



# Sun Cluster 3.1 Data Service for MySQL ガイド

---

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

Part No: 817-4845-10  
2003 年10 月, Revision A

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

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

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

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリコービイマジクス株式会社からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、AnswerBook、AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンのロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政事業庁が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DiComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *Sun Cluster 3.1 Data Service for MySQL Guide*

Part No: 817-3242-10

Revision A



040412@8606



# 目次

---

<b>Sun Cluster HA for MySQL のインストールと構成</b>	5
Sun Cluster HA for MySQL のインストールと構成	5
Sun Cluster HA for MySQL の概要	6
Sun Cluster HA for MySQL のインストールと構成の計画	6
構成上の制限事項	7
構成の要件	11
MySQL のインストールと構成	15
MySQL のインストールと構成方法	15
MySQL のインストールと構成の確認	18
▼ MySQL のインストールと構成の確認方法	18
Sun Cluster HA for MySQL パッケージのインストール	19
Web Start プログラムを使用して Sun Cluster HA for MySQL パッケージをインストールする方法	19
▼ <code>scsintall</code> ユーティリティを使用して Sun Cluster HA for MySQL パッケージをインストールする方法	20
Sun Cluster HA for MySQL の登録と構成	21
▼ フェイルオーバーサービスとしての Sun Cluster HA for MySQL の登録と構成方法	21
Sun Cluster HA for MySQL のインストールと構成の確認	26
▼ Sun Cluster HA for MySQL のインストールと構成の確認方法	26
Sun Cluster HA for MySQL 障害モニターの概要	26
リソースプロパティ	27
検証アルゴリズムと機能	27
Sun Cluster HA for MySQL のデバッグ	27
▼ Sun Cluster HA for MySQL でデバッグを有効にする方法	27
Sun Cluster HA for MySQL 使用時に SC3.1 にアップグレードする	29

Sun Cluster HA for MySQL 使用時に MySQL を 3.23.54 から 4.0.12 にアップグレード  
する 32

索引 33

# Sun Cluster HA for MySQL のインストールと構成

---

---

## Sun Cluster HA for MySQL のインストールと構成

表 1-1 に、Sun Cluster HA for MySQL のインストールと構成に必要な作業を示します。指定された順番どおりに、各作業を行なってください。

表 1-1 作業マップ: Sun Cluster HA for MySQL のインストールと構成

作業	参照先
インストールを計画する	6 ページの「Sun Cluster HA for MySQL の概要」 6 ページの「Sun Cluster HA for MySQL のインストールと構成の計画」
MySQL をインストールして構成する	15 ページの「MySQL のインストールと構成方法」
インストールと構成を確認する	18 ページの「MySQL のインストールと構成の確認方法」
Sun Cluster HA for MySQL パッケージをインストールする	20 ページの「 <code>scsintall</code> ユーティリティを使用して Sun Cluster HA for MySQL パッケージをインストールする方法」
Sun Cluster HA for MySQL を登録して構成する	21 ページの「フェイルオーバーサービスとしての Sun Cluster HA for MySQL の登録と構成方法」
Sun Cluster HA for MySQL のインストールと構成を確認する	26 ページの「Sun Cluster HA for MySQL のインストールと構成の確認方法」
fmv1; の障害モニターを理解する	26 ページの「Sun Cluster HA for MySQL 障害モニターの概要」

表 1-1 作業マップ : Sun Cluster HA for MySQL のインストールと構成 (続き)

作業	参照先
Sun Cluster HA for MySQL をデバッグする	27 ページの「Sun Cluster HA for MySQL のデバッグ」
Sun Cluster HA for MySQL 使用時に SC3.1 にアップグレードする	29 ページの「Sun Cluster HA for MySQL 使用時に SC3.1 にアップグレードする」
Sun Cluster HA for MySQL 使用時に MySQL を 3.23.54 から 4.0.12 にアップグレードする	32 ページの「Sun Cluster HA for MySQL 使用時に MySQL を 3.23.54 から 4.0.12 にアップグレードする」

## Sun Cluster HA for MySQL の概要

MySQL ソフトウェアは、超高速で堅牢なマルチスレッドおよびマルチユーザー型の SQL (Structured Query Language) データベースサーバーを提供します。MySQL サーバーはミッションクリティカルで高負荷の実稼働システムを対象にしていると同時に、大量に配備されるソフトウェアへの組み込みも対象にしています。MySQL は MySQL AB™ の商標です。

MySQL は、GNU (GNU 一般公衆利用許諾契約条件) のもとで無料で利用でき、<http://www.mysql.com> からダウンロードできます。

Sun Cluster HA for MySQL データサービスのメカニズムによって、MySQL サービスの順序正しい起動、停止、障害モニター、自動フェイルオーバーを実行できます。次の MySQL コンポーネントは、Sun Cluster HA for MySQL データサービスによって保護されます。

表 1-2 コンポーネントの保護

コンポーネント	保護する対象
MySQL サーバー	Sun Cluster HA for MySQL

## Sun Cluster HA for MySQL のインストールと構成の計画

ここでは、Sun Cluster HA for MySQL のインストールと構成の計画について説明します。

## 構成上の制限事項



---

注意 – 次の制限事項を守らないと、データサービスの構成がサポートされない場合があります。

---

ここで示す制限事項を考慮して、Sun Cluster HA for MySQL のインストールと構成の計画を行ってください。ここでは、Sun Cluster HA for MySQL にのみ適用されるソフトウェアとハードウェア構成の制限事項を示します。

すべてのデータサービスに適用される制限事項については、『Sun Cluster ご使用にあたって』を参照してください。

- Sun Cluster for MySQL は HA (高可用性) としてのみ構成できます。Scalable (スケーラブル) としては構成できません。
- MySQL 構成ファイル (my.cnf) は MySQL データベースディレクトリに格納してください。my.cnf をローカルのファイルシステムに格納する必要がある場合は、MySQL データベースディレクトリからシンボリックリンクを作成します。コマンド行オプションが書き換えられる可能性があるため、/etc (広域ファイル) には格納しないでください。
- サポートされる構成 – 現在 Sun Cluster HA for MySQL データサービスがサポートする構成は次のとおりです。
  - マスター構成の単一または複数の MySQL インスタンス
  - スレーブ構成の単一または複数の MySQL インスタンス

---

注 – マスターとスレーブのインスタンスを物理的に同じノードに存在させないでください。

---

- クラスタファイルシステムへの **MySQL** のインストール – MySQL の配布方法 (<http://www.mysql.com> から配布する、またはほかのソースから配布する) に関係なく、次の制限事項が適用されます。
  - 各 MySQL インスタンスは一意のデータベースディレクトリを持っている必要があります。このデータベースディレクトリは、フェイルオーバーシステムまたは広域ファイルシステムとしてマウントできます。

---

注 – 広域ファイルシステムをマウントする場合は /global という接頭辞を指定し、フェイルオーバーファイルシステムをマウントする場合は /local という接頭辞を指定するようにしてください。必須ではありませんが、こうするのが最も良い方法とされています。

---

#### 例 1-1 MySQL インスタンス (広域ファイルシステムの場合)

次に、MySQL と 2 つの MySQL インスタンス (mysql-data-1 と mysql-data-2) の両方を広域ファイルシステムにインストールする例を示します。最終出力は、Veritas Volume Manager を使用して配備された MySQL に対応する /etc/vfstab エントリのサブセットです。

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

例 1-1 MySQL インスタンス (広域ファイルシステムの場合) (続き)

```

drwxr-xr-x  2 mysql  mysql      512 Apr 14 11:16 BDB
drwxr-xr-x  2 mysql  mysql      512 Apr 14 11:14 innodb
drwxr-xr-x  2 mysql  mysql     2560 Apr 15 10:15 logs
-rw-r--r--  1 mysql  mysql     1459 Apr 14 11:13 my.cnf
drwx----- 2 mysql  mysql      512 Apr 14 11:14 mysql
-rw-rw----  1 mysql  mysql        5 Apr 15 10:10 mysqld.pid
drwx----- 2 mysql  mysql      512 Apr 15 10:10 sc3_test_database
drwx----- 2 mysql  mysql      512 Apr 14 11:14 test
drwx----- 2 mysql  mysql      512 Apr 14 11:16 testdb

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

#

```

---

注 - この例では、1 番目の MySQL インスタンスのデータベースディレクトリは /global/mysql-data-1 であり、2 番目の MySQL インスタンスのデータベースディレクトリは /global/mysql-data-2 です。

---

例 1-2 MySQL インスタンス (フェイルオーバーファイルシステムの場合)

次に、MySQL を局所ファイルシステムに、2 つの MySQL インスタンス (mysql-data-1 と mysql-data-2) をフェイルオーバーファイルシステムにインストールする例を示します。最終出力は、Veritas Volume Manager を使用して配備された MySQL に対応する /etc/vfstab エントリのサブセットです。

```

# ls -l /usr/local/mysql
total 10432
drwxr-xr-x 13 mysql  mysql      512 Mar 16 00:03 .
drwxrwxrwx  7 root   other     2048 Apr 11 09:53 ..
-rw-r--r--  1 mysql  mysql     19106 Mar 15 23:29 COPYING
-rw-r--r--  1 mysql  mysql     28003 Mar 15 23:29 COPYING.LIB
-rw-r--r--  1 mysql  mysql    126466 Mar 15 16:47 ChangeLog
-rw-r--r--  1 mysql  mysql     6811 Mar 15 23:29 INSTALL-BINARY
-rw-r--r--  1 mysql  mysql     1937 Mar 15 16:47 README
drwxr-xr-x  2 mysql  mysql     1536 Mar 16 00:03 bin
-rwxr-xr-x  1 mysql  mysql      773 Mar 16 00:03 configure
drwxr-x---  4 mysql  mysql      512 Mar 16 00:03 data
drwxr-xr-x  2 mysql  mysql     1024 Mar 16 00:03 include
drwxr-xr-x  2 mysql  mysql      512 Mar 16 00:03 lib
drwxr-xr-x  3 mysql  mysql      512 Mar 16 00:03 man
-rw-r--r--  1 mysql  mysql    2676944 Mar 15 23:23 manual.html
-rw-r--r--  1 mysql  mysql    2329252 Mar 15 23:23 manual.txt

```

例 1-2 MySQL インスタンス (フェイルオーバーファイルシステムの場合) (続き)

```

-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
#

```

---

注 - この例では、1 番目の MySQL インスタンスのデータベースディレクトリは /local/mysql-data-1 であり、2 番目の MySQL インスタンスのデータベースディレクトリは /local/mysql-data-2 です。

---



例 1-3 Sun Cluster 用の MySQL 構成/登録ファイル (続き)

```
#
# This file will be sourced in by mysql_register and the parameters
# listed below will be used.
#
(日本語訳)
# このファイルは mysql_register によって有効化され、以下のパラメタ
# が使用されます。

# Where is mysql installed (BASEDIR)
(日本語訳)
# mysql のインストール場所 (BASEDIR)
MYSQL_BASE=

# Mysql admin-user for localhost
(日本語訳)
# ローカルホストの mysql 管理ユーザー
MYSQL_USER=

# Password for mysql admin user
(日本語訳)
# mysql 管理ユーザー用のパスワード
MYSQL_PASSWD=

# Configured logicalhost
(日本語訳)
# 構成済み論理ホスト
MYSQL_HOST=

# Specify a username for a faultmonitor user
(日本語訳)
# 障害モニターユーザーのユーザー名を指定する
FMUSER=

# Pick a password for that faultmonitor user
(日本語訳)
# 障害モニターユーザー用のパスワードを選ぶ
FMPASS=

# Socket name for mysqld ( Should be /tmp/<logical host>.sock )
(日本語訳)
# mysqld のソケット名 ( /tmp/<logical host>.sock )
MYSQL_SOCKET=/tmp/<logical host>.sock

# FOR SC3.1 ONLY, Specify the physical hostname for the
# physical NIC that this logicalhostname belongs to for every node in the
# cluster this Resourcegroup can located on.
# IE: The logicalhost lh1 belongs to hme1 for physical-node phys-1 and
# hme3 for physical-node phys-2. The hostname for hme1 is phys-1-hme1 and
# for hme3 on phys-2 it is phys-2-hme3.
# IE: MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
(日本語訳)
# SC3.1 に限り、物理 NIC には物理ホスト名を指定してください。
# SC3.1 の論理ホスト名は、このリソースグループを置くことができる
```

### 例 1-3 Sun Cluster 用の MySQL 構成/登録ファイル (続き)

```
# クラスタのすべてのノードに属します。
# 例 : 論理ホスト lh1 は、物理ノード phys-1 では hme1 に属し、
# 物理のノード phys-2 では hme3 に属します。hme1 のホスト名は phys-1-hme1、
# phys-2 の hme3 のホスト名は phys-2-hme3 です。
# 例 : MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
MYSQL_NIC_HOSTNAME=""
::::::::::::
util/ha_mysql_config
::::::::::::
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# This file will be sourced in by ha_mysql_register and the parameters
# listed below will be used.
(日本語訳)
# このファイルは ha_mysql_register によって有効化され、以下のパラメタ
# が使用されます。
#
# These parameters can be customized in (key=value) form
#
#     RS - name of the resource for the application
#     RG - name of the resource group containing RS
#     BASEDIR - name of the Mysql bin directory
#     DATADIR - name of the Mysql Data directory
#     MYSQLUSER - name of the user Mysql should be started of
#     LH - name of the LogicalHostname SC resource
#     FMUSER - name of the Mysql fault monitor user
#     FMPASS - name of the Mysql fault monitor user password
#     LOGDIR - name of the directory mysqld should store it's logfile.
#     CHECK - should HA-MySQL check MyISAM index files before start YES/NO.
#     HAS_RS - name of the mysql HAStoragePlus SC resource
#
#     The following examples illustrate sample parameters
#     for Mysql
#
#     BASEDIR=/usr/local/mysql
#     DATADIR=/global/mysqldata
#     MYSQLUSER=mysql
#     LH=mysql1lh
#     FMUSER=fmuser
#     FMPASS=fmuser
#     LOGDIR=/global/mysqldata/logs
#     CHECK=YES
#
(日本語訳)
# このパラメタは (キー=値) 形式でカスタマイズできます。
#
#     RS - アプリケーションのリソース名
#     RG - RS を含むリソースグループ名
#     BASEDIR - Mysql bin ディレクトリ名
#     DATADIR - Mysql Data ディレクトリ名
#     MYSQLUSER - Mysql の起動ユーザー名
```

例 1-3 Sun Cluster 用の MySQL 構成/登録ファイル (続き)

```
#      LH - LogicalHostname SC リソース名
#      FMUSER - Mysql 障害モニターユーザー名
#      FMPASS - Mysql 障害モニターユーザー用のパスワード名
#      LOGDIR - mysqld がログファイルを格納するディレクトリ名
#      CHECK - HA-MySQL が MyISAM のインデックスファイルをチェックしてからスタート
#              する、YES/NO
#      HAS_RS - mysql HAStoragePlus SC リソース名
#
#      以下に、Mysql のパラメタの例を示します。
#
#      BASEDIR=/usr/local/mysql
#      DATADIR=/global/mysqldata
#      MYSQLUSER=mysql
#      LH=mysql1h
#      FMUSER=fmuser
#      FMPASS=fmuser
#      LOGDIR=/global/mysqldata/logs
#      CHECK=YES
#
RS=
RG=
BASEDIR=
DATADIR=
MYSQLUSER=
LH=
FMUSER=
FMPASS=
LOGDIR=
CHECK=
HAS_RS=
#
```

- **my.cnf** ファイル— Sun Cluster HA for MySQL データサービスには my.cnf のサンプルファイルが 2 つ (マスター構成用とスレーブ構成用) 用意されています。少なくとも次のパラメータが設定されていることを確認する必要があります。

---

注—my.cnf ファイルは MySQL にとって重要なファイルです。したがって、MySQL のマニュアルを参照して、次のパラメータの設定に関する詳細を確認してください。

---

- マスター構成のセクション [mysqld] 内の MySQL ファイル my.cnf
  - **bind address** には定義済みの論理ホストを設定する必要があります。
  - **socket** は /tmp/<論理ホスト>.sock として定義する必要があります。
  - **binlog-ignore-db** は、log-bin オプションが使用されている場合は、sc3\_test\_database という文字列を含んでいる必要があります。
- スレーブ構成のセクション [mysqld] 内の MySQL ファイル my.cnf

- **bind address** には定義済みの論理ホストを設定する必要があります。
- **socket** は /tmp/<論理ホスト>.sock として定義する必要があります。
- **binlog-ignore-db** は、log-bin オプションが使用されている場合は、sc3\_test\_database を含んでいる必要があります。
- **master-host** は、マスターインスタンスが存在するホスト名です。
- **master-user** は、スレーブがマスターに対する識別に使用するユーザー名です。
- **master-password** は、スレーブがマスターに対する識別に使用するパスワードです。
- **master-info-file** は、複製プロセス中、マスター上のどの場所にいたかを覚えておくためのファイルの場所です。このファイルは、グローバルファイルシステムまたはフェイルオーバーファイルシステムに格納する必要があります。

---

## MySQL のインストールと構成

ここでは MySQL をインストールして構成する手順について説明します。

これ以降のセクションでは、MySQL の特定のディレクトリが参照されます。参照で 사용되는一般的なパス名は次のとおりです。これらのパス名は前のセクションで説明した例の中で使用されています。7 ページの「構成上の制限事項」を参照してください。

- MySQL が <http://www.mysql.com> から広域ファイルシステムにインストールされ、MySQL インスタンスが広域ファイルシステムにインストールされる場合 (例 1-1を参照)。
  - MySQL ベースディレクトリ — /global/mysql
  - MySQL データベースディレクトリ — /global/mysqlldata
- MySQL が局所ファイルシステムにインストールされ、MySQL インスタンスがフェイルオーバーファイルシステムにインストールされる場合 (例 1-2を参照)。
  - MySQL ベースディレクトリ — /usr/local/mysql
  - MySQL データベースディレクトリ — /local/mysqlldata

## MySQL のインストールと構成方法

次の手順で、MySQL をインストールして構成します。

1. **Sun Cluster** における **MySQL** の配備方法を決定 – MySQL をどのように配備するかを決定する必要があります。

- 配備する MySQL インスタンス数を決定します。
  - 各 MySQL インスタンスで使用するクラスタファイルシステムを決定します。
2. **MySQL** クラスタファイルシステムのマウント – Sun Cluster における MySQL の配備方法を決定してから、クラスタファイルシステムがマウントされていることを確認する必要があります。

---

注 – MySQL インスタンスにフェイルオーバーファイルシステムを使用させる場合は、手動でファイルシステムをマウントする必要があります。

---

3. **Sun Cluster** 内の全ノードへの **MySQL** のインストール – MySQL は広域ファイルシステムにインストールすることをお勧めします。ローカルファイルシステムにソフトウェアをインストールした場合とクラスタファイルシステムにインストールした場合の長所と短所については、『*Sun Cluster* データサービスのインストールと構成』の「アプリケーションバイナリの格納先の決定」を参照してください。

- <http://www.mysql.com> から **MySQL** をダウンロードします。— しかし、MySQL ソフトウェア用にローカルディスクを使用する予定の場合、この手順を Sun Cluster 内のすべてのノード上で繰り返す必要があります。

4. **MySQL** を実行するクラスタ内のすべてのノードで **MySQL** 用の **mysql-user** と **mysql-group** を作成します。

- Sun Cluster のすべてのノードで、`/etc/group` にエントリを作成します。  

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

Sun Cluster のすべてのノードで、`/etc/passwd` にエントリを作成します。このユーザーのパスワードはロックしておく必要があります。

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

5. **MySQL** バイナリ用の所有者とグループを変更します。

- **MySQL** バイナリがすべてのノードにある場合は、すべてのノードでこの手順を繰り返す必要があります。

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

6. **MySQL** インスタンス用の **MySQL** データベースディレクトリを作成します。

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

---

注 – `<MySQL Database directory>` については、7 ページの「構成上の制限事項」を参照してください。7 ページの「構成上の制限事項」の例と次の例で使用されている一般的なパス名の一覧については、15 ページの「MySQL のインストールと構成」を参照してください。

---

#### 例 1-4 MySQL インスタンス用のディレクトリ

次に、ある MySQL インスタンスの例を示します。MySQL は <http://www.mysql.com> から広域ファイルシステムとしてマウントされている /global/mysql にインストールされています。この MySQL インスタンス用の MySQL データベースディレクトリは /global/mysql-data です。

```
# 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
#
```

- 要件に応じた **MySQL** ファイル **my.cnf** を作成します。— Sun Cluster HA for MySQL データサービスには my.cnf のサンプルファイルが 2 つ (マスター構成用とスレーブ構成用) 用意されています。

---

注 – Sun Cluster の初回のインストール時に Sun Cluster HA for MySQL パッケージ (SUNWscmys) をインストールしなかった場合は、19 ページの「Sun Cluster HA for MySQL パッケージのインストール」を参照してパッケージを SC3.1 にインストールする必要があります。そのあと、MySQL のインストールと構成作業を続けてください。

---

/opt/SUNWscmys/etc/my.cnf\_sample\_[master|slave] はサンプルの MySQL 構成ファイルであり、このファイルを使用すると、独自の MySQL インスタンス用の *<MySQL Databasedirectory>* /my.cnf を作成できます。ただし、各自の構成値が反映されるように、ファイルを編集する必要があります。

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

8. **MySQL** インスタンスのブートストラップを実行します。— これによって、特権テーブル `db`、`host`、`user`、`tables_priv`、および `columns_priv` が `func` テーブルとともに **MySQL** データベース内に作成されます。

```
# cd <MySQL Basedirectory>

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

9. **<MySQL Database Directory>** 内にログファイルを作成します。

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

10. **<MySQL Database Directory>** 用の所有者とグループを変更します。

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

11. **<MySQL Database Directory>/my.cnf** ファイルのアクセス権を変更します。

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

---

## MySQL のインストールと構成の確認

ここでは、インストールと構成を確認する手順について説明します。

### ▼ MySQL のインストールと構成の確認方法

次の手順でインストールと構成を確認します。データサービスをまだインストールしていないため、この手順ではアプリケーションの可用性が高いかどうかを確認することはできません。

---

注 – MySQL のインストールと構成を確認する前に、MySQL に対応する論理ホスト名が利用できることを確認してください。論理ホスト名を利用するには、21 ページの「フェイルオーバーサービスとしての Sun Cluster HA for MySQL の登録と構成方法」の手順 2 から手順 7 までを完了する必要があります。

---

1. インスタンス用の **MySQL** サーバーを起動します。

```
#cd <MySQL Basedirectory>

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

2. MySQL インスタンスに接続します。

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

3. MySQL サーバーインスタンスを停止します。

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

---

## Sun Cluster HA for MySQL パッケージのインストール

Sun Cluster のインストール中に Sun Cluster HA for MySQL パッケージをインストールしなかった場合は、次の手順を実行してインストールします。次の手順は、Sun Cluster HA for MySQL パッケージをインストールするすべてのクラスタノードで実行します。この手順を実行するには、Sun Cluster Agents CD-ROM が必要です。

同時に複数のデータサービスをインストールする場合は、『Sun Cluster 3.1 10/03 ソフトウェアのインストール』の「ソフトウェアのインストール」に記載されている手順を実行します。

Sun Cluster HA for MySQL パッケージをインストールするには、次のインストールツールの 1 つを使用します。

- Web Start プログラム
- scinstall ユーティリティ

---

注 – Web Start プログラムは Sun Cluster 3.1 Data Services 10/03 より前のリリースでは利用できません。

---

## Web Start プログラムを使用して Sun Cluster HA for MySQL パッケージをインストールする方法

Web Start プログラムは、コマンド行インタフェース (CLI) またはグラフィカルユーザーインタフェース (GUI) のどちらでも実行できます。インストールの内容と順番は CLI でも GUI でもほとんど同じです。Web Start プログラムの詳細は、`installer (1M)` マニュアルページを参照してください。

1. **Sun Cluster HA for MySQL** パッケージをインストールするクラスタノード上で、スーパーユーザーになります。
2. (省略可能) **Web Start** プログラムを **GUI** で実行する場合は、**DISPLAY** 環境変数が設定されていることを確認します。

3. **Sun Cluster Agents CD-ROM** を **CD-ROM** ドライブに挿入します。  
ボリューム管理デーモン `vol1d(1M)` が実行されており、**CD-ROM** デバイスを管理するように構成されている場合、**CD-ROM** は自動的に `/cdrom/scdataservices_3_1_vb` ディレクトリにマウントされます。
4. **CD-ROM** 上の **Sun Cluster HA for MySQL** コンポーネントディレクトリに移動します。  
このディレクトリに、**Sun Cluster HA for MySQL** データサービス用の **Web Start** プログラムが格納されています。  

```
# cd /cdrom/scdataservices_3_1_vb/コンポーネント/SunCluster_HA_MySQL_3.1
```
5. **Web Start** プログラムを起動します。  

```
# ./installer
```
6. プロンプトが表示されたら、インストールのタイプを選択します。
  - C ロケールだけをインストールするには、「一般」を選択します。
  - ほかのロケールをインストールするには、「カスタム」を選択します。
7. 画面上の指示に従って、**Sun Cluster HA for MySQL** パッケージをノードにインストールします。  
インストール完了後、**Web Start** はインストールの概要を表示します。この概要を使用すると、**Web Start** プログラムがインストール中に作成したログを表示できます。これらのログは `/var/sadm/install/logs` ディレクトリにあります。
8. **Web Start** プログラムを終了します。
9. **Sun Cluster Agents CD-ROM** を **CD-ROM** ドライブから取り出します。
  - a. **CD-ROM** が使用されないように、**CD-ROM** 上にないディレクトリに移動します。
  - b. **CD-ROM** を取り出します。  

```
# eject cdrom
```

## ▼ `scsinstall` ユーティリティを使用して **Sun Cluster HA for MySQL** パッケージをインストールする方法

この手順では、`scinstall` ユーティリティを **SC3.1** 上で使用して **Sun Cluster HA for MySQL** パッケージをインストールします。この作業には、**Sun Cluster Agents CD-ROM** が必要です。この手順は、**Sun Cluster** の初期インストール時にデータサービスパッケージをインストールしていない場合を想定しています。

Sun Cluster のインストール時に Sun Cluster HA for MySQL パッケージをインストールした場合は、21 ページの「Sun Cluster HA for MySQL の登録と構成」へ進んでください。

それ以外の場合は、次の手順で、Sun Cluster HA for MySQL パッケージをインストールします。この作業は、Sun Cluster HA for MySQL データサービスを実行できるすべてのノードで行います。

1. **CD-ROM** ドライブに **Sun Cluster Agents CD-ROM** を挿入します。
2. **scinstall** ユーティリティーをオプションなしで実行します。  
scinstall ユーティリティーが対話型モードで起動します。
3. メニューオプション「新しいデータサービスのサポートをこのクラスタノードに追加」を選択します。  
scinstall ユーティリティーにより、ほかの情報を入力するためのプロンプトが表示されます。
4. **Sun Cluster Agents CD-ROM** のパスを指定します。  
ユーティリティーはこの CD をデータサービス CD-ROM として示します。
5. インストールするデータサービスを指定します。  
選択したデータサービスが scinstall ユーティリティーによって示され、この選択内容の確認が求められます。
6. **scinstall** ユーティリティーを終了します。
7. ドライブから **CD** を取り出します。

---

## Sun Cluster HA for MySQL の登録と構成

ここでは Sun Cluster HA for MySQL の構成手順について説明します。

### ▼ フェイルオーバーサービスとしての Sun Cluster HA for MySQL の登録と構成方法

次の手順で、Sun Cluster HA for MySQL をフェイルオーバーデータサービスとして構成します。この手順は、Sun Cluster の初期インストール時にデータサービスパッケージをインストールしている場合を想定しています。

Sun Cluster のインストール時に Sun Cluster HA for MySQL をインストールしていない場合は、20 ページの「scsintall ユーティリティを使用して Sun Cluster HA for MySQL パッケージをインストールする方法」へ進んでください。

それ以外の場合は、次の手順で、フェイルオーバーサービスとして Sun Cluster HA for MySQL データサービスを構成します。

1. **MySQL** のホストとなるクラスタノードの 1 つで、スーパーユーザーになります。

2. **SUNW.gds** リソースタイプを登録します。

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

3. **SUNW.HAStoragePlus** リソースタイプを登録します。

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

4. フェイルオーバーリソースグループを作成します。

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

5. **MySQL** ディスク記憶装置に対応するリソースを作成します。

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

6. **MySQL** 論理ホスト名に対応するリソースを作成します。

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

7. **MySQL** ディスク記憶装置と 論理ホスト名のリソースが組み込まれたフェイルオーバーリソースグループを有効にします。

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

---

注 `-mysql_install_db` を実行したときに使用したノードと同じノード上でリソースグループが有効になっていることを確認してください。

---

8. **MySQL** サーバーインスタンスを手動で起動します。

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

9. 管理ユーザー用のパスワードを設定します。

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

10. MySQL インスタンス用の障害モニターユーザーとテストデータベースを作成します。

```
# cd /opt/SUNWscmys/util
mysql_config ファイルを編集し、そのファイルに次のコメントを追加します。
# mysql のインストール場所 (BASEDIR)
MYSQL_BASE=

# Mysql ローカルホストの管理ユーザー (root)
MYSQL_USER=

# mysql 管理ユーザー用のパスワード
MYSQL_PASSWD=

# 構成済み論理ホスト
MYSQL_HOST=

# 障害モニターユーザーのユーザー名を指定する
FMUSER=

# 障害モニターユーザー用のパスワードを選ぶ
FMPASS=

# mysqld のソケット名 ( /tmp/<Logical host>.sock )
MYSQL_SOCKET=

# SC3.1 に限り、物理 NIC には物理ホスト名を指定してください。
# SC3.1 の論理ホスト名は、このリソースグループを置くことができる
# クラスタのすべてのノードに属します。
# 例 : 論理ホスト lh1 は、物理ノード phys-1 では hme1 に属し、
# 物理のノード phys-2 では hme3 に属します。hme1 のホスト名は phys-1-hme1、
# phys-2 の hme3 のホスト名は phys-2-hme3 です。
# 例 : MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
MYSQL_NIC_HOSTNAME=""
```

---

注 - SC 3.1 での MySQL インスタンスの例は次のようになります。

---

```
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"
```

---

注 - SC3.0U3 を使用した場合は、MYSQL\_NIC\_HOSTNAME パラメータは省略されていてもかまいません。

---

mysql\_config を編集したあとは、mysql\_register スクリプトを実行する必要があります。

```
# ./mysql_register
```

### 11. MySQL サーバーインスタンスを手動で停止します。

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

### 12. フェイルオーバーデータサービスとして MySQL を作成して登録します。

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

ha\_mysql\_config ファイルを編集し、ファイル内の次のコメントに従います。

次のパラメータは (キー=値) の形式でカスタマイズできます。

```
#
#      RS - アプリケーションのリソース名
#      RG - RS を含むリソースグループ名
# BASEDIR - Mysql bin ディレクトリ名
# DATADIR - Mysql Data ディレクトリ名
# MYSQLUSER - Mysql の起動ユーザー名
#      LH - LogicalHostname SC リソース名
# FMUSER - Mysql 障害モニターユーザー名
# FMPASS - Mysql 障害モニターユーザー用のパスワード名
# LOGDIR - mysqld がログファイルを格納するディレクトリ名
# CHECK - HA-MySQL が MyISAM のインデックスファイルをチェックしてからスタートする、YES/NO
# HAS_RS - mysql HASToragePlus SC リソース名
#
```

MySQL インスタンスの例は次のようになります。

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

### 13. 各 MySQL リソースを有効にします。

複数のインスタンスを作成した場合は、MySQL インスタンスごとにこの手順を繰り返します。

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

14. *MySQL Logicalhost* を使用して、**MySQL** インスタンスにローカルにアクセスするための管理ユーザーを追加します。

---

注 - MySQL インスタンスにアクセスする方法がローカルホストからのソケット経由だけである場合、この手順は省略してもかまいません。

---

MySQL のブートストラップを実行するとき、コマンド `mysql_install_db` は 2 人の管理ユーザー、つまり、`localhost` に属する管理ユーザーと `mysql_install_db` を実行したノードに属する管理ユーザーを作成します。

SC3.0U3 を使用していた場合、物理ノードに属する `root` ユーザーを削除して、論理ホストに属する管理ユーザーを追加する必要があります。

SC3.0U3 での MySQL インスタンスの例は次のようでした。

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

SC3.1 を使用している場合は、この MySQL インスタンスを実行するクラスタ内のすべての物理ノードに対して、管理ユーザーを作成する必要があります。

---

注 - 物理インタフェースのノード名とホスト名が異なる場合は、ホスト名を使用する必要があります。

---

SC3.1 での MySQL インスタンスの例は次のようになります。

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

---

注 - 管理ユーザーには手動で `Grant_priv` を追加する必要があります (MySQL 管理マニュアルを参照してください)。

---

---

## Sun Cluster HA for MySQL のインストールと構成の確認

ここでは、データサービスが正しくインストールされて構成されているかどうかを確認する手順について説明します。

### ▼ Sun Cluster HA for MySQL のインストールと構成の確認方法

次の手順で、Sun Cluster HA for MySQL が正しくインストールされ、構成されているかどうかを確認してください。

1. **MySQL** のホストとなるクラスタノードの **1** つで、スーパーユーザーになります。
2. すべての **MySQL** リソースがオンラインになっていることを **scstat** で確認します。

```
# scstat
```

オンラインになっていない MySQL リソースごとに、次のように **scswitch** コマンドを使用します。

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

3. **scswitch** コマンドを実行し、**MySQL** リソースグループをほかのクラスタノード (*node2* など) に切り替えます。

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

---

## Sun Cluster HA for MySQL 障害モニターの概要

この情報は、Sun Cluster HA for MySQL 障害モニターを理解するうえで役立ちます。ここでは、Sun Cluster HA for MySQL 障害モニターのプローブアルゴリズムまたは機能について説明し、プローブの失敗に関連する条件、メッセージ、回復処理を示します。

障害モニターの概念については、『*Sun Cluster 3.1 の概念*』を参照してください。

## リソースプロパティ

Sun Cluster HA for MySQL 障害モニターでは、SUNW.gds というリソースタイプと同じリソースプロパティを使用します。使用するリソースプロパティの全リストについては、SUNW.gds (5) のマニュアルページを参照してください。

## 検証アルゴリズムと機能

### ■ MySQL

- Thorough\_probe\_interval の間、休止します。
- コマンド `mysqladmin`、引数 `ping`、および定義済みの障害モニターユーザー `<fmuser>` を使用して、MySQL インスタンスに接続します。これに失敗すると、MySQL リソースが再起動します。
- 300 秒ごとに次の内容がチェックされます。
- MySQL インスタンスがスレーブ構成の場合は、MySQL インスタンスがマスターに接続されているかどうかをチェックされます。スレーブがマスターに接続されていない場合は、エラーメッセージが `syslog` に書き込まれます。
- すべてのデータベースとテーブル (内容ではない) の一覧を表示できることが確認されます。何らかのエラーが発生した場合は、エラーメッセージが `syslog` に書き込まれます。
- 定義済みのテストデータベース上で機能テストが実行されます (たとえば、テーブルの作成、テーブルへの挿入、テーブルの更新、テーブルからの削除、テーブルのドロップ)。これらの操作のどれかが失敗した場合は、MySQL リソースが再起動されます。
- すべての MySQL プロセスが無効な場合は、`pmf` による割り込みが行なわれ、ただちに MySQL リソースが再起動されます。
- MySQL リソースが繰り返し再起動した結果、`Retry_interval` 内の `Retry_count` に達した場合に、`Failover_enabled` に `TRUE` が設定されているときは、リソースグループはほかのノードにフェイルオーバーされます。

---

## Sun Cluster HA for MySQL のデバッグ

### ▼ Sun Cluster HA for MySQL でデバッグを有効にする方法

ここでは、Sun Cluster HA for MySQL のデバッグを有効にする方法について説明します。

Sun Cluster HA for MySQL は複数の MySQL インスタンスで使用できます。デバッグは、すべての MySQL インスタンスに対して有効にすることも、特定の MySQL インスタンスに対して有効にすることもできます。

MySQL コンポーネントごとに DEBUG ファイルがあり、/opt/SUNWscmys/etc に格納されます。xxx は個々の MySQL コンポーネントを表す 3 文字の略語です。

これらのファイルによって、Sun Cluster の特定のノードで特定の MySQL インスタンスに対してデバッグを有効にすることも、またはすべての MySQL インスタンスに対してデバッグを有効にすることもできます。Sun Cluster 全体にわたって、Sun Cluster HA for MySQL のデバッグを有効にしなければならない場合は、Sun Cluster 内のすべてのノードでこの手順を繰り返す必要があります。

#### 1. /etc/syslog.conf を編集します。

/etc/syslog.conf を編集し、daemon.notice を daemon.debug に変更します。

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

daemon.notice を daemon.debug に変更し、syslogd を再起動します。次に示す grep daemon /etc/syslog.conf コマンドの出力例では、daemon.debug が設定されています。

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.debug;mail.crit      /var/adm/messages
*.alert;kern.err;daemon.err                   operator
#
# pkill -1 syslogd
#
```

#### 2. /opt/SUNWscmys/etc/config を編集します。

/opt/SUNWscmys/etc/config を編集し、DEBUG= を DEBUG=ALL または DEBUG= resource に変更します。

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

---

注 - デバッグを無効にするには、この手順を逆に実行して下さい。

---

---

## Sun Cluster HA for MySQL 使用時に SC3.1 にアップグレードする

ここでは、Sun Cluster HA for MySQL 使用時に SC3.1 にアップグレードする方法について説明します。

---

注 - ここでは、ノードを SC3.1 にアップグレードする方法ではなく、Sun Cluster HA for MySQL を SC3.1 にアップグレードする方法だけを説明していることに注意してください。

---

1. `scswitch -n -j MySQL- resource` を使用して、Sun Cluster HA for MySQL リソースをシャットダウンします。

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

2. Sun Cluster のマニュアルを参照して、ノードを **SC3.1** にアップグレードします。

3. SC 3.1 上で MySQL サーバーを手動で起動します。

```
#cd <MySQL Base
>

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

4. ソケットオプションを使用して、ローカルノードから MySQL インスタンスにアクセスします。

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

MySQL インスタンスの例は次のようになります。

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

5. Sun Cluster HA for MySQL のテストデータベース `sc3_test_database` を削除します。

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

6. `mysql-database` 内のテーブル `db` から、`User=' <MySQL Faultmonitor user>'` という文字列を含むエントリをすべて削除します。

```
# 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. **mysql-database** 内のテーブル **user** から、**User=' <MySQL Faultmonitor user>'** という文字列を含むエントリをすべて削除します。

```
# 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. 障害モニターユーザーとテストデータベースを **MySQL** に追加します。

```
# cd /opt/SUNWscmys/util
mysql_configファイルを編集し、そのファイルに次のコメントを追加します。

# mysql のインストール場所 (BASEDIR)
MYSQL_BASE=

# ローカルホストの mysql 管理ユーザー (root)
MYSQL_USER=

# mysql 管理ユーザー用のパスワード
MYSQL_PASSWD=

# 構成済み論理ホスト
MYSQL_HOST=

# 障害モニターユーザーのユーザー名を指定する
FMUSER=

# 障害モニターユーザー用のパスワードを選ぶ
FMPASS=

# mysqld のソケット名 ( /tmp/<Logical host>.sock )
MYSQL_SOCKET=

# SC3.1 に限り、物理 NIC には物理ホスト名を指定してください。
# SC3.1 の論理ホスト名は、このリソースグループを置くことができる
# クラスタのすべてのノードに属します。
# 例 : 論理ホスト lh1 は、物理ノード phys-1 では hme1 に属し、
# 物理のノード phys-2 では hme3 に属します。hme1 のホスト名は phys-1-hme1、
# phys-2 の hme3 のホスト名は phys-2-hme3 です。
# 例 : MYSQL_NIC_HOSTNAME="phys-1-hme1 phys-2-hme3"
MYSQL_NIC_HOSTNAME=""
```

---

注 - SC 3.1 での MySQL インスタンスの例は次のようになります。

---

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

9. `mysql_config` を編集したあと、`mysql_register` スクリプトを実行する必要があります。

```
# ./mysql_register
```

10. MySQL サーバーを手動で停止します。

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

11. `scswitch -e -j MySQL- resource` を使用して、**Sun Cluster HA for MySQL** リソースを起動します。

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

12. `MySQL Logicalhost` を使用して、**MySQL** インスタンスにローカルにアクセスする管理ユーザーのソースアドレスを変更します。

---

注 - MySQL インスタンスにアクセスする方法がローカルホストからのソケット経由だけである場合、この手順は省略してもかまいません。

---

SC3.0U3 を使用していた場合、論理ホストに属する `root` ユーザーを削除して、物理ホストに属する管理ユーザーを追加する必要があります。

---

注 - 物理インタフェースのノード名とホスト名が異なる場合は、ホスト名を使用する必要があります。

---

SC3.1 での MySQL インスタンスの例は次のようになります。

```
# 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;
```

---

注 - 管理ユーザーには手動で `Grant_priv` を追加する必要があります (MySQL 管理マニュアルを参照してください)。

---

---

## Sun Cluster HA for MySQL 使用時に MySQL を 3.23.54 から 4.0.12 にアップグレードする

ここでは、Sun Cluster HA for MySQL 使用時に MySQL 4.0.12 にアップグレードする方法について説明します。

---

注 - ここでは、MySQL を MySQL 4.0.12 にアップグレードする方法ではなく、Sun Cluster HA for MySQL を MySQL 4.0.12 にアップグレードする方法だけを説明していることに注意してください。この手順では、新しい MySQL バイナリが前回と同じ場所にインストールされているとみなしています。新しいバイナリが新しいディレクトリにインストールされている場合は、新しい *MySQL Basedir* を使用して MySQL リソースを登録し直す必要があります。

---

1. `sctest -n -j MySQL- resource` を使用して、**Sun Cluster HA for MySQL** をシャットダウンします。  

```
# sctest -n -j MySQL- resource
```
2. 「MySQL のインストールと構成方法」の手順 3 と 4 に従って、新しい MySQL バイナリをインストールします。
3. `sctest -e -j MySQL- resource` を使用して、**Sun Cluster HA for MySQL** を起動します。  

```
# sctest -e -j MySQL- resource
```
4. `sctest -n -M -j MySQL- resource` を使用して、**Sun Cluster HA for MySQL** の障害モニターをシャットダウンします。  

```
# sctest -n -M -j MySQL- resource
```
5. MySQL のマニュアルを参照して **MySQL** データベースのアップグレードします。
6. `sctest -e -M -j MySQL- resource` を使用して、**Sun Cluster HA for MySQL** の障害モニターを起動します。  

```
# sctest -e -M -j MySQL- resource
```

# 索引

---

## C

C ロケール, 20

## M

MySQL, 6

## S

scinstall, 21

## V

/var/sadm/install/logs ディレクトリ, 20

## W

Web Start プログラム, 19

## あ

アップグレード

MySQL の 3.23.54 から 4.0.12 へのアップグレード, 32

Sun Cluster HA for MySQL 使用時の SC3.1 へのアップグレード, 29

## い

インストール, 7, 15

Sun Cluster 3.1, 20

Sun Cluster HA for MySQL

Web Start プログラムによる, 19

作成されたログファイル, 20

## か

管理ユーザー, 25

## こ

構成

MySQL, 15

Sun Cluster HA for MySQL の登録, 21

確認, 18

サポートされる構成, 7

制限, 7

要件, 11

コマンド

chmod, 18

chown, 16, 18

groupadd, 16

scrgadm, 22

scstat, 26

scswitch, 22, 26

useradd, 16

## し

### 障害モニター

検証アルゴリズムと機能, 27

理解, 26

リソースプロパティ, 27

## て

デバッグ, 有効にする方法, 27

## ふ

ファイル, インストールログ, 20

ブートストラップ, 18

## り

### リソースタイプ

SUNW.gds, 22

SUNW.HAStoragePlus, 22

## ろ

ログファイル, インストール, 20

ロケール, 20