



# Sun Cluster 3.1 Data Service for MySQL Guide

---

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

Part No: 817-3242-10  
October 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 REPOUDRE 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.



030731@5943



# Contents

---

<b>Installing and Configuring Sun Cluster HA for MySQL</b>	<b>5</b>
Installing and Configuring Sun Cluster HA for MySQL	5
Sun Cluster HA for MySQL Overview	6
Planning the Sun Cluster HA for MySQL Installation and Configuration	6
Configuration Restrictions	7
Configuration Requirements	11
Installing and Configuring MySQL	14
How to Install and Configure MySQL	14
Verifying the Installation and Configuration of MySQL	17
▼ How to Verify the Installation and Configuration of MySQL	17
Installing the Sun Cluster HA for MySQL Packages	18
How to Install the Sun Cluster HA for MySQL Packages by Using the Web Start Program	18
▼ How to Install the Sun Cluster HA for MySQL Packages by Using the <code>scsintall</code> Utility	19
Registering and Configuring Sun Cluster HA for MySQL	20
▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service	20
Verifying the Sun Cluster HA for MySQL Installation and Configuration	25
▼ How to Verify the Sun Cluster HA for MySQL Installation and Configuration	25
Understanding the Sun Cluster HA for MySQL Fault Monitor	25
Resource Properties	26
Probing Algorithm and Functionality	26
Debug Sun Cluster HA for MySQL	26
▼ How to turn debug on for Sun Cluster HA for MySQL	26
Upgrade to SC3.1 when using Sun Cluster HA for MySQL	28

Upgrade to MySQL 4.0.12 from 3.23.54 when using Sun Cluster HA for MySQL 31

**Index** 33

# Installing and Configuring Sun Cluster HA for MySQL

---

---

## Installing and Configuring Sun Cluster HA for MySQL

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

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

<b>Task</b>	<b>For Instructions, Go To</b>
Plan the installation	“Sun Cluster HA for MySQL Overview” on page 6 “Planning the Sun Cluster HA for MySQL Installation and Configuration” on page 6
Install and configure MySQL	“How to Install and Configure MySQL” on page 14
Verify installation and configuration	“How to Verify the Installation and Configuration of MySQL” on page 17
Install Sun Cluster HA for MySQL Packages	“How to Install the Sun Cluster HA for MySQL Packages by Using the <code>scsintall</code> Utility” on page 19
Register and Configure Sun Cluster HA for MySQL	“How to Register and Configure Sun Cluster HA for MySQL as a Failover Service” on page 20
Verify Sun Cluster HA for MySQL Installation and Configuration	“How to Verify the Sun Cluster HA for MySQL Installation and Configuration” on page 25
Understand Sun Cluster HA for MySQL fault monitor	“Understanding the Sun Cluster HA for MySQL Fault Monitor” on page 25
Debug Sun Cluster HA for MySQL	“Debug Sun Cluster HA for MySQL” on page 26

**TABLE 1-1** Task Map: Installing and Configuring Sun Cluster HA for MySQL (Continued)

<b>Task</b>	<b>For Instructions, Go To</b>
Upgrade to SC3.1 when using Sun Cluster HA for MySQL	"Upgrade to SC3.1 when using Sun Cluster HA for MySQL" on page 28
Upgrade to MySQL 4.0.12 from 3.23.54 when using Sun Cluster HA for MySQL	"Upgrade to MySQL 4.0.12 from 3.23.54 when using Sun Cluster HA for MySQL" on page 31

---

## Sun Cluster HA for MySQL Overview

The MySQL software delivers a very 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 can be downloaded 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 1-2** Protection of Components

<b>Component</b>	<b>Protected by</b>
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



---

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

---

Consider the restrictions in this section to plan the installation and configuration of Sun Cluster HA for MySQL. 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*.

- The Sun Cluster for MySQL *can only be configured as a HA and not a Scalable*.
- The MySQL configuration file (my.cnf) should only be placed in the MySQL Database directory. If my.cnf has to be placed in local filesystem then create a symlink from the MySQL Database directory. No my.cnf should be in /etc (Global file) because it could override command line options
- **Supported configurations** – Currently, only 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, ie from <http://www.mysql.com> or from other source, the following restrictions apply.
  - Each MySQL instance must have a unique *Database directory*. This *Database directory* can be mounted as either a Failover File System or Global File System.

---

**Note** – It is considered best practice when mounting Global File Systems to mount them with the /global prefix and to mount Failover File Systems with the /local prefix. However, be aware that this is simply viewed as best practice.

---

### EXAMPLE 1-1 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.

**EXAMPLE 1-1** MySQL instances with Global File Systems (Continued)

```
# 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 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 /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
```

**EXAMPLE 1-1** MySQL instances with Global File Systems (Continued)

```
-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/rdsk/dg1/vol01 /global/mysql ufs 2
yes global,logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdsk/dg2/vol01 /global/mysql-data-1 ufs 2
yes global,logging
/dev/vx/dsk/dg2/vol01 /dev/vx/rdsk/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 1-2** 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
-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
```

**EXAMPLE 1-2** MySQL instances with Failover File Systems (Continued)

```
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



---

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

---

Use the requirements in this section to plan the installation and configuration of Sun Cluster HA for MySQL. These requirements 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.

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

**TABLE 1-3** MySQL components and their dependencies (via → symbol)

Componet	Description
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.

The MySQL component has two configuration and registration files under `/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 already been applied.

**EXAMPLE 1-3** 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.
#
```

**EXAMPLE 1-3** MySQL configuration and registration file for Sun Cluster (Continued)

```
# 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
# cluster this Resourcegroup can 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=""
:::::::::::::
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
# FMUSER - name of the Mysql fault monitor user
# FMPASS - name of the Mysql fault monitor user password
```

**EXAMPLE 1-3** MySQL configuration and registration file for Sun Cluster (Continued)

```
# 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
# FMUSER=fmuser
# FMPASS=fmuser
# LOGDIR=/global/mysqldata/logs
# CHECK=YES
#
RS=
RG=
BASEDIR=
DATADIR=
MYSQLUSER=
LH=
FMUSER=
FMPASS=
LOGDIR=
CHECK=
HAS_RS=
#
```

- **The 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 you should ensure that at least the following parameters are set.

---

**Note** – The my.cnf file is a important file within MySQL. Therefore you are recommended to 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** has to be set to the defined logical host.
  - **socket** should be defined as /tmp/<logical host>.sock.
  - **binlog-ignore-db** should contain sc3\_test\_database if log-bin option is being used.
- MySQL my.cnf file in [mysqld] section in a slave configuration
  - **bind address** has to be set to the defined logical host.

- **socket** should be defined as /tmp/<logical host>.sock.
- **binlog-ignore-db** should contain 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 we left off on the master during the replication process. This file should be placed on a GFS/FFS

---

## Installing and Configuring MySQL

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

Throughout the following next sections, references will be made to certain directories for MySQL. The following list shows common pathnames for these references as shown in previous examples. Refer to “Configuration Restrictions” on page 7 where these examples are shown.

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

## How to Install and Configure MySQL

Use this procedure to install and configure MySQL.

1. **Determine how MySQL will be deployed in Sun Cluster** – Here you need to determine how MySQL will be deployed.
  - 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** – Once you have determined how MySQL will be deployed within Sun Cluster, you must ensure the Cluster File Systems are mounted.

---

**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, however 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>** — However, 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.

```
# 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 7 for a description of the `<MySQL Database directory>` and to “Installing and Configuring MySQL” on page 14 for a list of common pathnames used from the examples in “Configuration Restrictions” on page 7 and in the further example below.

---

**EXAMPLE 1-4** Directories for MySQL instance

The following example 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`.

**EXAMPLE 1-4** Directories for MySQL instance (Continued)

```
# 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 samples 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, you should proceed to “Installing the Sun Cluster HA for MySQL Packages” on page 18 to install it on SC3.1 and SC3.0U3. Afterwards, return back here to continue the Installation and Configuration of MySQL.

---

The contents of

/opt/SUNWscmys/etc/my.cnf\_sample\_[master|slave] provides a sample MySQL configuration file that you can use to create your MySQL instance <MySQL Databasedirectory>/my.cnf. However, note that 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** — That 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 filepermission 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

Use this procedure to verify the installation and configuration. 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. To do this 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 20.

---

#### 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 3.1 10/03 Software Installation Guide*.

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

- The Web Start program
- The `scinstall` utility

---

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

---

## 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.

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. **Load 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/scdataservices\_3\_1\_vb 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/scdataservices_3_1_vb/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 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. **Unload 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 by Using the scsinstall Utility

Use this procedure to install the Sun Cluster HA for MySQL packages by using scsinstall utility on SC3.1. 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 20.

Otherwise, use this procedure to install the Sun Cluster HA for MySQL packages. Perform this procedure on all nodes that can run Sun Cluster HA for MySQL data service.

- 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

Use this procedure to configure Sun Cluster HA for MySQL as a failover data 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 by Using the `scsintall` Utility” on page 19.

Otherwise, use this procedure to configure the Sun Cluster HA for MySQL data service as a failover service.

**1. Become superuser on one of the nodes in the cluster that will host MySQL.**

**2. Register the SUNW.gds resource 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 resourcegroup 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, ie
# 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 th
e
# 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 a
nd
# 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 the MYSQL\_NIC\_HOSTNAME parameter can be omitted.

---

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

```
# ./mysql_register
```

### 11. Stop the MySQL Server instance manually

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

---

**Note** – If you only want to access the MySQL instance through the socket (localhost) this step can be omitted.

---

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

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

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 SC3.1 is being used you should 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 is different then the hostname for the physical interface should be used.

---

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 the `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

Use this procedure to verify that you installed and configured Sun Cluster HA for MySQL correctly.

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

Use the information in this section to understand 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**
  - 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 if the MySQL instance is connected to it's 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 operation 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 debug on 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 debug on for all MySQL instances or a particular MySQL instance.

Each MySQL component has a DEBUG file under `/opt/SUNWscmys/etc`, where `xxx` is a three character abbreviation for the respective MySQL component.

These files allow you to turn debug on 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.

### 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`. Note that the output below, from the command `grep daemon /etc/syslog.conf`, shows that `daemon.debug` has now 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`

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, simply 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** – The step showed will *not* describe how to upgrade to SC3.1 only the steps to upgrade Sun Cluster HA for MySQL to SC3.1 are mentioned.

---

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 documentations.**

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

```
#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 &
```

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, ie

# 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` you must now 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 sourceaddress for adminuser for accessing locally a MySQL instance with a MySQL Logicalhost.

---

**Note** – If you only want to access the MySQL instance through the socket (localhost) this step can be omitted.

---

When SC3.0U3 has been used you should delete the root user belonging to the logicalhost and add a adminusers belonging to the physicalhost.

---

**Note** – If the nodename and the hostname for the physical interface is different then the hostname for the physical interface should be used.

---

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 4.0.12 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 step showed will *not* describe how to upgrade to 4.0.12 only the steps to upgrade Sun Cluster HA for MySQL to 4.0.12 are mentioned. This steps assumes that the new MySQL binaries is installed in the same place. If the new binaries is installed in a new directory the MySQL resource has to be re-registered with new *MySQL Basedir*.

---

**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 MySQL Documentation in how 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, 24

## B

Bootstrap, 16

## C

C locale, 19

Command

chmod, 17

chown, 15, 17

groupadd, 15

scrgadm, 21

scstat, 25

scswitch, 21, 25

useradd, 15

Configuration

MySQL, 14

Registering Sun Cluster HA for MySQL, 20

Requirements, 11

Restrictions, 7

Supported configuration, 7

Verify, 17

## D

Debugging, How to turn on, 26

## F

Fault Monitor

Probing Algorithm and Functionality, 26

Resource Properties, 26

Understanding, 25

files, installation logs, 19

## I

Install, 7, 14

Installation, Sun Cluster 3.1, 19

installing

Sun Cluster HA for MySQL

by using Web Start program, 18

log files created, 19

## L

locales, 19

log files, installation, 19

## M

MySQL, 6

## R

Resource Types

SUNW.gds, 21

Resource Types (Continued)  
SUNW.HAStoragePlus, 21

## **S**

scinstall, 20

## **U**

Upgrade

Upgrade to MySQL 4.0.12 from 3.23.54, 31

Upgrade to SC3.1 when using Sun Cluster  
HA for MySQL, 28

## **V**

/var/sadm/install/logs directory, 19

## **W**

Web Start program, 18