-has-res
```

13. Enable each MySQL resource.

Repeat this step for each MySQL instance, if multiple instances were created.

```
# scstat -g
# scswitch -e -j MySQL-resource
```

14. Add adminuser for accessing locally a MySQL instance with a MySQL Logicalhost ip name.

Note – If you want to access the MySQL instance only through the socket (localhost), omit this step.

When bootstrapping MySQL the command `mysql_install_db` will create two adminusers, one belonging to localhost and one belonging to the node on which `mysql_install_db` was executed.

If SC3.0U3 is being used you should delete the root user belonging to the physical node and add a adminuser belonging to the logical host

The following is an example for MySQL instance on SC3.0U3.

```
# mysql -S /tmp/hahostix1.sock -uroot
mysql> use mysql;
mysql> DELETE FROM user where User='root' AND Host='clusterix1';
mysql> GRANT ALL ON *.* TO 'root'@'hahostix1' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

If you are using SC3.1, add a adminuser for every physical-node in the cluster that will run this MySQL instance.

Note – If the nodename and the hostname for the physical interface are different, use the hostname for the physical interface.

The following is an example for MySQL instance on SC3.1.

```
# mysql -S /tmp/hahostix1.sock -uroot
mysql> use mysql;
mysql> mysql> GRANT ALL ON *.* TO 'root'@'clusterix2' IDENTIFIED BY 'rootpasswd';
mysql> mysql> GRANT ALL ON *.* TO 'root'@'clusterix1' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

Note – You have to manually add *Grant_priv* to the admin users. See MySQL Administration documentation.

Verifying the Sun Cluster HA for MySQL Installation and Configuration

This section contains the procedure you need to verify that you installed and configured your data service correctly.

▼ How to Verify the Sun Cluster HA for MySQL Installation and Configuration

- Steps**
1. Become superuser on one of the nodes in the cluster that will host MySQL.
 2. Ensure all the MySQL resources are online with `scstat`.

```
# scstat
```

For each MySQL resource that is not online, use the `scswitch` command as follows.

```
# scswitch -e -j MySQL-resource
```

3. Run the `scswitch` command to switch the MySQL resource group to another cluster node, such as `node2`.

```
# scswitch -z -g MySQL-failover-resource-group -h node2
```

Understanding the Sun Cluster HA for MySQL Fault Monitor

This section describes the Sun Cluster HA for MySQL fault monitor's probing algorithm or functionality, states the conditions, messages, and recovery actions associated with unsuccessful probing.

For conceptual information on fault monitors, see the *Sun Cluster Concepts Guide*.

Resource Properties

The Sun Cluster HA for MySQL fault monitor uses the same resource properties as resource type `SUNW.gds`. Refer to the `SUNW.gds(5)` man page for a complete list of resource properties used.

Probing Algorithm and Functionality

- **MySQL probing steps.**
 - Sleeps for `Thorough_probe_interval`.
 - Tries to connect to the MySQL instance, with `mysqladmin` command with argument `ping`, with the defined faultmonitor user fault `<fmuser>`. If this fails, then the probe will restart the MySQL resource.
 - Every 300 seconds the probe will also check the following:
 - If the MySQL instance is a slave configuration, the probe will check whether the MySQL instance is connected to its master. If the Slave is not connected the probe will write an error message to `syslog`.
 - Verify that the probe can list all databases and tables, not the contents. If the probe receive any errors the probe will write an error message to `syslog`.

Conduct an function test on the defined test-database, Create Table, Insert into Table, Update Table, Delete from Table and Drop Table. If any of those operations fails then the probe will restart the MySQL resource.

- If all MySQL processes have died, pmf will interrupt the probe to immediately restart the mysql resource.
- If the mysql resource is repeatedly restarted and subsequently exhausts the `Retry_count` within the `Retry_interval` then a failover is initiated for the Resource Group onto another node if `Failover_enabled` is set to `TRUE`.

Debug Sun Cluster HA for MySQL

▼ How to turn on debug for Sun Cluster HA for MySQL

Use the information in this section to understand how to turn on debug for Sun Cluster HA for MySQL.

Sun Cluster HA for MySQL can be used by multiple MySQL instances. However, it is possible to turn on debug for all MySQL instances or for a particular MySQL instance.

The MySQL component has a `DEBUG` file under `/opt/SUNWscmys/etc`.

This file allow you to turn on debug for all MySQL instances or for a specific MySQL instance on a particular node with Sun Cluster. If you require debug to be turned on for Sun Cluster HA for MySQL across the whole Sun Cluster, you will need to repeat this step on all nodes within Sun Cluster.

Steps 1. Edit `/etc/syslog.conf`

Edit `/etc/syslog.conf` and change `daemon.notice` to `daemon.debug`

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit      /var/adm/messages
*.alert;kern.err;daemon.err                   operator
#
```

Change the `daemon.notice` to `daemon.debug` and restart `syslogd`. The output below, from the command `grep daemon /etc/syslog.conf`, shows that `daemon.debug` has been set.

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.debug;mail.crit      /var/adm/messages
*.alert;kern.err;daemon.err                   operator
```

```
#  
# pkill -1 syslogd  
#
```

2. Edit `/opt/SUNWscmys/etc/config` and change `DEBUG=` to `DEBUG=ALL` or `DEBUG=resource`

```
# cat /opt/SUNWscmys/etc/config  
#  
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.  
# Use is subject to license terms.  
#  
# Usage:  
#     DEBUG=<RESOURCE_NAME> or ALL  
#  
DEBUG=ALL  
#
```

Note – To turn off debug, reverse the steps above.

Upgrade to SC3.1 when Using Sun Cluster HA for MySQL

Use the information in this section to understand how to upgrade to SC3.1 when using Sun Cluster HA for MySQL.

Note – This procedure will not describe how to upgrade to SC3.1. It includes only the steps to upgrade Sun Cluster HA for MySQL to SC3.1.

▼ Upgrade to Sun Cluster 3.1 when Using Sun Cluster HA for MySQL

This procedure shows the steps how to upgrade Sun Cluster HA for MySQL to SC3.1.

- Steps**
1. Shutdown Sun Cluster HA for MySQL resource with `scswitch -n -j MySQL-resource`.

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

2. Upgrade the nodes to SC3.1 according to Sun Cluster documentation.

3. Start the MySQL Server manually on SC3.1.

```
#cd <MySQL Basedirectory>

# ./bin/mysqld --defaults-file=<MySQL Databasedirectory>/my.cnf \
--basedir=<MySQL Basedirectory>\
--datadir=<MySQL Databasedirectory>\
--user=mysql\
--pid-file=<MySQL Databasedirectory>/mysqld.pid &
```

4. Access the MySQL instance from local node with the socket option.

```
#<MySQL Basedirectory>
/bin/mysql -s <MySQL Socket> -uroot \
-p<Adminpassword>\
```

The following is an example for a MySQL instance.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql>
```

5. Drop the Sun Cluster HA for MySQL test database `sc3_test_database`.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql> DROP DATABASE sc3_test_database;
Query OK, 0 rows affected (0.03 sec)
```

6. Delete all entries in db-table in mysql-database that contain `User='<MySQL Faultmonitor user>'`.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql> use mysql;
Database changed
mysql> DELETE FROM db WHERE User='fmuser';
Query OK, 1 row affected (0.03 sec)
```

7. Delete all entries in user-table in mysql-database that contain `User='<MySQL Faultmonitor user>'`.

```
# mysql -s /tmp/hahostix1.sock -uroot -proot
mysql> use mysql;
Database changed
mysql> DELETE FROM user WHERE User='fmuser';
Query OK, 1 row affected (0.03 sec)
```

8. Add faultmonitor-user and a test-database to MySQL.

```
# cd /opt/SUNWscmys/util
```

Edit the `mysql_config` file and follow the comments within that file:

```
# Where is mysql installed (BASEDIR)
MYSQL_BASE=

# Mysql admin-user for localhost (Should be root)
MYSQL_USER=

# Password for mysql admin user
```

```

MYSQL_PASSWD=

# Configured logicalhost
MYSQL_HOST=

# Specify a username for a faultmonitor user
FMUSER=

# Pick a password for that faultmonitor user
FMPASS=

# Socket name for mysqld ( Should be /tmp/<Logical host>.sock )
MYSQL_SOCKET=

# FOR SC3.1 ONLY, Specify the physical hostname for the
# physical NIC that this logicalhostname belongs to for every node in the
# cluster this Resourcegroup can located on.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme1 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
MYSQL_NIC_HOSTNAME=""

```

Note – The following is an example for MySQL instance on SC3.1.

```

MYSQL_BASE=/global/mysql
MYSQL_USER=root
MYSQL_PASSWD=root
MYSQL_HOST=hahostix1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCKET=/tmp/hahostix1.sock
MYSQL_NIC_HOSTNAME="clusterix1 clusterix2"

```

9. After editing `mysql_config`, run the `mysql_register` script.

```
# ./mysql_register
```

10. Stop the MySQL Server manually.

```
# kill -TERM `cat <MySQL Databasedirectry>/mysqld.pid`
```

11. Startup Sun Cluster HA for MySQL resource with `scswitch -e -j MySQL-resource`

```
# scswitch -e -j MySQL-resource
```

12. Change source addresses for adminuser for accessing locally a MySQL instance with a MySQL Logicalhost.

Note – If you want to access the MySQL instance only through the socket (localhost), omit this step.

When SC3.0U3 has been used, delete the root user belonging to the logicalhost and add an adminuser belonging to the physicalhost.

Note – If the nodename and the hostname for the physical interface are different, use the hostname for the physical interface.

The following is an example for MySQL instance on SC3.1.

```
# mysql -S /tmp/hahostix1.sock -uroot
mysql> use mysql;
mysql> DELETE FROM user where User='root' AND Host='hahostix1';
mysql> GRANT ALL ON *.* TO 'root'@'clusterix1' IDENTIFIED BY 'rootpasswd';
mysql> GRANT ALL ON *.* TO 'root'@'clusterix2' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

Note – You have to manually add the Grant_priv to the admin users, See MySQL Administration documentation.

Upgrade to MySQL 4.x.x from 3.23.54 when using Sun Cluster HA for MySQL

Use the information in this section to understand how to upgrade to 4.0.12 when using Sun Cluster HA for MySQL.

Note – The procedure will not describe how to upgrade to 4.x.x. It includes only the steps to upgrade Sun Cluster HA for MySQL to 4.x.x. These steps assume that the new MySQL binaries is installed in the same place. If the new binaries is installed in a new directory, you must reregister the MySQL resource with new *MySQL Basedir*.

▼ Upgrade to MySQL 4.x.x from 3.23.54

Procedure to upgrade to MySQL 4.x.x from 3.23.54

- Steps**
1. **Shutdown the Sun Cluster HA for MySQL with `scswitch -n -j MySQL- resource`**
`# scswitch -n -j MySQL- resource`
 2. **Install the new MySQL binaries, follow Step 3 and 4 in Section How to Install and Configure MySQL.**
 3. **Startup Sun Cluster HA for MySQL with `scswitch -e -j MySQL- resource`.**
`# scswitch -e -j MySQL- resource`
 4. **Shutdown Sun Cluster HA for MySQL faultmonitor with `scswitch -n -M -j MySQL- resource`.**
`# scswitch -n -M -j MySQL- resource`
 5. **Follow the MySQL Documentation to upgrade MySQL Database.**
 6. **Startup Sun Cluster HA for MySQL faultmonitor with `scswitch -e -M -j MySQL- resource`.**
`# scswitch -e -M -j MySQL- resource`

Index

A

Adminuser, 30

B

Bootstrap, 23

C

C locale, 26

Command

 chmod, 23

 chown, 22, 23

 groupadd, 22

 scrgadm, 27, 28

 scstat, 31

 scswitch, 28, 32

 useradd, 22

commands, node information, 9

Configuration

 MySQL, 21-23

 Registering Sun Cluster HA for
 MySql, 27-31

 Requirements, 17-20

 Restrictions, 13-17

 Supported configuration, 13

 Verify, 24

D

directories, /var/sadm/install/logs, 26

F

Fault Monitor

 Probing Algorithm and Functionality, 32-33

 Resource Properties, 32

 Understanding, 32-33

files, installation logs, 26

G

global zone, 25

I

Install, 13, 21-23

installing

 Sun Cluster HA for MySQL

 by using Web Start program, 25-26

 log files created, 26

L

local zones, 25

locales, 26

log files, installation, 26

M

MySQL, 12

P

prtconf -v command, 9
prtdiag -v command, 9
psrinfo -v command, 9

R

Resource Types

SUNW.gds, 27

SUNW.HAStoragePlus, 28

restrictions, zones, 25

S

scinstall, 27

scinstall -pv command, 9

showrev -p command, 9

Sun Cluster HA for MySQL

installing

by using Web Start program, 25-26

T

training, 8

U

Upgrade

Upgrade to MySQL 4.0.12 from
3.23.54, 37-38

Upgrade to SC3.1 when using Sun Cluster
HA for MySQL, 34-37

V

/var/sadm/install/logs directory, 26

W

Web Start program, 25-26

Z

zones, 25