



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: 820-5027-10
January 2009, Revision A

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

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.

This distribution may include materials developed by third parties.

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, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries 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.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, 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.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

Preface	7
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
MySQL and Solaris Containers	13
Configuration Restrictions	13
Configuration Requirements	18
Installing and Configuring MySQL	23
▼ How to Enable MySQL to run in a Global Zone Configuration	24
▼ How to Install and Configure MySQL in a Global Zone	24
▼ How to Enable MySQL to run in a Zone Configuration	27
▼ How to Install and Configure MySQL in a Zone	27
▼ How to Enable MySQL to run in an HA Container Configuration	30
▼ How to Install and Configure MySQL in an HA Container	31
Verifying the Installation and Configuration of MySQL	34
▼ How to Verify the Installation and Configuration of MySQL	34
Installing the Sun Cluster HA for MySQL Packages	35
▼ How to Install the Sun Cluster HA for MySQL Packages	35
Registering and Configuring Sun Cluster HA for MySQL	37
▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Global Zone Configuration	37
▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Zone Configuration	42
▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in an HA Container Configuration	47
▼ How to Modify Parameters in the Sun Cluster HA for MySQL Manifest	52

▼ How to Remove a Sun Cluster HA for MySQL Resource From an HA container	52
Verifying the Sun Cluster HA for MySQL Installation and Configuration	53
▼ How to Verify the Sun Cluster HA for MySQL Installation and Configuration	53
Understanding the Sun Cluster HA for MySQL Fault Monitor	54
Resource Properties	54
Probing Algorithm and Functionality	54
Debug Sun Cluster HA for MySQL	55
▼ How to Activate Debugging for Sun Cluster HA for MySQL	55
Upgrade to Sun Cluster 3.2 when Using Sun Cluster HA for MySQL	56
▼ Upgrade from Sun Cluster 3.0 to Sun Cluster 3.2 when Using Sun Cluster HA for MySQL	56
Upgrade to MySQL 4.x.x from 3.23.54 when using Sun Cluster HA for MySQL	60
▼ Upgrade to MySQL 4.x.x from 3.23.54	60
 A Deployment Example: Installing MySQL in the Global Zone	61
Target Cluster Configuration	61
Software Configuration	61
Assumptions	62
Installing and Configuring MySQL on Local Storage in the Global Zone	62
▼ Example: Preparing the Cluster for MySQL	62
▼ Example: Configuring Cluster Resources for MySQL	63
▼ Example: Installing and bootstrapping the MySQL Software on Local Storage	63
▼ Example: Modifying the MySQL Configuration Files	65
▼ Example: Enabling the MySQL Software to Run in the Cluster	66
 B Deployment Example: Installing MySQL in the Non-Global HA Container	69
Target Cluster Configuration	69
Software Configuration	69
Assumptions	70
Installing and Configuring MySQL on Local Storage in the HA Container	70
▼ Example: Preparing the Cluster for MySQL	70
▼ Example: Configuring Cluster Resources for MySQL	71
▼ Example: Configuring the HA Container	71
▼ Example: Installing and bootstrapping the MySQL Software on Local Storage	73
▼ Example: Modifying the MySQL Configuration Files	75

▼ Example: Enabling the MySQL Software to Run in the Cluster	75
C Deployment Example: Installing MySQL in a Non-Global Zone	79
Target Cluster Configuration	79
Software Configuration	79
Assumptions	80
Installing and Configuring MySQL on Local Storage in a Non-Global Zone	80
▼ Example: Preparing the Cluster for MySQL	80
▼ Example: Configuring the Zone	80
▼ Example: Configuring Cluster Resources for MySQL	82
▼ Example: Installing the MySQL Software on Local Storage	82
▼ Example: Bootstrapping the MySQL Software on Local Storage	83
▼ Example: Modifying the MySQL Configuration File	85
▼ Example: Enabling the MySQL Software to Run in the Cluster	85
Index	89

Preface

Sun Cluster Data Service for MySQL Guide for Solaris OS explains how to install and configure Sun[™] Cluster HA for MySQL.

Note – This Sun Cluster release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, and Intel 64. In this document, x86 refers to the larger family of 64-bit x86 compatible products. Information in this document pertains to all platforms unless otherwise specified.

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 book assume knowledge of the Solaris[™] Operating System (Solaris OS) and expertise with the volume-manager software that is used with Sun Cluster software.

Using 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 conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <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>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

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

TABLE P-2 Shell Prompts

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

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 3.1 - 3.2 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

The Sun web site provides information about the following additional resources:

- [Documentation](http://www.sun.com/documentation/) (<http://www.sun.com/documentation/>)
- [Support](http://www.sun.com/support/) (<http://www.sun.com/support/>)
- [Training](http://www.sun.com/training/) (<http://www.sun.com/training/>)

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to <http://docs.sun.com> and click Feedback.

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 number and serial number of your systems
- The release number of the Solaris Operating System (for example, Solaris 10)
- The release number of Sun Cluster (for example, Sun Cluster 3.2)

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

Command	Function
<code>prtconf -v</code>	Displays the size of the system memory and reports information about peripheral devices
<code>psrinfo -v</code>	Displays information about processors
<code>showrev -p</code>	Reports which patches are installed
<code>prtdiag -v</code>	Displays system diagnostic information
<code>/usr/cluster/bin/clnode show-rev</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 23
- “Verifying the Installation and Configuration of MySQL” on page 34
- “Installing the Sun Cluster HA for MySQL Packages” on page 35
- “Registering and Configuring Sun Cluster HA for MySQL” on page 37
- “Verifying the Sun Cluster HA for MySQL Installation and Configuration” on page 53
- “Understanding the Sun Cluster HA for MySQL Fault Monitor” on page 54
- “Debug Sun Cluster HA for MySQL” on page 55
- “Upgrade to Sun Cluster 3.2 when Using Sun Cluster HA for MySQL” on page 56
- “Upgrade to MySQL 4.x.x from 3.23.54 when using Sun Cluster HA for MySQL” on page 60

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

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

Task	For Instructions, Go To
2. Install and configure MySQL.	“How to Install and Configure MySQL in a Global Zone” on page 24
3. Verify installation and configuration.	“How to Verify the Installation and Configuration of MySQL” on page 34
4. Install Sun Cluster HA for MySQL Packages.	“Installing the Sun Cluster HA for MySQL Packages” on page 35
5. Register and Configure Sun Cluster HA for MySQL.	“How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Global Zone Configuration” on page 37
6. Verify Sun Cluster HA for MySQL Installation and Configuration.	“How to Verify the Sun Cluster HA for MySQL Installation and Configuration” on page 53
7. Understand Sun Cluster HA for MySQL fault monitor.	“Understanding the Sun Cluster HA for MySQL Fault Monitor” on page 54
8. Debug Sun Cluster HA for MySQL.	“Debug Sun Cluster HA for MySQL” on page 55
9. Upgrade to Sun Cluster 3.2 when using Sun Cluster HA for MySQL.	“Upgrade to Sun Cluster 3.2 when Using Sun Cluster HA for MySQL” on page 56
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 60

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.

MySQL and Solaris Containers

Sun Cluster HA for MySQL is supported in Solaris Containers. Sun Cluster offers the following concepts for Solaris Containers.

- Zones are containers which are running after a reboot of the node. These containers, combined with resource groups having the node name *nodename:zonename* as a valid “node name” in the resource groups node name list.
- HA Containers are managed by the Solaris Container agent, and are represented by a resource of a resource group.

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.

Restriction for the Sun Cluster HA for MySQL Data Service Configuration

The Sun Cluster for MySQL can be configured only as a failover data service and not as a scalable data service.

Restrictions on the MySQL Configuration File

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 symbolic link 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

Restrictions for the MySQL 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.

Restrictions on the MySQL Database Directory

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 always a good practice to mount the Global File Systems with the `/global` prefix and to mount the Failover File Systems with the `/local` prefix.

Restriction for the MySQL smf Service Name in an HA Container

The MySQL configuration in an HA container uses the smf component of Sun Cluster HA for Solaris Containers. The registration of the MySQLdata service in an HA container defines an smf service to control the MySQL database. The name of this smf service is generated in this naming scheme: `svc:/application/sczone-agents:resource-name`. No other smf service with exactly this name can exist.

The associated smf manifest is automatically created during the registration process in this location and naming scheme:
`/var/svc/manifest/application/sczone-agents/resource-name.xml`. No other manifest can coexist with this name.

Examples for the File Systems Layout

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
```

EXAMPLE 1 MySQL instances with Global File Systems (Continued)

```

#
# 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

```

EXAMPLE 1 MySQL instances with Global File Systems (Continued)

```

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
-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
```

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

```

-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 Between Sun Cluster HA for MySQL Components in Failover Configurations	
Component	Dependency
MySQL resource in a Solaris 10 global zone,zone or in Solaris 9.	SUNW.HAStoragePlus This dependency is required only, if the configuration uses a failover file system or file systems in a zone.
	SUNW.LogicalHostName
MySQL resource in a Solaris 10 HA container.	Sun Cluster HA for the Solaris Container boot resource.
	SUNW.HAStoragePlus
	SUNW.LogicalHostName — This dependency is required only if the zones boot resource does not manage the zone's IP address.

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
# more util/*config
:::::::::::::
ha_mysql_config
:::::::::::::
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident    "@(#)ha_mysql_config.ksh 1.3    06/03/08 SMI"

# 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
#
#     To have the mysql agent local zone aware, 4 Variables are needed:
#     ZONE - the zone name where the Mysql Database should run in
#           Optional
#     ZONEBT - The resource name which controls the zone.
#             Optional
#     PROJECT - A project in the zone, that will be used for this service
#               specify it if you have an su - in the start stop or probe,
#               or to define the smf credentials. If the variable is not set,
#               it will be translated as :default for the sm and default
#               for the zsh component
#             Optional
#     ZUSER - A user in the the zone which is used for the smf method
#             credentials. Yur smf servic e will run under this user
#             Optional
#
# Mysql specific Variables
#
#     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 Logical Hostname Sun Cluster 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
```

EXAMPLE 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 Sun Cluster resource
#
# The following examples illustrate sample parameters
# for Mysql
#
# BASEDIR=/usr/local/mysql
# DATADIR=/global/mysqldata
# MYSQLUSER=mysql
# LH=mysqllh
# MYSQLHOST=mysqllh
# FMUSER=fmuser
# FMPASS=fmuser
# LOGDIR=/global/mysqldata/logs
# CHECK=YES
#

RS=
RG=
PORT=
LH=
HAS_RS=

# local zone specific options

ZONE=
ZONE_BT=
PROJECT=

# mysql specifications

BASEDIR=
DATADIR=
MYSQLUSER=
MYSQLHOST=
FMUSER=
FMPASS=
LOGDIR=
CHECK=
:::::::::::::
mysql_config
:::::::::::::
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
```

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

```

#

#ident    "@(#)mysql_config.ksh 1.3      06/03/08 SMI"

# 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 (Default is 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=

# Specify the physical hostname for the physical NIC that this
logical hostname
# belongs to for every node in the cluster this Resource group can be
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-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"
MYSQL_NIC_HOSTNAME=

# Where are your databases installed? (location of my.cnf)
MYSQL_DATADIR=

#

```

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 host's IP name.

Note – Some MySQL do not work with **bind address** if the name of the logical host is set. In these cases do not set the **bind address** parameter, or use the absolute address of the logical hostname.

- **socket** is defined as */tmp/logical host's 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 host's IP name.

Note – Some MySQL builds do not work with names in the **bind address** parameter. In these cases do not set the **bind address** parameter, or use the absolute IP address of the logical hostname.

- **socket** is defined as */tmp/logical host's 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 path names for these references. For more information, refer to the [“Configuration Restrictions” on page 13](#) section.

- 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

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.
- Determine the type of the target zone where you will install MySQL. Valid zone types are, the global zone, the HA container, or a zone.

To install and configure MySQL in a *global zone* configuration, complete the following tasks:

- [“How to Enable MySQL to run in a Global Zone Configuration” on page 24](#)
- [“How to Install and Configure MySQL in a Global Zone” on page 24](#)

To install and configure MySQL in a zone configuration, complete the following tasks:

- [“How to Enable MySQL to run in a Zone Configuration” on page 27](#)
- [“How to Install and Configure MySQL in a Zone” on page 27](#)

To install and configure MySQL in an HA container configuration, complete the following tasks:

- [“How to Enable MySQL to run in an HA Container Configuration” on page 30](#)
- [“How to Install and Configure MySQL in an HA Container” on page 31](#)

You will find installation examples for each zone type in:

- [Deployment Example: Installing MySQL in the Global Zone](#)
- [Deployment Example: Installing MySQL in the Non-Global HA Container](#)
- [Deployment Example: Installing MySQL in a Non-Global Zone](#)

▼ How to Enable MySQL to run in a Global Zone Configuration

- 1 Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on one of the nodes in the cluster that will host MySQL.

- 2 Register the *SUNW.gds* and *SUNW.HAStoragePlus* resource type.

```
# clresourcetype register SUNW.gds SUNW.HAStoragePlus
```

- 3 Create a failover resource group.

```
# clresourcegroup create MySQL-failover-resource-group
```

- 4 Create a resource for the MySQL Disk Storage.

```
# clresource create \  
-g MySQL-failover-resource-group \  
-t SUNW.HAStoragePlus \  
-p FilesystemMountPoints=MySQL-instance-mount-points MySQL-has-resource
```

- 5 Create a resource for the MySQL Logical Hostname.

```
# clreslogicalhostname \  
-g MySQL-failover-resource-group \  
-h MySQL-logical-hostname \  
MySQL-lh-resource
```

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

```
# clresourcegroup online -M -n current-node MySQL-failover-resource-group
```

▼ How to Install and Configure MySQL in a Global Zone

- 1 Insure that you are on the node where you enabled your resource group.

- 2 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.

3 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
```

4 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
```

5 Create your MySQL Database directory for your MySQL Instance or Instances.

```
# mkdir MySQL Database directory
```

Note – Refer to the “[Configuration Restrictions](#)” on page 13 section for a description of the *MySQL Database directory* and to the “[Installing and Configuring MySQL](#)” on page 23 section for a list of common path names.

The following listing 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
#
```

- 6 **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 35](#) to install it on your cluster. 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 Database directory/my.cnf
```

- 7 **Bootstrap MySQL instance — This creates the privilege tables db, host, user, tables_priv, and columns_priv in the mysql database, as well as the func table.**

```
# cd MySQL Base directory

# ./scripts/mysql_install_db \
  --datadir=MySQL Database directory
```

- 8 **Create a logfile directory in MySQL Database Directory.**

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

- 9 **Create directories for your storage engines.**

```
# mkdir MySQL Database Directory/innodb
# mkdir MySQL Database Directory/BDB
```

- 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
```

▼ How to Enable MySQL to run in a Zone Configuration

- 1 Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on one of the nodes in the cluster that will host MySQL.

- 2 Create and boot your zone *MySQL-zone* on all the nodes to host your MySQL data base.

- 3 Register the *SUNW.gds* and *SUNW.HAStoragePlus* resource type.

```
# clresource type register SUNW.gds SUNW.HAStoragePlus
```

- 4 Create a failover resource group.

```
# clresourcegroup create \
-n node1:MySQL-zone,node2:MySQL-zone \
MySQL-failover-resource-group
```

- 5 Create a resource for the MySQL Disk Storage.

```
# clresource create \
-g MySQL-failover-resource-group \
-t SUNW.HAStoragePlus \
-p FilesystemMountPoints=MySQL-instance-mount-points MySQL-has-resource
```

- 6 Create a resource for the MySQL Logical Hostname.

```
# clreslogicalhostname \
-g MySQL-failover-resource-group \
-h MySQL-logical-hostname \
MySQL-lh-resource
```

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

```
# clresourcegroup online -M -n current-node MySQL-failover-resource-group
```

▼ How to Install and Configure MySQL in a Zone

- 1 Insure that you are on the node where you enabled your resource group.

- 2 Log in to your zone.

```
# zlogin MySQL-zone
```

- 3 Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on one of the nodes in the cluster that will host MySQL.

4 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.

5 Create a mysql-user and mysql-group for MySQL on all nodes:zones 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
```

6 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
```

7 Create your MySQL Database directory for your MySQL Instance or Instances.

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

Note – Refer to the “[Configuration Restrictions](#)” on page 13 section for a description of the *MySQL Database directory* and to the “[Installing and Configuring MySQL](#)” on page 23 section for a list of common path names.

The following listing 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
#

```

- 8 **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 35](#) to install it on your cluster. 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 Database directory>/my.cnf
```

- 9 **Bootstrap MySQL instance — This creates the privilege tables db, host, user, tables_priv and columns_priv in the mysql database, as well as the func table.**

```
# cd <MySQL Base directory>
```

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

- 10 **Create a logfile directory in MySQL Database Directory**

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

- 11 **Create directories for your storage engines.**

```
# mkdir <MySQL Database Directory>/innodb
# mkdir <MySQL Database Directory>/BDB
```

12 Change owner and group for <MySQL Database Directory>

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

13 Change file permission for <MySQL Database Directory>/my.cnf

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

▼ How to Enable MySQL to run in an HA Container Configuration

1 Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on one of the nodes in the cluster that will host MySQL.**2 As superuser register the `SUNW.HAStoragePlus` and the `SUNW.gds` resource types.**

```
# clresourcetype register SUNW.HAStoragePlus SUNW.gds
```

3 Create a failover resource group.

```
# clresourcegroup create MySQL-resource-group
```

4 Create a resource for the MySQL zone's disk storage.

```
# clresource create -t SUNW.HAStoragePlus \  
-p FileSystemMountPoints=MySQL-instance-mount-points \  
MySQL-has-resource
```

5 (Optional) If you want the protection against a total adapter failure for your public network, create a resource for the MySQL's logical hostname.

```
# clreslogicalhostname create -g MySQL-resource-group \  
-h logical-hostname \  
MySQL-logical-hostname-resource-name
```

6 Place the resource group in the managed state.

```
# clresourcegroup online -M MySQL-resource-group
```

7 Install the zone.

Install the zone according to the Sun Cluster HA for Solaris Containers agent documentation, assuming that the resource name is `MySQL-zone-rs` and that the zone name is `MySQL-zone`.

8 Verify the zone's installation.

```
# zoneadm -z MySQL-zone boot  
# zoneadm -z MySQL-zone halt
```

9 Register the zone's boot component.

a. Copy the container resource boot component configuration file.

```
# cp /opt/SUNWsczone/sczbt/util/sczbt_config zones-target-configuration-file
```

b. Use a plain text editor to set the following variables:

```
RS=MySQL-zone-rs
RG=MySQL-resource-group
PARAMETERDIR=MySQL-zone-parameter-directory
SC_NETWORK=true|false
SC_LH=MySQL-logical-hostname-resource-name
FAILOVER=true|false
HAS_RS=MySQL-has-resource
Zonename=MySQL-zone
Zonebootopt=zone-boot-options
Milestone=zone-boot-milestone
Mounts=
```

c. Create the parameter directory for your zone's resource.

```
# mkdir MySQL-zone-parameter-directory
```

d. Execute the Sun Cluster HA for Solaris Container's registration script.

```
# /opt/SUNWsczone/sczbt/util/sczbt_register -f zones-target-configuration-file
```

e. Enable the Solaris Container resource

```
# clresource enable MySQL-zone-rs
```

10 Enable the resource group.

```
# clresourcegroup online MySQL-resource-group
```

▼ How to Install and Configure MySQL in an HA Container

1 Insure that you are on the node where your enabled your resource group.

2 Log in to your zone

```
# zlogin MySQL-zone
```

3 Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on one of the nodes in the cluster that will host MySQL.

4 Install MySQL.

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.

5 Create a mysql-user and mysql-group for MySQL zone that will run MySQL.

Create an entry in /etc/group in the zone.

```
# 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
```

6 Change owner and group for MySQL binaries.

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

7 Create your MySQL Database directory for your MySQL Instance or Instances.

```
# mkdir MySQL Database directory
```

Note – Refer to the “[Configuration Restrictions](#)” on page 13 section for a description of the *MySQL Database directory* and to the “[Installing and Configuring MySQL](#)” on page 23 section for a list of common path names.

The following listing 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
#

```

- 8 **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 35](#) to install it on your cluster. 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 Database directory>/my.cnf

```

- 9 **Bootstrap MySQL instance — This creates the privilege tables db, host, user, tables_priv and columns_priv in the mysql database, as well as the func table.**

```

# cd <MySQL Base directory>

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

```

- 10 **Create a logfile directory in MySQL Database Directory.**

```

# mkdir MySQL Database Directory/logs

```

- 11 **Create directories for your storage engines.**

```

# mkdir MySQL Database Directory/innodb
# mkdir MySQL Database Directory/BDB

```

- 12 **Change owner and group for MySQL Database Directory.**

```

# chown -R mysql:mysql MySQL Database Directory

```

13 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 the steps 1 to 6 the task [“How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Global Zone Configuration”](#) on page 37, the steps 1 to 7 in the task [“How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in an HA Container Configuration”](#) on page 47 or the step 1 to 7 in the task [“How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Zone Configuration”](#) on page 42 according to your zone type.

1 (Optional) Log in to your target zone.

```
# zlogin mysql-zone
```

2 Start the MySQL Server for this instance.

```
#cd <MySQL Base directory>
```

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

3 Connect to the MySQL instance.

```
# <MySQL Base directory>/bin/mysql -S /tmp/<Logical host>.sock -uroot
```

4 Stop the MySQL server instance.

```
# kill -TERM 'cat <MySQL Database directory>/mysqld.pid
```

5 (Optional) Leave the target zone.

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. To install the packages, use the Sun Java™ Enterprise System Installation Wizard.

Note – You need to install the Sun Cluster HA for MySQL packages in the global cluster and not in the zone cluster.

▼ How to Install the Sun Cluster HA for MySQL Packages

Perform this procedure on each cluster node where you are installing the Sun Cluster HA for MySQL packages.

You can run the Sun Java Enterprise System Installation Wizard 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.

Note – Even if you plan to configure this data service to run in non-global zones, install the packages for this data service in the global zone. The packages are propagated to any existing non-global zones and to any non-global zones that are created after you install the packages.

Before You Begin Ensure that you have the Sun Java Availability Suite DVD-ROM.

If you intend to run the Sun Java Enterprise System Installation Wizard with a GUI, ensure that your DISPLAY environment variable is set.

- 1 **On the cluster node where you are installing the data service packages, become superuser.**
- 2 **Load the Sun Java Availability Suite DVD-ROM into the DVD-ROM drive.**
If the Volume Management daemon `vold(1M)` is running and configured to manage DVD-ROM devices, the daemon automatically mounts the DVD-ROM on the `/cdrom` directory.
- 3 **Change to the Sun Java Enterprise System Installation Wizard directory of the DVD-ROM.**
 - **If you are installing the data service packages on the SPARC® platform, type the following command:**

```
# cd /cdrom/cdrom0/Solaris_sparc
```

- If you are installing the data service packages on the x86 platform, type the following command:

```
# cd /cdrom/cdrom0/Solaris_x86
```

4 Start the Sun Java Enterprise System Installation Wizard.

```
# ./installer
```

5 When you are prompted, accept the license agreement.

If any Sun Java Enterprise System components are installed, you are prompted to select whether to upgrade the components or install new software.

6 From the list of Sun Cluster agents under Availability Services, select the data service for MySQL.

7 If you require support for languages other than English, select the option to install multilingual packages.

English language support is always installed.

8 When prompted whether to configure the data service now or later, choose Configure Later.

Choose Configure Later to perform the configuration after the installation.

9 Follow the instructions on the screen to install the data service packages on the node.

The Sun Java Enterprise System Installation Wizard displays the status of the installation. When the installation is complete, the wizard displays an installation summary and the installation logs.

10 (GUI only) If you do not want to register the product and receive product updates, deselect the Product Registration option.

The Product Registration option is not available with the CLI. If you are running the Sun Java Enterprise System Installation Wizard with the CLI, omit this step.

11 Exit the Sun Java Enterprise System Installation Wizard.

12 Unload the Sun Java Availability Suite DVD-ROM from the DVD-ROM drive.

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

- b. Eject the DVD-ROM.

```
# eject cdrom
```

Next Steps See [“Registering and Configuring Sun Cluster HA for MySQL” on page 37](#) to register Sun Cluster HA for MySQL and to configure the cluster for the data service.

Registering and Configuring Sun Cluster HA for MySQL

This section contains the procedures you need to configure Sun Cluster HA for MySQL. According to your zone type you need to complete either one of the following tasks.

- “How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Global Zone Configuration” on page 37
- “How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Zone Configuration” on page 42
- “How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in an HA Container Configuration” on page 47

▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in a Global Zone Configuration

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 [“Installing the Sun Cluster HA for MySQL Packages” on page 35](#).

- 1 **Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on the node in the cluster that host MySQL.**
- 2 **Start the MySQL Server instance manually.**

```
#cd <MySQL Base directory>
```

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

- 3 **Configure the admin password for the admin user.**

```
# <MySQL Database directory>/bin/mysqladmin \
-S /tmp/<Logical host>.sock password 'admin password'
```

- 4 **Add the admin user 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* creates two admin users, one belonging to localhost and one belonging to the node on which *mysql_install_db* was executed.

Add an admin user for every physical node in the cluster that runs this MySQL instance.

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

The following is an example for a MySQL instance on Sun Cluster 3.2.

```
# mysql -S /tmp/hahostix1.sock -uroot -p'admin-password'
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'clusterix2' IDENTIFIED BY 'rootpasswd';
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 the MySQL Administration documentation.

The following is an example for a MySQL 4.x or 5.0.x on Sun Cluster 3.2.

```
# mysql -S /tmp/hahostix1.sock -uroot -p'admin-password'
mysql> use mysql;
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='clusterix1';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='clusterix2';
mysql> exit;
```

Note – If you experience any problems here, refer to the MySQL administration documentation.

5 Copy the MySQL configuration files to your home directory.

```
# cp /opt/SUNWscmys/util/ha_mysql_config /my-place
# cp /opt/SUNWscmys/util/mysql_config /my-place
```

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

```
# cd my-place
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident "@(#)mysql_config.ksh 1.3 06/03/08 SMI"

# 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 (Default is 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=

# Specify the physical hostname for the physical NIC that this
logical hostname
# belongs to for every node in the cluster this Resource group can be
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-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"
MYSQL_NIC_HOSTNAME=
# Where are your databases installed? (location of my.cnf)
MYSQL_DATADIR
```

Note – The following is an example for a MySQL instance on Sun Cluster 3.2.

```

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"
MYSQL_DATADIR=/global/mysql-data

```

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

```
# /opt/SUNWscmys/util/mysql_register -f my-place/mysql_config
```

7 Stop the MySQL Server instance manually.

```
# kill -TERM 'cat <MySQL Database directry>/mysqld.pid'
```

8 Create and register MySQL as a failover data service.

```
# cd my-place
```

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

```

#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident    "@(#)ha_mysql_config.ksh 1.3      06/03/08 SMI"

# 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
#
#       To have the mysql agent local zone aware, 4 Variables are needed:
#       ZONE - the zone name where the Mysql Database should run in
#       Optional
#       ZONEBT - The resource name which controls the zone.
#       Optional
#       PROJECT - A project in the zone, that will be used for this service
#       specify it if you have an su - in the start stop or probe,

```

```

#           or to define the smf credentials. If the variable is not set,
#           it will be translated as :default for the sm and default
#           for the zsh component
#           Optional
#       ZUSER - A user in the the zone which is used for the smf method
#               credentials. Yur smf servic e will run under this user
#               Optional
#
# Mysql specific Variables
#
#   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 Logical Hostname Sun Cluster 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 Sun Cluster resource
#

```

The following is an example for a 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
ZONE=
ZONE_BT=
PROJECT=

```

Register the MySQL resource.

```
# /opt/SUNWscmys/util/ha_mysql_register -f my-place/ha_mysql_config
```

9 Enable each MySQL resource.

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

```

# clresource status
# clresource enable MySQL-resource

```

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

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 [“Installing the Sun Cluster HA for MySQL Packages” on page 35](#).

- 1 **Log in to the zone on the node that hosts your *MySQL* resource-group**
- 2 **Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on the nodes zone in the cluster that host MySQL.**

- 3 **Start the MySQL Server instance manually.**

```
# cd <MySQL Base directory>

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

- 4 **Configure the admin password for the admin user.**

```
# <MySQL Database directry>/bin/mysqladmin \
-S /tmp/<Logical host>.sock password 'admin password'
```

- 5 **Add the admin user 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` creates two admin users, one belonging to localhost and one belonging to the node on which `mysql_install_db` was executed. Add an admin user for every physical node in the cluster that runs this MySQL instance.

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

The following is an example for a MySQL instance on Sun Cluster 3.2.

```
# mysql -S /tmp/hahostix1.sock -uroot -p'adminpassword'
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'zone1' IDENTIFIED BY 'rootpasswd';
mysql> GRANT ALL ON *.* TO 'root'@'zone2' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

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

The following is an example for a MySQL 4.x or 5.0.x on Sun Cluster 3.2.

```
# mysql -S /tmp/hahostix1.sock -uroot -p'adminpassword'
mysql> use mysql;
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone1';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone2';
mysql> exit;
```

Note – If you experience any problems here, refer to the MySQL administration documentation.

6 Copy the MySQL configuration file to your home directory.

```
# cp /opt/SUNWscmys/util/mysql_config /my-place
```

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

```
# cd my-place
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident    "@(#)mysql_config.ksh 1.3      06/03/08 SMI"

# 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 (Default is 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=

# Specify the physical hostname for the physical NIC that this
logical hostname

# belongs to for every node in the cluster this Resource group can be
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-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.

# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"

MYSQL_NIC_HOSTNAME=

# Where are your databases installed? (location of my.cnf)

MYSQL_DATADIR=

```

Note – The following is an example for a MySQL instance on Sun Cluster 3.2.

```

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="zone1 zone2"
MYSQL_DATADIR=/global/mysql-data
```

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

```
# /opt/SUNWscmys/util/mysql_register -f my-place/mysql_config
```

8 Stop the MySQL Server instance manually.

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

9 Leave the zone and become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on the nodes global zone in the cluster that host MySQL.

10 Copy the MySQL configuration file to your home directory.

```
# cp /opt/SUNWscmys/util/ha_mysql_config /my-place
```

11 Create and register MySQL as a failover data service.

```
# cd my-place
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident "@(#)ha_mysql_config.ksh 1.3 06/03/08 SMI"

# 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
#
# To have the mysql agent local zone aware, 4 Variables are needed:
# ZONE - the zone name where the Mysql Database should run in
# Optional
# ZONEBT - The resource name which controls the zone.
# Optional
# PROJECT - A project in the zone, that will be used for this service
# specify it if you have an su - in the start stop or probe,
# or to define the smf credentials. If the variable is not set,
# it will be translated as :default for the sm and default
# for the zsh component
# Optional
```

```
#      ZUSER - A user in the the zone which is used for the smf method
#              credentials. Yur smf servic e will run under this user
#              Optional
#
# Mysql specific Variables
#
#   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 Logical Hostname Sun Cluster 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 Sun Cluster resource
#
```

The following is an example for a 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
ZONE=
ZONE_BT=
PROJECT=
```

Register the MySQL resource.

```
# /opt/SUNWscmys/util/ha_mysql_register -f my-place/ha_mysql_config
```

12 Enable each MySQL resource.

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

```
# clresource status
# clresource enable MySQL-resource
```

▼ How to Register and Configure Sun Cluster HA for MySQL as a Failover Service in an HA Container Configuration

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 [“Installing the Sun Cluster HA for MySQL Packages”](#) on page 35.

- 1 **Log in to the zone on the node that hosts your *MySQLresource-group***
- 2 **Become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on the nodes zone in the cluster that host MySQL.**

- 3 **Start the MySQL Server instance manually.**

```
# cd <MySQL Base directory>

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

- 4 **Configure the admin password for the admin user.**

```
# <MySQL Databasedirectory>/bin/mysqladmin \
-S /tmp/<Logical host>.sock password 'admin password'
```

- 5 **Add the admin user 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` creates two admin users, one belonging to localhost and one belonging to the node on which `mysql_install_db` was executed.

Add an admin user for every physical node in the cluster that runs this MySQL instance.

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

The following is an example for a MySQL instance on Sun Cluster 3.2.

```
# mysql -S /tmp/hahostix1.sock -uroot -p'admin-password'
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'zone1' IDENTIFIED BY 'rootpasswd';
mysql> exit;
```

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

The following is an example for a MySQL 4.x or 5.0.x on Sun Cluster 3.2.

```
# mysql -S /tmp/hahostix1.sock -uroot -p'admin-password'
mysql> use mysql;
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone';
mysql> exit;
```

Note – If you experience any problems here, refer to the MySQL administration documentation.

6 Copy the MySQL configuration file to your home directory.

```
# cp /opt/SUNWscmys/util/mysql_config /my-place
```

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

```
# cd my-place
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident    "@(#)mysql_config.ksh 1.3      06/03/08 SMI"

# 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 (Default is 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=

# Specify the physical hostname for the physical NIC that this
logical hostname

# belongs to for every node in the cluster this Resource group can be
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-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.

# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"

MYSQL_NIC_HOSTNAME=

# Where are your databases installed? (location of my.cnf)

MYSQL_DATADIR=
```

Note – The following is an example for a MySQL instance on Sun Cluster 3.2.

```
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="zone1"
MYSQL_DATADIR=/global/mysql-data
```

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

```
# /opt/SUNWscmys/util/mysql_register -f my-place/mysql_config
```

8 Stop the MySQL Server instance manually.

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

9 Leave the zone and become superuser or assume a role that provides `solaris.cluster.verb` RBAC authorization on the nodes global zone in the cluster that host MySQL.

10 Copy the MySQL configuration file to your home directory.

```
# cp /opt/SUNWscmys/util/ha_mysql_config /my-place
```

11 Create and register MySQL as a failover data service.

```
# cd my-place
```

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

```
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#

#ident    "@(#)ha_mysql_config.ksh 1.3      06/03/08 SMI"

# 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
#
#       To have the mysql agent local zone aware, 4 Variables are needed:
#       ZONE - the zone name where the Mysql Database should run in
#             Optional
#       ZONEBT - The resource name which controls the zone.
#             Optional
#       PROJECT - A project in the zone, that will be used for this service
#                 specify it if you have an su - in the start stop or probe,
#                 or to define the smf credentials. If the variable is not set,
#                 it will be translated as :default for the sm and default
#                 for the zsh component
#             Optional
#       ZUSER - A user in the the zone which is used for the smf method
#                 credentials. Yur smf servic e will run under this user
#             Optional
```

```
#
# Mysql specific Variables
#
#   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 Logical Hostname Sun Cluster 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 Sun Cluster resource
#
```

The following is an example for a 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
ZONE=zone1
ZONE_BT=zone1-rs
PROJECT=MySQL-project
```

Register the MySQL resource.

```
# /opt/SUNWscmys/util/ha_mysql_register -f my-place/ha_mysql_config
```

12 Enable each MySQL resource.

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

```
# clresource status
# clresource enable MySQL-resource
```

▼ How to Modify Parameters in the Sun Cluster HA for MySQL Manifest

Perform this task to change parameters in the Sun Cluster HA for MySQL manifest and to validate the parameters in the HA container. Parameters for the Sun Cluster HA for MySQL manifest are stored as properties of the SMF service. To modify parameters in the manifest, change the related properties in the SMF service then validate the parameter changes.

- 1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations on the zones console.**
- 2 **Change the Solaris Service Management Facility (SMF) properties for the Sun Cluster HA for MySQL manifest.**

```
# svccfg svc:/application/sczone-agents:resource
```

For more information, see the [svccfg\(1M\)](#) man page.

- 3 **Validate the parameter changes.**

```
# /opt/SUNWscmys/bin/control_mysql validate resource
```

Messages for this command are stored in the `/var/adm/messages/` directory of the HA container.

- 4 **Disconnect from the HA container's console.**

▼ How to Remove a Sun Cluster HA for MySQL Resource From an HA container

- 1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**
- 2 **Disable and remove the resource that is used by the Sun Cluster HA for MySQL data service.**

```
# clresource disable resource
```

```
# clresource delete resource
```

- 3 **Log in as superuser to the HA container's console.**
- 4 **Unregister Sun Cluster HA for MySQL from the Solaris Service Management Facility (SMF) service.**

```
# /opt/SUNWscmys/util/ha_mysql_smf_remove -f filename
```

-f Specifies the configuration file name.

filename The name of the configuration file that you used to register Sun Cluster HA for MySQL with the SMF service.

Note – If you no longer have the configuration file that you used to register Sun Cluster HA for MySQL with the SMF service, create a replacement configuration file:

- a. Make a copy of the default file, `/opt/SUNWscmys/util/ha_mysql_config`.
 - b. Set the `ZONE` and `RS` parameters with the values that are used by the data service.
 - c. Run the `ha_mysql_smf_remove` command and use the `-f` option to specify this configuration file.
-

- 5 **Disconnect from the HA container's console.**

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

- 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`.**

```
# cluster status
```

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

```
# clresource enable MySQL-resource
```

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

```
# clresourcegroup -h node2 MySQL-failover-resource-group
```

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

Sun Cluster HA for MySQL has a file named `config` that enables you to activate debugging for MySQLresources. This file is in the `/opt/SUNWscmys/etc` directory.

▼ How to Activate Debugging for Sun Cluster HA for MySQL

1 Determine whether you are in a global zone or in an HA container configuration.

If your operating system is Solaris 10 and your MySQL resource is dependent on a Solaris Container boot component resource, you are in an HA container configuration. In any other case, especially on a Solaris 9 system, you are in a global zone configuration.

2 Determine whether debugging for Sun Cluster HA for MySQL is active.

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

If debugging is inactive, `daemon.notice` is set in the file `/etc/syslog.conf` of the appropriate zone.

3 If debugging is inactive, edit the `/etc/syslog.conf` file in the appropriate zone to change `daemon.notice` to `daemon.debug`.

4 Confirm that debugging for Sun Cluster HA for MySQL is active.

If debugging is active, `daemon.debug` is set in the file `/etc/syslog.conf`.

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

5 Restart the `syslogd` daemon in the appropriate zone.

- If your operating system is Solaris 9, type:

```
# pkill -1 syslogd
```

- If your operating system is Solaris 10, type:

```
# svcadm refresh svc:/system/system-log:default
```

6 Edit the `/opt/SUNWsczone/sczbt/etc/config` file to change the `DEBUG=` variable according to one of the examples:

- `DEBUG=ALL`
- `DEBUG=resource name`
- `DEBUG=resource name,resource name, ...`

```
# cat /opt/SUNWscmys/etc/config
#
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# Usage:
#     DEBUG=<RESOURCE_NAME> or ALL
#
DEBUG=ALL
#
```

Note – To deactivate debugging, repeat step 1 to 6, changing `daemon.debug` to `daemon.notice` and changing the `DEBUG` variable to `DEBUG=`.

Upgrade to Sun Cluster 3.2 when Using Sun Cluster HA for MySQL

Use the information in this section to understand how to upgrade to Sun Cluster 3.2 when using Sun Cluster HA for MySQL.

Note – This procedure will not describe how to upgrade to Sun Cluster 3.2. It includes only the steps to upgrade Sun Cluster HA for MySQL to Sun Cluster 3.2.

▼ Upgrade from Sun Cluster 3.0 to Sun Cluster 3.2 when Using Sun Cluster HA for MySQL

Use this procedure to upgrade Sun Cluster HA for MySQL from Sun Cluster 3.0 to Sun Cluster 3.2.

1 Shut down Sun Cluster HA for MySQL resource.

```
# clresource disableMySQL-resource
```

2 Upgrade the nodes to Sun Cluster 3.2 according to Sun Cluster documentation.**3 Start the MySQL Server manually on Sun Cluster 3.2.**

```
#cd <MySQL Base directory>
```

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

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

```
#<MySQL Base directory>
/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
```

Copy the `mysql_config` to *myplace* and 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=

# Specify the physical hostname for the physical NIC that this
logical hostname

# belongs to for every node in the cluster this Resource group can be
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-hme0 and
# for hme3 on phys-2 it is phys-2-hme3.

# IE: MYSQL_NIC_HOSTNAME="phys-1-hme0 phys-2-hme3"

MYSQL_NIC_HOSTNAME=

# Where are your databases installed? (location of my.cnf)

MYSQL_DATADIR=
```

Note – The following is an example for a MySQL instance on Sun Cluster 3.2.

```
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"
MYSQL_DATADIR=/global/mysql-data
```

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

```
# ./mysql_register -f myplace/mysql_config
```

10 Stop the MySQL Server manually.

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

11 Start up Sun Cluster HA for MySQL resource.

```
# clresource enable MySQL-resource
```

12 Change source addresses for admin user 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 Sun Cluster 3.0 12/01 has been used, delete the root user belonging to the logical host and add an admin user belonging to the physical host.

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

The following is an example for a MySQL instance on Sun Cluster 3.2.

```
# 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 the 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

Use this procedure to upgrade from MySQL 3.23.54 to MySQL 4.x.x

- 1 Shut down the Sun Cluster HA for MySQL resource.**
`# clresource disable MySQL-resource`
- 2 Install the new MySQL binaries, follow Step 3 and 4 in Section How to Install and Configure MySQL.**
- 3 Start up Sun Cluster HA for MySQL resource.**
`# clresource enable MySQL-resource`
- 4 Shut down Sun Cluster HA for MySQL fault monitor.**
`# clresource unmonitor MySQL-resource`
- 5 Follow the MySQL Documentation to upgrade MySQL Database.**
- 6 Start up Sun Cluster HA for MySQL fault monitor.**
`# clresource monitor MySQL-resource`



Deployment Example: Installing MySQL in the Global Zone

This appendix presents a complete example of how to install and configure the MySQL application and data service in the global zone. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of MySQL installation in a non-global zone, see [Deployment Example: Installing MySQL in the Non-Global HA Container](#) or [Deployment Example: Installing MySQL in a Non-Global Zone](#) according to your zone type.

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- phys - schost - 1 (a physical node, which owns the file system)
- phys - schost - 2 (a physical node)

This configuration also uses the logical host name ha - host - 1.

Software Configuration

This deployment example uses the following software products and versions:

- Solaris 10 6/06 software for SPARC or x86 platforms
- Sun Cluster 3.2 core software
- Sun Cluster HA for MySQL
- MySQL max version 5.0.22
- Your preferred text editor

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Note – The steps for installing MySQL in a cluster that runs on Solaris version 9 OS are identical to the steps in this example.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for your preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures as superuser or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in the Global Zone

The tasks you must perform to install and configure MySQL in the global zone are as follows:

- “[Example: Preparing the Cluster for MySQL](#)” on page 62
- “[Example: Configuring Cluster Resources for MySQL](#)” on page 63
- “[Example: Installing and bootstrapping the MySQL Software on Local Storage](#)” on page 63
- “[Example: Modifying the MySQL Configuration Files](#)” on page 65
- “[Example: Enabling the MySQL Software to Run in the Cluster](#)” on page 66

▼ Example: Preparing the Cluster for MySQL

- 1 **Install and configure the cluster as instructed in *Sun Cluster Software Installation Guide for Solaris OS*.**

Install the following cluster software components on both nodes.

- Sun Cluster core software
- Sun Cluster data service for MySQL

- 2 **Beginning on the node that owns the file system, add the `mysql` user.**

```
phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
phys-schost-2# groupadd -g 1000 mysql
phys-schost-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

▼ Example: Configuring Cluster Resources for MySQL

- 1 Register the necessary data types on one of the nodes.

```
phys-schost-1# clresource type register SUNW.gds SUNW.HASStoragePlus
```

- 2 Create the MySQL resource group.

```
phys-schost-1# clresourcegroup create RG-MYS
```

- 3 Create the logical host.

```
phys-schost-1# clreslogicalhostname create -g RG-MYS ha-host-1
```

- 4 Create the HASStoragePlus resource in the RG-MYS resource group.

```
phys-schost-1# clresource create -g RG-MYS -t SUNW.HASStoragePlus -p AffinityOn=TRUE \
-p FilesystemMountPoints=/global/mnt3,/global/mnt4 RS-MYS-HAS
```

- 5 Enable the resource group.

```
phys-schost-1# clresourcegroup online -M RG-MYS
```

▼ Example: Installing and bootstrapping the MySQL Software on Local Storage

These steps illustrate how to install the MySQL software in the default directory `/usr/local/mysql`. As long as only one node is mentioned it needs to be the node where your resource group is online.

- 1 Install the MySQL binaries on both nodes.

```
phys-schost-1# cd /usr/local
phys-schost-1# tar xvf mysql-max-5.0.22-solaris10-architecture_64.tar.gz
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-architecture_64 mysql
```

- 2 Change the ownership of the MySQL binaries on both nodes.

```
phys-schost-1# chown -R mysql:mysql /usr/local/mysql
```

- 3 Create your data base directories.

```
phys-schost-1# mkdir -p /global/mnt3/mysql-data/logs
phys-schost-1# mkdir /global/mnt3/mysql-data/innodb
phys-schost-1# mkdir /global/mnt3/mysql-data/BDB
```

- 4 Bootstrap MySQL.

```
phys-schost-1# cd /usr/local/mysql
phys-schost-1# ./scripts/* --datadir=/global/mnt3/mysql-data
```

5 Create your my.cnf config-file in /global/mnt3/mysql-data

```
phys-schost-1# cat > /global/mnt3/mysql-data/my.cnf << EOF
[mysqld]
server-id=1
#port=3306
# 10.18.5.1 is the address of the logical host
bind-address=10.18.5.1 # this is the address of the logical host
socket=/tmp/hahostix1.sock
log=/global/mnt3/mysql-data/logs/log1
log-bin=/global/mnt3/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/global/mnt3/mysql-data/logs/log-slow-queries
#log-update=/global/mnt3/mysql-data/logs/log-update

# InnoDB
#skip-innodb
innodb_data_home_dir = /global/mnt3/mysql-data/innodb
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /global/mnt3/mysql-data/innodb
innodb_log_arch_dir = /global/mnt3/mysql-data/innodb
# You can set ..buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
set-variable = innodb_buffer_pool_size=50M
set-variable = innodb_additional_mem_pool_size=20M
# Set ..log_file_size to 25 % of buffer pool size
set-variable = innodb_log_file_size=12M
set-variable = innodb_log_buffer_size=4M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50

# BDB
# uncomment the skip-bdb if you used a binary download.
# binary downloads come very often without the bdb support.
#skip-bdb
bdb-home=/global/mnt3/mysql-data
bdb-no-recover
bdb-lock-detect=DEFAULT
bdb-logdir=/global/mnt3/mysql-data/BDB
bdb-tmpdir=/global/mnt3/mysql-data/BDB
#bdb_max_lock=10000

# Repliking Slave
#server-id=2
#master-host=administerix
#master-user=repl
#master-password=repl
#master-info-file=/global/mnt3/mysql-data/logs/master.info
```

```
# MySQL 4.x
#relay-log=/global/mnt3/mysql-data/logs/slave-bin.log
#relay-log-info-file=/global/mnt3/mysql-data/logs/slave-info
```

6 Change the ownership of the MySQL data directory.

```
phys-schost-1# chown -R mysql:mysql /global/mnt3/mysql-data
```

7 Change the permission of the my.cnf file.

```
phys-schost-1# chmod 644 /global/mnt3/mysql-data/my.cnf
```

▼ Example: Modifying the MySQL Configuration Files

1 Copy the MySQL configuration file from the agent directory to its deployment location.

```
phys-schost-1# cp /opt/SUNWscmys/util/mysql_config /global/mnt3
phys-schost-1# cp /opt/SUNWscmys/util/ha_mysql_config /global/mnt3
```

2 Add this cluster's information to the mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
.
.
.
MYSQL_BASE=/usr/local/mysql
MYSQL_USER=root
MYSQL_PASSWD=root
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCKET=/tmp/ha-host-1.sock
MYSQL_NIC_HOSTNAME="phys-schost-1 phys-schost-2"
MYSQL_DATADIR=/global/mnt3/mysql-data
```

3 Add this cluster's information to the ha_mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
.
.
.
RS=RS-PGS
RG=RG-MYS
PORT=5432
LH=hahostix1
HAS_RS=RS-MYS-HAS
.
```

```

.
.
BASEDIR=/usr/local/mysql
DATADIR=/global/mnt3/mysql-data
MYSQLUSER=mysql
MYSQLHOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES

```

- 4 Save and close the file.

▼ Example: Enabling the MySQL Software to Run in the Cluster

- 1 Start the MySQL database manually on the node where the resource group is online.

```

phys-schost-1# cd /usr/local/mysql
phys-schost-1# ./bin/mysqld --defaults-file=/global/mnt3/mysql-data/my.cnf \
--basedir=/usr/local/mysql --datadir=/global/mnt3/mysql-data \
--pid-file=/global/mnt3/mysql-data/mysqld.pid \
--user=mysql >> /global/mnt3/mysql-data/logs/ha-host-1.log 2>&1 &

```

- 2 Set the password for localhost in MySQL to root.

```

phys-schost-1# /usr/local/mysql/bin/mysqladmin -S /tmp/ha-host-1.sock -uroot \
password 'root'

```

- 3 Add an admin user in the MySQL database for the logical host.

```

phys-schost-1# /usr/local/mysql/bin/mysql -S /tmp/ha-host-1.sock -uroot -proot
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'phys-schost-1' IDENTIFIED BY 'root';
mysql> GRANT ALL ON *.* TO 'root'@'phys-schost-2' IDENTIFIED BY 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-1';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='phys-schost-2';
mysql> exit

```

- 4 Prepare the Sun Cluster specific test database.

```

phys-schost-1# ksh /opt/SUNWscmys/util/mysql_register -f /global/mnt3/mysql_config

```

- 5 Stop the MySQL database.

```

phys-schost-1# kill -TERM `cat /global/mnt3/mysql-data/mysqld.pid`

```

6 Run the `ha_mysql_register` script to register the resource.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \  
-f /global/mnt3/ha_mysql_config
```

7 Enable the resource.

```
phys-schost-1# clresource enable RS-MYS
```




Deployment Example: Installing MySQL in the Non-Global HA Container

This appendix presents a complete example of how to install and configure the MySQL application and data service in a non-global high-availability container. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of MySQL installation in a global zone, see [Deployment Example: Installing MySQL in the Global Zone](#). For an installation in a zone, see [Deployment Example: Installing MySQL in a Non-Global Zone](#).

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- phys - schost - 1 (a physical node, which owns the file system)
- phys - schost - 2 (a physical node)
- clu1 the zone to be failed over

This configuration also uses the logical host name ha - host - 1.

Software Configuration

This deployment example uses the following software products and versions:

- Solaris 10 6/06 software for SPARC or x86 platforms
- Sun Cluster 3.2 core software
- Sun Cluster HA for MySQL
- Sun Cluster HA for Solaris Container
- MySQL max version 5.0.22
- Your preferred text editor

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for your preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures as superuser or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in the HA Container

The tasks you must perform to install and configure MySQL in the HA container are as follows:

- [“Example: Preparing the Cluster for MySQL” on page 70](#)
- [“Example: Configuring Cluster Resources for MySQL” on page 71](#)
- [“Example: Configuring the HA Container” on page 71](#)
- [“Example: Installing and bootstrapping the MySQL Software on Local Storage” on page 73](#)
- [“Example: Modifying the MySQL Configuration Files” on page 75](#)
- [“Example: Enabling the MySQL Software to Run in the Cluster” on page 75](#)

▼ Example: Preparing the Cluster for MySQL

- 1 **Install and configure the cluster as instructed in *Sun Cluster Software Installation Guide for Solaris OS*.**

Install the following cluster software components on both nodes.

- Sun Cluster core software
- Sun Cluster data service for MySQL
- Sun Cluster data service for Solaris Containers

- 2 **Beginning on the node that owns the file system, add the `mysql` user.**

```
phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
phys-schost-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

▼ Example: Configuring Cluster Resources for MySQL

- 1 Register the necessary data types on one of the nodes.

```
phys-schost-1# clresourcetype register SUNW.gds SUNW.HAStoragePlus
```

- 2 Create the MySQL resource group.

```
phys-schost-1# clresourcegroup create RG-MYS
```

- 3 Create the HAStoragePlus resource in the RG-MYS resource group.

```
phys-schost-1# clresource create -g RG-MYS -t SUNW.HAStoragePlus -p AffinityOn=TRUE \
-p FilesystemMountPoints=/global/mnt3,/global/mnt4 RS-MYS-HAS
```

- 4 Enable the resource group.

```
phys-schost-1# clresourcegroup online -M RG-MYS
```

▼ Example: Configuring the HA Container

- 1 On shared cluster storage, create a directory for the HA container root path.

This example presents a sparse root zone. You can use a whole root zone if that type better suits your configuration.

```
phys-schost-1# mkdir /global/mnt3/zones
```

- 2 Create a temporary file, for example /tmp/x, and include the following entries:

```
create -b
set zonepath=/global/mnt3/zones/clu1
set autoboot=false
set pool=pool_default
add inherit-pkg-dir
set dir=/lib
end
add inherit-pkg-dir
set dir=/platform
end
add inherit-pkg-dir
set dir=/sbin
end
add inherit-pkg-dir
set dir=/usr
end
add net
set address=ha-host-1
set physical=hme0
end
```

```
add attr
set name=comment
set type=string
set value="MySQL cluster zone"      Put your desired zone name between the quotes here.
end
```

3 Configure the HA container, using the file you created.

```
phys-schost-1# zonecfg -z clu1 -f /tmp/x
```

4 Install the zone.

```
phys-schost-1# zoneadm -z clu1 install
```

5 Log in to the zone.

```
phys-schost-1# zlogin -C clu1
```

6 Open a new window to the same node and boot the zone?

```
phys-schost-1a# zoneadm -z clu1 boot
```

7 Close this terminal window and disconnect from the zone console.

```
phys-schost-1# ~.
```

8 Copy the containers configuration file to a temporary location.

```
phys-schost-1# cp /opt/SUNWsczone/sczbt/util/sczbt_config /tmp/sczbt_config
```

9 Edit the /tmp/sczbt_config file and set variable values as shown:

```
RS=RS-MYS-ZONE
RG=RG-MYS
PARAMETERDIR=/global/mnt3/zonepar
SC_NETWORK=false
SC_LH=
FAILOVER=true
HAS_RS=RS-MYS-HAS
```

```
Zonename=clu1
Zonebootopt=
Milestone=multi-user-server
Mounts=
```

Note – For more information about creating a zone, see the *Sun Cluster Data Service for Solaris Containers Guide*

10 Register the zone resource.

```
phys-schost-1# ksh /opt/SUNWsczone/sczbt/util/sczbt_register -f /tmp/sczbt_config
```

11 Enable the zone resource.

```
phys-schost-1# clresource enable RS-MYS-ZONE
```

▼ Example: Installing and bootstrapping the MySQL Software on Local Storage

These steps illustrate how to install the MySQL software in the default directory `/usr/local/MySQL`. As long as only one node is mentioned it needs to be the node where your resource group is online. It is assumed, that you inherited `/usr`, so you can not write to `/usr/local` in the zones. If you linked `/usr/local` to a local directory, start at step 4

1 Add the mysql user.

```
phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

2 Install the MySQL binaries on both nodes.

```
phys-schost-1# cd /usr/local
phys-schost-1# tar xvf mysql-max-5.0.22-solaris10-architecture_64.tar.gz
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-x86_64 mysql
```

3 Change the ownership of the MySQL binaries on both nodes.

```
phys-schost-1# chown -R mysql:mysql /usr/local/mysql
```

4 Log in to the zone.

```
phys-schost-1# zlogin clu1
```

5 Create the parent for the mysql home directory.

```
zone# mkdir -p /global/mnt3
```

6 Add the mysql user.

```
zone# groupadd -g 1000 mysql
zone# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

7 Create your data base directories.

```
zone# mkdir -p /global/mnt3/mysql-data/logs
zone# mkdir /global/mnt3/mysql-data/innodb
zone# mkdir /global/mnt3/mysql-data/BDB
```

8 Bootstrap MySQL.

```
zone# cd /usr/local/mysql
zone# ./scripts/* --datadir=/global/mnt3/mysql-data
```

9 Create your my.cnf config-file in /global/mnt3/mysql-data

```

zone# cat > /global/mnt3/mysql-data/my.cnf << EOF
[mysqld]
server-id=1
#port=3306
# 10.18.5.1 is the address of the logical host
bind-address=10.18.5.1 # this is the address of the logical host
socket=/tmp/ha-host-1.sock
log=/global/mnt3/mysql-data/logs/log1
log-bin=/global/mnt3/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/global/mnt3/mysql-data/logs/log-slow-queries
#log-update=/global/mnt3/mysql-data/logs/log-update

# InnoDB
#skip-innodb
innodb_data_home_dir = /global/mnt3/mysql-data/innodb
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /global/mnt3/mysql-data/innodb
innodb_log_arch_dir = /global/mnt3/mysql-data/innodb
# You can set ..buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
set-variable = innodb_buffer_pool_size=50M
set-variable = innodb_additional_mem_pool_size=20M
# Set ..log_file_size to 25 % of buffer pool size
set-variable = innodb_log_file_size=12M
set-variable = innodb_log_buffer_size=4M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50

# BDB
# uncomment the skip-bdb if you used a binary download.
# binary downloads come very often without the bdb support.
#skip-bdb
bdb-home=/global/mnt3/mysql-data
bdb-no-recover
bdb-lock-detect=DEFAULT
bdb-logdir=/global/mnt3/mysql-data/BDB
bdb-tmpdir=/global/mnt3/mysql-data/BDB
#bdb_max_lock=10000

# Repliking Slave
#server-id=2
#master-host=administerix
#master-user=repl
#master-password=repl
#master-info-file=/global/mnt3/mysql-data/logs/master.info

```

```
# MySQL 4.x
#relay-log=/global/mnt3/mysql-data/logs/slave-bin.log
#relay-log-info-file=/global/mnt3/mysql-data/logs/slave-info
```

10 Change the ownership of the MySQL data directory.

```
zone# chown -R mysql:mysql /global/mnt3/mysql-data
```

11 Change the permission of the my.cnf file.

```
zone# chmod 644 /global/mnt3/mysql-data/my.cnf
```

▼ Example: Modifying the MySQL Configuration Files

1 Copy the MySQL database configuration file from the agent directory to its deployment location.

```
phys-schost-1# cp /opt/SUNWscmys/util/mysql_config /config-files
```

2 Add this cluster's information to the mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
.
.
.
MYSQL_BASE=/usr/local/mysql
MYSQL_USER=root
MYSQL_PASSWD=root
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCKET=/tmp/ha-host-1.sock
MYSQL_NIC_HOSTNAME="ha-host-1"
MYSQL_DATADIR=/global/mnt3/mys-data
```

3 Save and close the file.

▼ Example: Enabling the MySQL Software to Run in the Cluster

1 Start the MySQL database manually on the node where the resource group is online.

```
zone# cd /usr/local/mysql
zone# ./bin/mysqld --defaults-file=/global/mnt3/mysql-data/my.cnf \
--basedir=/usr/local/mysql --datadir=/global/mnt3/mysql-data \
```

```
--pid-file=/global/mnt3/mysql-data/mysqld.pid \
--user=mysql >> /global/mnt3/mysql-data/logs/ha-host-1.log 2>&1 &
```

2 Set the password for localhost in MySQL to root.

```
zone# /usr/local/mysql/bin/mysqladmin -S /tmp/ha-host-1.sock -uroot \
password 'root'
```

3 Add an admin user in the MySQL database for the logical host.

```
zone# /usr/local/mysql/bin/mysql -S /tmp/ha-host-1.sock -uroot -proot
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'ha-host-1' IDENTIFIED BY 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='ha-host-1';
mysql> exit
```

4 Prepare the Sun Cluster specific test database.

```
zone# ksh /opt/SUNWscmys/util/mysql_register -f /config-files/mysql_config
```

5 Stop the MySQL database.

```
zone# kill -TERM 'cat /global/mnt3/mysql-data/mysqld.pid'
```

6 Leave the zone

7 Copy the MySQL database configuration file from the agent directory to its deployment location.

```
phys-schost-1# cp /opt/SUNWscmys/util/ha_mysql_config /global/mnt3
```

8 Add this cluster's information to the ha_mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
.
.
.
RS=RS-MYS
RG=RG-MYS
PORT=5432
LH=ha-host-1
HAS_RS=RS-MYS-HAS
.
.
.
ZONE=c1u1
ZONE_BT=RS-MYS-ZONE
PROJECT=.
.
.
BASEDIR=/usr/local/mysql
```

```
DATADIR=/global/mnt3/mysql-data
MYSQLUSER=mysql
MYSQLHOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES
```

9 Save and close the file.

10 Run the `ha_mysql_register` script to register the resource.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \
-f /global/mnt3/ha_mysql_config
```

11 Enable the resource.

```
phys-schost-1# clresource enable RS-MYS
```




Deployment Example: Installing MySQL in a Non-Global Zone

This appendix presents a complete example of how to install and configure the MySQL application and data service in a non-global zone. It presents a simple two-node cluster configuration. If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of MySQL in the global zone, see [Deployment Example: Installing MySQL in the Global Zone](#). For installation in an HA container, see [Deployment Example: Installing MySQL in the Non-Global HA Container](#).

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- phys - schost - 1 (a physical node, which owns the file system)
- zone - 1 (a zone defined on phys - schost - 1, which owns the file system)
- phys - schost - 2 (a physical node)
- zone - 2 (a physical node)

Software Configuration

This deployment example uses the following software products and versions:

- Solaris 10 6/06 software for SPARC or x86 platforms
- Sun Cluster 3.2 core software
- Sun Cluster Data Service for PostgreSQL
- MySQL version 5.0.22 tarball

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for your preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures as superuser or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring MySQL on Local Storage in a Non-Global Zone

These instructions assume that you are installing the MySQL software as the `mysql` user in a local directory.

The tasks you must perform to install and configure MySQL in the zone are as follows:

- [“Example: Preparing the Cluster for MySQL” on page 80](#)
- [“Example: Configuring the Zone” on page 80](#)
- [“Example: Configuring Cluster Resources for MySQL” on page 82](#)
- [“Example: Installing the MySQL Software on Local Storage” on page 82](#)
- [“Example: Bootstrapping the MySQL Software on Local Storage” on page 83](#)
- [“Example: Modifying the MySQL Configuration File” on page 85](#)
- [“Example: Enabling the MySQL Software to Run in the Cluster” on page 85](#)

▼ Example: Preparing the Cluster for MySQL

- Install and configure the cluster as instructed in [Sun Cluster Software Installation Guide for Solaris OS](#).

Install the following cluster software components on both nodes.

- Sun Cluster core software
- Sun Cluster data service for MySQL

▼ Example: Configuring the Zone

In this task you will install the Solaris Container on `phys-schost-1` and `phys-schost-2`. Therefore perform this procedure on both hosts.

1 On local cluster storage of , create a directory for the zone root path.

This example presents a sparse root zone. You can use a whole root zone if that type better suits your configuration.

```
phys-schost-1# mkdir /zones
```

2 Create a temporary file, for example /tmp/x, and include the following entries:

```
create -b
set zonepath=/zones/clu1
set autoboot=true
set pool=pool_default
add inherit-pkg-dir
set dir=/lib
end
add inherit-pkg-dir
set dir=/platform
end
add inherit-pkg-dir
set dir=/sbin
end
add inherit-pkg-dir
set dir=/usr
end
add net
set address=zone-1      Choose a different address (zone-2) on the second node.
set physical=hme0
end
add attr
set name=comment
set type=string
set value="MySQL cluster zone"    Put your desired zone name between the quotes here.
end
```

3 Configure the HA container, using the file you created.

```
phys-schost-1# zonecfg -z clu1 -f /tmp/x
```

4 Install the zone.

```
phys-schost-1# zoneadm -z clu1 install
```

5 Log in to the zone.

```
phys-schost-1# zlogin -C clu1
```

6 Open a new window to the same node and boot the zone?

```
phys-schost-1# zoneadm -z clu1 boot
```

- 7 **Close this terminal window and disconnect from the zone console.**

```
phys-schost-1# ~.
```

▼ **Example: Configuring Cluster Resources for MySQL**

- 1 **Register the HAStoragePlus resource type.**

```
phys-schost-1# clresource type register SUNW.gds SUNW.HAStoragePlus
```

- 2 **Create the MySQL resource group.**

```
phys-schost-1# clresourcegroup create -n phys-host-1:clu1,phys-host-2:clu1 RG-MYS
```

- 3 **Create the HAStoragePlus resource in the RG-MYS resource group.**

```
phys-schost-1# clresource create -g RG-MYS -t SUNW.HAStoragePlus -p AffinityOn=TRUE \
-p FilesystemMountPoints=/global/mnt3,/global/mnt4 RS-MYS-HAS
```

- 4 **Enable the resource group.**

```
phys-schost-1# clresourcegroup online -M RG-MYS
```

▼ **Example: Installing the MySQL Software on Local Storage**

These steps illustrate how to install the MySQL software in the default directory `/usr/local/mysql`. As long as only one node is mentioned it needs to be the node where your resource group is online. It is assumed, that you inherited `/usr`, so you can not write to `/usr/local` in the zones. If you linked `/usr/local` to a local directory, start at step 4.

- 1 **Add the mysql user.**

```
phys-schost-1# groupadd -g 1000 mysql
phys-schost-1# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
phys-schost-2# groupadd -g 1000 mysql
phys-schost-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

- 2 **Install the MySQL binaries on both nodes.**

```
phys-schost-1# cd /usr/local
phys-schost-1# tar xvf mysql-max-5.0.22-solaris10-architecture_64.tar.gz
phys-schost-1# ln -s mysql-max-5.0.22-solaris10-architecture_64 mysql
```

- 3 **Change the ownership of the MySQL binaries on both nodes.**

```
phys-schost-1# chown -R mysql:mysql /usr/local/mysql
```

4 Log in to the zone.

```
phys-schost-1# zlogin clu1
phys-schost-2# zlogin clu1
```

5 Add the mysql group and user.

```
zone-1# groupadd -g 1000 mysql
zone-1# groupadd -g 1000 mysql
zone-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
zone-2# useradd -g 1000 -d /global/mnt3/mysql -m -s /bin/ksh mysql
```

6 Leave the zone.

▼ Example: Bootstrapping the MySQL Software on Local Storage

These steps illustrate how to bootstrapp the MySQL software in the default directory /usr/local/pgsql. As long as only one node is mentioned it needs to be the node where your resource group is online.

1 Log in to the zone

```
phys-schost-1# zlogin clu1
```

2 Create your data base directories.

```
zone-1# mkdir -p /global/mnt3/mysql-data/logs
zone-1# mkdir /global/mnt3/mysql-data/innoDB
zone-1# mkdir /global/mnt3/mysql-data/BDB
```

3 Bootstrap MySQL.

```
zone-1# cd /usr/local/mysql
zone-1# ./scripts/* --datadir=/global/mnt3/mysql-dat
```

4 Create your my.cnf config-file in /global/mnt3/mysql-data.

```
zone-1# cat > /global/mnt3/mysql-data/my.cnf << EOF
[mysqld]
server-id=1
#port=3306
# 10.18.5.1 is the address of the logical host
bind-address=10.18.5.1 # this is the address of the logical host
socket=/tmp/ha-host-1.sock
log=/global/mnt3/mysql-data/logs/log1
log-bin=/global/mnt3/mysql-data/logs/bin-log
binlog-ignore-db=sc3_test_database
log-slow-queries=/global/mnt3/mysql-data/logs/log-slow-queries
```

```
#log-update=/global/mnt3/mysql-data/logs/log-update

# InnoDB
#skip-innodb
innodb_data_home_dir = /global/mnt3/mysql-data/innodb
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /global/mnt3/mysql-data/innodb
innodb_log_arch_dir = /global/mnt3/mysql-data/innodb
# You can set ..buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
set-variable = innodb_buffer_pool_size=50M
set-variable = innodb_additional_mem_pool_size=20M
# Set ..log_file_size to 25 % of buffer pool size
set-variable = innodb_log_file_size=12M
set-variable = innodb_log_buffer_size=4M
innodb_flush_log_at_trx_commit=1
set-variable = innodb_lock_wait_timeout=50

# BDB
# uncomment the skip-bdb if you used a binary download.
# binary downloads come very often without the bdb support.
#skip-bdb
bdb-home=/global/mnt3/mysql-data
bdb-no-recover
bdb-lock-detect=DEFAULT
bdb-logdir=/global/mnt3/mysql-data/BDB
bdb-tmpdir=/global/mnt3/mysql-data/BDB
#bdb_max_lock=10000

# Repliking Slave
#server-id=2
#master-host=administerix
#master-user=repl
#master-password=repl
#master-info-file=/global/mnt3/mysql-data/logs/master.info

# MySQL 4.x
#relay-log=/global/mnt3/mysql-data/logs/slave-bin.log
#relay-log-info-file=/global/mnt3/mysql-data/logs/slave-info
```

5 Change the ownership of the MySQL data directory.

```
zone-1# chown -R mysql:mysql /global/mnt3/mysql-data
```

6 Change the permission of the my.cnf file.

```
zone-1# chmod 644 /global/mnt3/mysql-data/my.cnf
```

▼ Example: Modifying the MySQL Configuration File

In this task you will create the configuration file to prepare the MySQL database. It is assumed, that you are still logged in to this zone.

- 1 **Copy the MySQL database configuration file from the agent directory to its deployment location.**

```
zone-1# cp /opt/SUNWscmys/util/mysql_config /config-files
```

- 2 **Add this cluster's information to the mysql_config configuration file.**

The following listing shows the relevant file entries and the values to assign to each entry.

```
.
.
.
MYSQL_BASE=/usr/local/mysql
MYSQL_USER=root
MYSQL_PASSWD=root
MYSQL_HOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
MYSQL_SOCKET=/tmp/ha-host-1.sock
MYSQL_NIC_HOSTNAME="zone-1 zone-2"
MYSQL_DATADIR=/global/mnt3/mys-data
```

- 3 **Save and close the file.**

▼ Example: Enabling the MySQL Software to Run in the Cluster

This task will initialize and prepare your database, it is essential, that you perform it on one node only. It is assumed, that you are still logged in to this zone.

- 1 **Start the MySQL database manually on the zone where the resource group is online.**

```
zone-1# cd /usr/local/mysql
zone-1# ./bin/mysqld --defaults-file=/global/mnt3/mysql-data/my.cnf \
--basedir=/usr/local/mysql --datadir=/global/mnt3/mysql-data \
--pid-file=/global/mnt3/mysql-data/mysqld.pid \
--user=mysql >> /global/mnt3/mysql-data/logs/ha-host-1.log 2>&1 &
```

- 2 **Set the password for localhost in MySQL to root.**

```
zone-1# /usr/local/mysql/bin/mysqladmin -S /tmp/ha-host-1.sock -uroot \
password 'root'
```

3 Add an admin user in the MySQL database for the logical host.

```
zone-1# /usr/local/mysql/bin/mysql -S /tmp/ha-host-1.sock -uroot -proot
mysql> use mysql;
mysql> GRANT ALL ON *.* TO 'root'@'zone-1' IDENTIFIED BY 'root';
mysql> GRANT ALL ON *.* TO 'root'@'zone-2' IDENTIFIED BY 'root';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone-1';
mysql> UPDATE user SET Grant_priv='Y' WHERE User='root' AND Host='zone-2';
mysql> exit
```

4 Prepare the Sun Cluster specific test database.

```
zone-1# ksh /opt/SUNWscmys/util/mysql_register -f /global/mnt3/mysql_config
```

5 Stop the MySQL database.

```
zone-1# kill -TERM `cat /global/mnt3/mysql-data/mysqld.pid`
```

6 Leave the zone.**7 Copy the MySQL database configuration file from the agent directory to its deployment location.**

```
phys-schost-1# cp /opt/SUNWscmys/util/ha_mysql_config /global/mnt3
```

8 Add this cluster's information to the ha_mysql_config configuration file.

The following listing shows the relevant file entries and the values to assign to each entry.

```
.
.
.
RS=RS-MYS
RG=RG-MYS
PORT=5432
LH=ha-host-1
HAS_RS=RS-MYS-HAS
.
.
.
BASEDIR=/usr/local/mysql
DATADIR=/global/mnt3/mysql-data
MYSQLUSER=mysql
MYSQLHOST=ha-host-1
FMUSER=fmuser
FMPASS=fmuser
LOGDIR=/global/mnt3/mys-data/logs
CHECK=YES
```

9 Save and close the file.

10 Run the `ha_mysql_register` script to register the resource.

```
phys-schost-1# ksh /opt/SUNWscmys/util/ha_mysql_register \  
-f /global/mnt3/ha_mysql_config
```

11 Enable the resource.

```
phys-schost-1# clresource enable RS-MYS
```


Index

A

Admin user, 37, 42, 47

B

Bootstrap, 26, 29, 33

C

clnode command, 10

Command

chmod, 26, 30, 34

chown, 25, 26, 28, 30, 33

clresource, 53

clresourcegroup, 24, 27

clresourcetype, 24, 27

cluster, 53

groupadd, 25, 28, 32

useradd, 25, 28, 32

commands

clresourcegroup, 31

node information, 10

Configuration

MySQL, 23-34

Registering Sun Cluster HA for MySQL, 37-53

Requirements, 18-22

Restrictions, 13-18

Verify, 34

D

deployment examples

MySQL in a local zone, 79

MySQL in a non-global zone, 79

MySQL in an HA container, 79

MySQL in the global zone, 61-67, 69-77

MySQL with Solaris 9 OS, 61-67, 69-77

E

enabling resource group, 31

examples

MySQL in a local zone, 79

MySQL in a non-global zone, 79

MySQL in an HA container, 79

MySQL in the global zone, 61-67, 69-77

MySQL with Solaris 9 OS, 61-67, 69-77

F

Fault Monitor

Probing Algorithm and Functionality, 54

Resource Properties, 54

Understanding, 54

G

global zone, 35

H

help, 10

I

Install, 23-34

installing, Sun Cluster HA for MySQL, 35-36

L

local zones, *See* non-global zones

M

messages file, 10

MySQL, 12

N

non-global zones, 35

P

packages, 35-36

prtconf -v command, 10

prtdiag -v command, 10

psrinfo -v command, 10

R

resource group, enabling, 31

Resource Types

 SUNW.gds, 24, 27

restrictions, zones, 35

S

show-rev subcommand, 10

showrev -p command, 10

software packages, 35-36

Sun Cluster HA for MySQL

 installing, 35-36

 software packages, installing, 35-36

T

technical support, 10

U

Upgrade

 Upgrade to MySQL 4.0.12 from 3.23.54, 60

 Upgrade to Sun Cluster 3.2 when using Sun Cluster
 HA for MySQL, 56-60

V

/var/adm/messages file, 10

Z

zones, 35