

Using MySQL Toolkit in a HP Serviceguard Cluster

=====
Revision: B.06.00

CONTENTS:

- A. MySQL Package Configuration Overview
- B. Setting Up the Database Server Application
- C. Setting Up MySQL with the Toolkit
 - C.1 Toolkit Overview
 - C.2 MySQL Configuration File (my.cnf)
 - C.3 Toolkit Configuration File (hamysql.conf)
- D. Package Configuration File and Control Script
- E. Applying the Configuration and Running the Package
- F. MySQL Database Server Maintenance
- G. Do and Don't with MySQL Toolkit
 - A. DOs
 - B. DON'Ts

PREFACE

This README file describes the MySQL Toolkit for use in the HP Serviceguard environment. The README file is intended for users who want to configure MySQL Database Server application toolkit under HP Serviceguard cluster environment using MySQL Toolkit. This toolkit supports the Enterprise MySQL Database Server Application 5.0.56 and later.

It assumes that readers are already familiar with the HP Serviceguard cluster configuration as well as MySQL Database server concepts and their installation and configuration procedures.

NOTE: This toolkit supports:

- HP Serviceguard versions
 - * A.11.19
 - * A.11.20

- HP-UX 11i v2 and HP-UX 11i v3

At time of publication, this version supports the above mentioned SG, application and HPUX versions. More recent versions of these products may be certified with B.06.00. For the latest information, see the compatibility matrix located at:

<http://docs.hp.com/en/ha.html>

NOTE:

This version of the toolkit supports both, legacy and modular style packages.

NOTE:

This version of the toolkit is not supported with VxVM and CFS.

This toolkit consists of a set of shell scripts used by a Package

Control Script to start, stop, and monitor the MySQL Database Server. The toolkit works with the Serviceguard package control script, managed by Serviceguard in case of legacy packages and Serviceguard master control script in case of modular packages. Each MySQL Database server instance is configured in its own Serviceguard package. This README file assumes that users have used swinstall to properly install both Serviceguard and the Enterprise Cluster Master Toolkit (referred to as the ECMT), which includes the scripts for MySQL.

For legacy packages, there will be one user configuration script (hamysql.conf) and three functional scripts (toolkit.sh, hamysql.sh, hamysql.mon) which work with each other to integrate MySQL with the Serviceguard package control scripts.

For modular packages, there is an Attribute Definition File(ADF), a Toolkit Module Script (tkit_module.sh) and a Toolkit Configuration File Generator Script (tkit_gen.sh) which work with the four scripts mentioned above for legacy packages to integrate MySQL with the Serviceguard Master Control Script.

When ECMT is installed the following files are located in /opt/cmcluster/toolkit/mysql

File Name	Description
README	-- This file
hamysql.conf	-- User defined variables for configuring MySQL in a specific environment
hamysql.sh	-- The main shell script
hamysql.mon	-- Script to monitor MySQL to ensure it is running
toolkit.sh	-- Interface between the Serviceguard package control script and hamysql.sh in case of legacy packages and interface between the Serviceguard master control script and hamysql.sh in case of modular packages.

Following three files are also installed and they are used only for the modular method of packaging.

The following Attribute Definition File (ADF) is installed in /etc/cmcluster/modules/ecmt/mysql.

File Name	Description
mysql	For every parameter in the legacy toolkit user configuration file, there is an attribute in the ADF. It also has an additional attribute TKIT_DIR which is analogous to the package directory in the legacy method of packaging. The ADF is used to generate a modular package ASCII template file.

For the alert mail notification feature, an additional parameter called ALERT_MAIL_ID is introduced in the ADF.

ALERT_MAIL ID: Sends an email message to the specified email address when

fail, packages fail. This email is generated only when packages fail,

and not when a package is halted by the operator.

To send this email message to multiple recipients, a group email ID must be created and specified for this parameter.

When an email ID is not specified for this parameter, the script does not send out this email.

The following files are located in /etc/cmcluster/scripts/ecmt/mysql after installation.

File Name	Description
tkit_module.sh	This script is called by the Master Control Script and acts as an interface between the Master Control Script and the Toolkit interface script (toolkit.sh). It is responsible for calling the Toolkit Configuration File Generator Script (described below).
tkit_gen.sh	This script is called by the Module Script when the package configuration is applied using 'cmapplyconf' to generate the toolkit user configuration file in the package directory (TKIT_DIR).

The MySQL Database server must be installed on all nodes that will run the MySQL Packages in the HP Serviceguard Cluster. The node where this runs is termed as PRIMARY node. The other nodes that are ready to take up the service in case of failure are termed STANDBY nodes. The MySQL Database Server application runs on the primary node servicing MySQL Database requests from clients. In the event of a failure on the primary node, the package will fail over to the standby node. This means that all necessary configuration information on all nodes must be identical and the resources must be available to all supporting nodes. The data must be stored on shared disks and these disks must be accessible by the same pathnames on each node.

MySQL database server supports multiple DB instances running on the same node. This toolkit helps in creating multiple MySQL packages. Each package corresponds to a separate DB server instance with its own database, toolkit files, and configuration.

A. MySQL Package Configuration Overview

The database server can be configured in two different ways:

- * (shared config) by putting the configuration files in a shared filesystem
- * (local config) by putting the configuration files on a single node, then replicating the files to all other nodes in the cluster

~~~~~  
Local Configuration

If you opt to store the configuration files on a local disk, the configuration MUST be replicated to local disks on all nodes configured to run the package. If any change is made to the configuration, the file must be copied to all nodes. As it is up to the user to ensure the systems remain synchronized, the recommendation is to put all configuration and data files on shared storage.

~~~~~  
Shared Configuration

This is the recommended configuration. Here, the configuration and database files are on shared disks, visible to all nodes. Since the storage is shared, there is no additional work to ensure all nodes have the same configuration at any point in time.

~~~~~  
To run MySQL in a HP Serviceguard environment:

- \* Each node must have the same version of the MySQL Database Server software installed.
- \* Each node that will be configured to run the package must have access to the configuration files. The most efficient way to ensure all nodes have access to the configuration files is to put the configuration files on shared storage that is accessible to all nodes.
- \* Multiple databases configured in the same cluster will require unique configuration per database. Each DB instance will require its own volume group, set of configuration files, and copy of the toolkit customized for the unique instance.
- \* All data files must be accessible to all nodes configured to run the package. This means that all data files will need to reside on shared storage that is accessible to all nodes.
- \* Each package is assigned an IP address called the relocatable (or package) IP address.

When the package fails over from one node to another, the following actions occur:

Primary Node

- 1. The package is halted on the node where it is currently running.

As a result, all (package) resources are halted.

2. The relocatable IP address is removed from this node.
3. File systems are unmounted and all volume groups assigned to this package are deactivated.

Standby Node

- 
4. Volume groups are activated and file systems are mounted.
  5. The relocatable IP address is moved over to the new node.
  6. All resources are started up, and the database is brought up.
  7. Clients will (continue to) connect through the same (relocatable) IP address.

#### Multiple Database Instances

MySQL database server supports multiple DB instances running on the same node. Using the toolkit, multiple MySQL packages can be easily configured to run on the same node. One copy of the toolkit can serve exactly one package. So, to configure multiple packages, you must create separate directories - one per package - and configure each package to run a unique database instance.

#### B. Setting Up The Database Server Application

Before creating the packages, you must configure the shared storage and create the database.

1. Follow the instructions in the chapter entitled "Building an HA Cluster" in the document "Managing Serviceguard" for volume group (VG) and file system creation. Create a unique VG and filesystem on shared storage for each instance of the DB to be included in the configuration.
2. On each node, create a mount point for each filesystem. The mount point for a specified instance/package must be the same for all nodes.

The following example shows a configuration of MySQL for a filesystem named "/MySQL\_1" on /dev/vg01.

1. Assuming that LVM has been set up on the shared storage, create a filesystem (e.g., /MySQL\_1) on the LV and mount point (e.g., /MySQL\_1) on each of the nodes to mount the filesystem. If you need help creating the VG and filesystem, please refer to the chapter entitled "Building an HA Cluster" in the manual "Managing Serviceguard" for explicit instructions.

2. Following the instructions in the documentation for MySQL, create the database on the lvol in /MySQL\_1. This information may be viewed online at the following link: [http://www.mysql.com/documentation/mysql/bychapter/manual\\_Tutorial.html#Creating\\_database](http://www.mysql.com/documentation/mysql/bychapter/manual_Tutorial.html#Creating_database)
3. Copy the configuration file "/etc/my.cnf" to "/MySQL\_1/my.cnf".
4. Modify "/MySQL\_1/my.cnf" to configure the DB for your unique environment. Changes may include specific assignments to the following parameters:

```
[mysqld]
* datadir=/MySQL_1/mysql
* socket=/MySQL_1/mysql/mysql.sock
* port= <UNIQUE PORT NUMBER>

[mysqld_safe]
* err-log=/MySQL_1/mysql/mysqld.err
* pid-file=/etc/cmcluster/pkg/mysql1/mysqld.pid
```

Since each database instance resides in its own filesystem, multiple database instances can be configured in the environment using this method.

## C. Setting Up MySQL with the Toolkit

### C.1 Toolkit Overview

The following files are included in the toolkit for MySQL:

| File Name    | Description                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| README       | Readme file for the MySQL toolkit (this file)                                                                        |
| toolkit.sh   | Toolkit Interface Script. Interface between the Package Control Script and the toolkit main script (hamysql.sh).     |
| hamysql.conf | Toolkit Configuration File. Contains a list of pre-defined variables that may be changed for your unique environment |
| hamysql.sh   | Toolkit Main Script. Contains the internal functions that support start/stop of a MySQL instance.                    |
| hamysql.mon  | Toolkit Monitor Script. Contains the internal functions for monitoring a DB server instance.                         |

NOTE: The following three scripts are used only during the modular method of packaging.

- Attribute Definition File (mysql)
 

For every parameter in the legacy toolkit user configuration file, there is an attribute in the ADF. It also has an additional attribute TKIT\_DIR which is analogous to the package directory in the legacy method of packaging. The ADF is used to generate a package ASCII template file.
- Module Script (tkit\_module.sh)
 

This script is called by the Master Control Script and acts as an interface between the Master Control Script and the Toolkit interface script (toolkit.sh). It is also responsible for calling the Toolkit Configuration File Generator Script (see description below).
- Toolkit Configuration File Generator Script (tkit\_gen.sh)
 

This script is called by the Module Script when the package configuration is applied using 'cmapplyconf' to generate the user configuration file in the package directory(TKIT\_DIR).

-----

## C.2 MySQL Configuration File (my.cnf)

The following parameters are contained in the configuration file /etc/my.cnf. This file must be copied to the file system on the shared storage (in our example, /etc/my.cnf would be copied to /MySQL\_1/my.cnf). Then parameters need to be manually set with unique values for each DB instance configured.

~~~~~

```
[mysqld]

datadir=/MySQL_1/mysql # Data Directory for MySQL DB

socket=/MySQL_1/mysql/mysql.sock # Socket file for Client
                                # Communication.

port=3310                    # Port Number for Client
                                # Communication.

[mysqld_safe]

err-log=/MySQL_1/mysql/mysql.err # Safe-mysqld's
                                # Error Log file.

pid-file=/etc/cmcluster/pkg/MySQL1/mysqld.pid # pid file Path.
~~~~~
```

=====

NOTE : mysqld_safe was previously known as "safe_mysqld". For older version of MySQL user need to modify suitable sections in the my.cnf file. Do the following for running this toolkit with the older versions of the MySQL :-

- Create a soft link named "mysqld_safe" in the MySQL installation directory (typically /usr/bin for RH) to point to "safe_mysqld"

```
CMD> ln -s /usr/bin/mysqld_safe /usr/bin/safe_mysqld
```

- Grant the execute permission to the newly created link.

```
CMD> chmod 755 mysqld_safe
```

=====

NOTE: The following sections in this document describe the legacy mode of Serviceguard packages. For information on creating the Serviceguard package using the modular method, please refer to the whitepaper "Modular package support in Serviceguard for Linux and ECM Toolkits" at <http://docs.hp.com/en/ha.html>

C.3 Toolkit Configuration File (hamysql.conf)

All the toolkit user configuration variables are stored in a single file - hamysql.conf, in Shell script format. User variables and sample values are included below:

1. CONFIGURATION_FILE_PATH="/MySQL_1/mysql/my.cnf"
or
DATA_DIRECTORY="/MySQL_1/mysql"

Only one of the two variables (either CONFIGURATION_FILE_PATH or DATA_DIRECTORY) should be defined. If both are defined, CONFIGURATION_FILE_PATH is used and DATA_DIRECTORY is ignored.

NOTE: If DATA_DIRECTORY is used, "my.cnf" MUST reside in the DATA_DIRECTORY location. This directory is also used as the data directory for this instance of the database server.

2. PID_FILE="/var/run/mysql/mysqld.pid"

This is the path where PID file for the MySQL daemon is created for the Parent PID. If this variable is defined, it overrides the "pid-file" defined in the MySQL configuration file "my.cnf". The PID identified in this file is monitored by the toolkit monitor, "hamysql.mon".

3. MAINTENANCE_FLAG (For example: MAINTENANCE_FLAG="yes")

This variable will enable or disable maintenance mode for the MySQL package. By default this is set to "yes". In order to disable this feature MAINTENANCE_FLAG should be set to "no". When MySQL needs to be maintained the file "<package directory>/mysql.debug" needs to be touched. During this maintenance period MySQL process monitoring is paused. Even if the MySQL instance is brought down the package will

not be failed over to the adoptive node. To continue monitoring and turn off maintenance mode, you should remove the "mysql.debug" file. The user should ensure that the MySQL instance is running properly before exiting the maintenance phase, since monitoring of the instance will resume once out of the maintenance phase.

4. RETRY_TIMES=0

This variable is recommended to be set to "0". This will multiply the SERVICE_RESTART of the MySQL package service (monitor).

For Example: If the package service is set to

```
SERVICE_RESTART = 2 &  
RETRY_TIMES=2
```

Then actual retries = (SERVICE_RESTART + 1) * RETRY_TIMES.

5. MONITOR_INTERVAL=5 # Seconds.

This is the interval at which hamysql.mon will monitor the process with PID in the PID_FILE (mysqld parent process).

The time unit granularity is seconds. So, setting MONITOR_INTERVAL=5 means the parent process is monitored every 5 seconds.

D. Package Configuration File and Control Script

The following steps identify the changes needed to the Serviceguard package configuration file and control script templates to customize them for your specific configuration. This information was extracted from the chapter entitled "Configuring Packages and Their Services" of the "Managing Serviceguard" manual. For detailed information on configuring and managing a package, please refer to the "Managing Serviceguard" manual.

The following parameters are either in the configuration file or control script as identified below, and must be manually edited for your unique environment. In the following example, values are assigned for a package named "mysql_1".

```
-----  
Variable Name          Value          Description  
-----  
[configuration file parameters]  
PACKAGE_NAME          : mysql_1      # Package Name  
  
NODE_NAME              : node1        # nodes that can run the package  
NODE_NAME              : node2  
  
RUN_SCRIPT             : <control script> # Script to start up the service  
  
HALT_SCRIPT            : <control script> # Script to halt the service  
-----
```

```

SERVICE_NAME      : "mysql_monitor"      # Service Name

[control script parameters]

VG                  : vgMySQL              # VG created for this package

LV                  : /dev/vgMySQL/lvol1    # Logical vol created in VT

FS                  : /MySQL_1            # File system for DB

FS_TYPE             : "ext2"              # FS type is "Extended 2"

FS_MOUNT_OPT       : "-o rw"             # mount with read/write options

SUBNET              : "192.70.183.0"      # Package Subnet

IP                  : "192.70.183.171"    # Relocatable IP

#The service name must be the same as defined in the package
#configuration file.

SERVICE_NAME="mysql_monitor"
SERVICE_CMD="/etc/cmcluster/pkg/MYSQL1/toolkit.sh monitor"
SERVICE_RESTART="-r 0"

```

In addition to the variables, modify the following functions in the package control script:

```

function: customer_defined_run_cmds
change  : replace the "empty" line (": #do nothing...") with
         "/etc/cmcluster/pkg/MySQL1/toolkit.sh start"

function: customer_defined_halt_cmds
change  : replace the "empty" line (": #do nothing...") with
         "/etc/cmcluster/pkg/MySQL1/toolkit.sh stop"

```

Assuming the cluster has already been configured, edit the cluster configuration file to update the "MAX_CONFIGURED_PACKAGES" (e.g., increase MAX_CONFIGURED_PACKAGES by the number of packages you are adding to the cluster). After editing the configuration file apply the change via the `cmapplyconf -C <cluster_config_file>` command.

Distribute the cluster configuration file to all the nodes of the cluster using `rcp` (this is optional but recommended).

The cluster has been updated so that you can add packages. To actually bring the cluster up, issue the command `"cmruncl"`, then `"cmviewcl"` to see the cluster up and running.

More information on cluster configuration, management, and maintenance may be found in the "Managing Serviceguard" manual.

E. Applying the Configuration and Running the Package

Once the database has been set up on the shared storage, you are ready to configure the toolkit and package.

1. Create a directory in the `cmcluster` directory for each package (e.g., `/etc/cmcluster/pkg/MySQL1`).
2. Copy the toolkit files from `/opt/cmcluster/toolkit/mysql` to the package directory (`/etc/cmcluster/pkg/MySQL1`), and change directory to the package directory.
3. Configure `hamysql.conf` for your unique configuration as described in the section entitled "Toolkit Configuration File (`hamysql.conf`)" of this document.
4. In the package directory, generate the package configuration and control templates with the `cmmakepkg` command: `cmmakepkg -p MySQL1.conf` (configuration template) `cmmakepkg -s MySQL1.cntl` (control template)
5. Use any editor to modify the package templates with your specific information, as described in the preceding section ("Package Configuration File and Control Script") of this document.
6. Change the owner & group of the package directory to the "mysql" user.

Ex. `chown mysql:mysql /etc/cmcluster/pkg/MySQL1`

7. Ensure both `root` and `mysql` users have read/write/execute permissions for the package directory
8. Distribute the package directory to all nodes in the cluster.
9. Apply the Serviceguard package configuration using the command `"cmapplyconf -P MySQL1.conf"`
10. Enable package switching for MySQL package using `cmmodpkg -e -n node1 -n node2 mysql_1`
`cmmodpkg -e mysql_1`
11. The package should now be running. If it is not, you can start up the package by issuing the `cmrunpkg` command:
`cmrunpkg mysql_1`

Repeat these procedures to create multiple MySQL instances running in the Serviceguard environment.

F. Database Maintenance

There might be situations, when the MySQL database has to be taken down for maintenance purposes like changing configuration, without

having the instance to migrate to standby node. The following procedure should be followed during maintenance:

Note: The example assumes that the package name is `mysql_1`, package directory is `/etc/cmcluster/pkg/MySQL1` and the MySQL DATADIR is configured as `/MySQL_1/mysql`

- Disable the failover of the package through `cmmodpkg` command.

```
$ cmmodpkg -d mysql_1
```

- Pause the monitor script

Create an empty file `/etc/cmcluster/pkg/MySQL1/mysql.debug` as shown below:

```
$ touch /etc/cmcluster/pkg/MySQL1/mysql.debug
```

Toolkit monitor script which continuously monitors MySQL process, would now stop monitoring this daemon process. A message "MySQL toolkit pausing monitoring and entering maintenance mode" appears in the Serviceguard Package Control script log.

- If required, stop the MySQL database instance as shown below:

```
$ /etc/cmcluster/pkg/MySQL1/toolkit.sh stop
```

- Perform maintenance actions (Example: changing the configuration parameters in the parameter file of the MySQL instance. If this file is changed, please remember to distribute the new file to all cluster nodes).

- Start the MySQL database instance again if you stopped it

```
$ /etc/cmcluster/pkg/MySQL1/toolkit.sh start
```

- Allow monitoring scripts to continue normally as shown below:

```
$ rm -f /etc/cmcluster/pkg/MySQL1/mysql.debug
```

A message "Starting MySQL toolkit monitoring again after maintenance" appears in the Serviceguard Package Control script log.

- Enable the package failover

```
$ cmmodpkg -e mysql_1
```

Note: If the package fails during maintenance (e.g., the node crashes), the package will not automatically fail over to an adoptive node. It is the responsibility of the user to start the package up on an adoptive node. Please refer to the manual "Managing Serviceguard" for more details.

Note: This feature is enabled only when the configuration variable MAINTENANCE_FLAG is set to "yes" in the MySQL toolkit configuration file.

G. Do and Don't with MySQL Toolkit

Each of the Toolkits are designed with few assumptions that integrate with the respective application and with HP Serviceguard. It is recommended to follow the below mentioned guidelines for setting up the Package using MySQL Toolkit.

A. DOs

1. Name the package name, package configuration file name, Package control script name in a consistent and self-explanatory protocol.

```
Ex. PACKAGE NAME           mysql1
    Package Configuration File mysql1.ascii
    Package Control Script   mysql1.cntl
```

2. Keep the package configuration file, package control file and toolkit files in separate directories for each of the packages.

3. Add <Package directory path>/toolkit.sh start | stop in the customer_defined_run_cmds | customer_defined_halt_cmds functions respectively for the MySQL toolkit.

4. Ensure that the toolkit.sh, hamysql.sh, hamysql.mon & package control script have execute permission.

B. DON'Ts

Do not enable "HA_APP_SERVER" in package control script for each of the MySQL packages.

```
Ex. #HA_APP_SERVER="pre-IP"
    #HA_APP_SERVER="post-IP"
```

© Copyright 2010 Hewlett-Packard Development Company, L.P. Confidential computer software. Valid license from HP required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license. The information contained herein is subject

to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.