Symantec™ ApplicationHA Agent for MySQL Configuration Guide

Linux on KVM

6.2

December 2014

Symantec

Symantec™ ApplicationHA Agent for MySQL Configuration Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Agent version:: 5.1.2.0

Document version: 6.2 Rev 0

Legal Notice

Copyright © 2014 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Veritas, Veritas Storage Foundation, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THE 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. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations, whether delivered by Symantec as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apj@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec website.

https://sort.symantec.com/documents

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

Contents

Technical Support 4

Chapter 1 Introducing the Symantec ApplicationHA Agent for MySQL 9

About the Symantec ApplicationHA agent for MySQL 9
About installing and removing the ApplicationHA agent for MySQL 10
Supported software 10
Supported application versions 10
Supported virtualization environments 10
Supported operating systems on virtual machines 11
About MySQL 11
MySQL agent functions 11
Online 11
Offline 12
Monitor 12
Clean 13

Chapter 2 Installing MySQL in an ApplicationHA environment 14

Configuring the MySQL base directory and database directory 14
Configuring Ports 14
Configuring the MySQL database user 15
Adding a dedicated database administrator with shutdown privileges only 15

Chapter 3 Configuring application monitoring with Symantec ApplicationHA 17

About configuring application monitoring with ApplicationHA 17
Before configuring application monitoring for MySQL 18
Accessing the Symantec High Availability view 18
Configuring application monitoring for MySQL agent 19

Chapter 4 Troubleshooting the agent for MySQL 23

Starting the MySQL instance outside the Symantec ApplicationHA environment 23
Reviewing error log files 24
Using MySQL log files 25
Reviewing ApplicationHA log files 25
Reviewing cluster log files 25
Using trace level logging 25

Appendix A Resource type definitions 27

About the resource type and attribute definitions 27
Resource type definition for MySQL agent 27
Attribute definition for MySQL agent 28

Appendix B Detail monitoring 33

Setting the PATH variable 33
Setting up detail monitoring for ApplicationHA agent for MySQL Server 33

Chapter 1

Introducing the Symantec ApplicationHA Agent for MySQL

This chapter includes the following topics:

- About the Symantec ApplicationHA agent for MySQL
- About installing and removing the ApplicationHA agent for MySQL
- Supported software
- About MySQL
- MySQL agent functions

About the Symantec ApplicationHA agent for MySQL

The Symantec ApplicationHA agents monitor specific resources within an enterprise application. They determine the status of resources and start or stop them according to external events.

The Symantec ApplicationHA agent for MySQL Server provides high availability for MySQL Servers in a cluster.

About installing and removing the ApplicationHA agent for MySQL

When you install or uninstall Symantec ApplicationHA, the ApplicationHA agent for MySQL is automatically installed or removed. For more information, see the *Symantec ApplicationHA Installation and Upgrade Guide*.

When you run the installer or uninstall program that accompanies the quarterly agent pack release of high availability agents from Symantec, the latest version of the ApplicationHA agent for MySQL is automatically installed or removed. For more information, see the *Symantec ApplicationHA Agent Pack Installation Guide*.

Supported software

The Symantec ApplicationHA agent for MySQL supports the following software versions:

- Symantec ApplicationHA agent for MySQL can be installed and run inside virtual machines that have Symantec ApplicationHA 6.2 installed.
- The following versions of the Veritas Operations Manager components are supported:
  - Veritas Operations Manager Management Server 6.0 or later
  - Veritas Operations Manager managed host for Linux: 6.0 or later

Supported application versions

Table 1-1 lists the MySQL versions that Symantec ApplicationHA 6.2 currently supports on virtual machine.

Table 1-1 Supported application versions

Application	Version
MySQL	5.x

Supported virtualization environments

Symantec ApplicationHA can be installed and run inside virtual machines in a KVM virtualization environment, running Red Hat Enterprise Linux (RHEL) 6, Update 3 and 4 in the physical machine.

Supported operating systems on virtual machines

Table 1-2 shows the supported operating systems for Symantec ApplicationHA 6.2.

Table 1-2 Supported guest operating systems

Operating systems	Levels	Kernel version
Red Hat Enterprise Linux 6	Updates 3, 4, and 5	2.6.32-279.el6 2.6.32-358.el6 2.6.32-431
Red Hat Enterprise 7	-	3.10.0-123

Note: Only 64-bit operating systems are supported.

If your system is running a lower level of Red Hat Enterprise Linux, than indicated in Table 1-2, you must upgrade it before attempting to install Symantec ApplicationHA. Consult the Red Hat documentation for more information on upgrading or reinstalling your operating system.

Symantec supports only Red Hat distributed kernel binaries.

Symantec products operate on subsequent kernel and patch releases provided the operating systems maintain kernel ABI (application binary interface) compatibility.

About MySQL

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

MySQL agent functions

The agent consists of resource type declarations and agent executables. The agent executables are organized into online, offline, monitor, and clean functions.

Online

The online function performs the following tasks:

- Verifies that the required attributes are set correctly.

- Verifies that the MySQL Server instance is not already online. If the instance is online, the online operation exits immediately.
- Kills any MySQL processes remains by using the user name associated with the specific resource.
- Attempts to start the MySQL server instance with the following command:

  ```
  $ BaseDir/bin/mysqld_safe --defaults-file=MyCnf \
  --datadir=DataDir --user=MySQLUser
  ```

  This command gets executed in the context of MySQLUser, specifying the MySQL configuration file.
- Checks if the server starts completely.

Offline

The offline function performs the following tasks:

- Verifies that the required attributes are set correctly.
- Verifies that the MySQL Server instance is not offline.
- If the instance is already offline, the operation verifies if any processes belonging to this MySQL resource, exist.
- Attempts to stop the MySQL server instance with the command:

  ```
  $ BaseDir/bin/mysqladmin --user=MySQLAdmin \
  --password=MySQLAdminPasswd shutdown
  ```

  The command always gets executed in the context of MySQLUser. Then the offline operation kills any existing process that belongs to this MySQL server instance.

Monitor

The monitor function monitors the states of the MySQL Servers on all nodes within the cluster. The operation performs the following tasks:

- The monitor function conducts a first level check to determine that the MySQL Server processes are running on the system in the cluster. If the first level check does not find these processes running on the node, the check exits immediately, and reports the instance as OFFLINE.

  Note: The agent sets the value of level attribute ToleranceLimit of cluster MySQL type to 1. This ensures that the application gets an opportunity to restart a failed mysql instance before the agent flags the instance OFFLINE to initiate a failover.

- If the LevelTwoMonitorFreq attribute is set to greater than 0, the monitor operation conducts a second level check.
- The agent uses a connect(3c) method on the IP address specified by the HostName agent attribute to check for the MySQL server to listen to the port defined by the Port attribute.
- The agent then uses the monitor command to verify that the MySQL server is up.

  ```
  $ BaseDir/bin/mysqladmin --user=MySQLAdmin \
  --password=MySQLAdminPasswd status
  ```

  The command is executed in the context of the MySQLUser.
- The monitor operation can perform a customized check using a user-supplied monitoring utility depending upon the MonitorProgram attribute.

Clean

In case of a failure or after an unsuccessful attempt to online or offline a MySQL Server instance, the clean operation performs the following tasks:

- Attempts to gracefully shut down the MySQL server instance with the command:

  ```
  $ BaseDir/bin/mysqladmin --user=MySQLAdmin \
  --password=MySQLAdminPasswd shutdown
  ```

  The command always gets executed in the context of MySQLUser.
- Kills any remaining process pertaining to this MySQL instance.

Chapter 2

Installing MySQL in an ApplicationHA environment

This chapter includes the following topics:

- Configuring the MySQL base directory and database directory
- Configuring Ports
- Configuring the MySQL database user
- Adding a dedicated database administrator with shutdown privileges only

Configuring the MySQL base directory and database directory

Ensure that each database instance manages a unique database directory, specified by the “datadir” configuration parameter. The base (or installation) directory, specified by the “basedir” configuration parameter may be shared across multiple instances of the database server. These values need to be registered with the agent using the BaseDir and DataDir agent attributes.

Review the information on the BaseDir and DataDir agent attributes.

Configuring Ports

To ensure that multiple instances can be hosted on the same failover node, the HostName/Port pair combination has to be unique. MySQL server by default listens on port 3306. This can be changed using the “port” configuration parameter.

Configuring the MySQL database user

Symantec recommends to use a non-root user while starting a MySQL database. This is the UNIX user owning the database directory and its files. The value for this attribute should be identical to the database configuration parameter "user", if specified in the database configuration file (my.cnf), and should be registered with the agent using the MySQLUser agent attribute.

The following is an excerpt from a typical MySQL configuration file (my.cnf) that is used to start a database instance.

```
# The following options will be passed to all MySQL clients
[client]
# password       = your_password
port            = 3306
socket          = /tmp/mysql.sock
# Here follows entries for some specific programs
# The MySQL server
[mysqld]
user            = mysql
basedir         = /usr/local/mysql
datadir         = /usr
pid-file        = /var/lib/mysql
port            = 3306
socket          = /tmp/mysql.sock
tmpdir          = /var/tmp
.. truncated ..
# Specify the bind address
bind-address    = 11.111.1.1
```

Adding a dedicated database administrator with shutdown privileges only

It is strongly recommended that you create a dedicated database administrator with privileges only to shutdown a particular instance of the database, locally. Do not use the default “root” database administrator that has unrestricted database privileges, as the agent does not need them.

To add a dedicated database administrator *MySQLAdmin* at the mysql prompt, do the following:

```
mysql> create user 'MySQLAdmin'@'localhost' identified by 'XXXXXX' ;
Query OK, 0 rows affected (0.00 sec)
mysql> create user 'MySQLAdmin'@'127.0.0.1' identified by 'XXXXXX' ;
```

```
Query OK, 0 rows affected (0.00 sec)
mysql> grant shutdown on *.* to 'MySQLAdmin'@'localhost' ;
Query OK, 0 rows affected (0.00 sec)
mysql> grant shutdown on *.* to 'MySQLAdmin'@'l27.0.0.1' ;
Query OK, 0 rows affected (0.00 sec)
mysql> quit
```

This assumes that the session owner has grant access to add a database user and assign privileges for database shutdown to that user.

Ensure that you can shutdown the database instance using this database user:

```
$ BaseDir/bin/mysqladmin --user=MySQLAdmin --password=XXXXXX shutdown
```

Where *MySQLAdmin* is the database administrator being created and registered with the agent, and *XXXXXX* is the password being set for this administrator, which is encrypted and specified using the MySQLAdminPasswd agent attribute.

For information on MySQLAdmin and MySQLAdminPasswd attributes, review the Agent attributes section.

Chapter 3

Configuring application monitoring with Symantec ApplicationHA

This chapter includes the following topics:

- About configuring application monitoring with ApplicationHA
- Before configuring application monitoring for MySQL
- Accessing the Symantec High Availability view
- Configuring application monitoring for MySQL agent

About configuring application monitoring with ApplicationHA

This chapter describes the steps to configure application monitoring with ApplicationHA in a virtualization environment.

Consider the following points before you proceed:

- You configure an application for monitoring on a virtual machine using the Symantec ApplicationHA Configuration Wizard.
- The Symantec ApplicationHA Configuration Wizard is launched when you click **Configure Application Monitoring** in the Symantec High Availability view of the Veritas Operations Manager (VOM) Management Server console.
- In this release, the wizard allows you to configure monitoring for only one application per virtual machine.

To configure another application using the wizard, you must first unconfigure the existing application monitoring.

- After you have configured monitoring for an application using the wizard, you can configure monitoring for other applications residing in the same virtual machine, using Symantec Cluster Server (VCS) commands.
  For more information read the following technote:
  http://www.symantec.com/docs/TECH159846
- After configuring MySQL for monitoring, if you create another MySQL instance, this new instance is not monitored as part of the existing configuration.
  In such a case, you must first unconfigure the existing configuration and then reconfigure the application using the wizard. You can then select all the instances for monitoring.

Before configuring application monitoring for MySQL

Ensure that you complete the following tasks before configuring application monitoring for MySQL on a virtual machine:

- Install Veritas Operations Manager (VOM) Management Server. For more information on working with VOM, see the *Symantec ApplicationHA User's Guide*. For information on accessing the Symantec High Availability view: See "Accessing the Symantec High Availability view " on page 18.
- Install ApplicationHA guest components on the virtual machine that you need to monitor.
- Assign ApplicationHA - Configure Application Monitoring (Admin) privileges to the logged-on user on the virtual machine where you want to configure application monitoring.
- Install the application and the associated components that you wish to monitor on the virtual machine.
- If you have configured a firewall, ensure that your firewall settings allow access to ports used by ApplicationHA installer, wizards, and services.
  Refer to the *Symantec ApplicationHA Installation Guide* for a list of ports and services used.

Accessing the Symantec High Availability view

To administer an application on a virtual machine that is running in the KVM environment, you must access the Symantec High Availability view of the Veritas Operations Manager (VOM) Management Server console.

From the Symantec High Availability view, you can perform administrative actions such as:

- Start an application
- Stop an application
- Configure application monitoring
- Unconfigure application monitoring
- Enable application heartbeat
- Disable application heartbeat
- Enter maintenance mode
- Exit maintenance mode

To access the Symantec High Availability view

1. Log on to the VOM Management Server console.
2. Select the Server perspective and expand Manage in the left pane.
3. Expand the Organization, or Uncategorized Hosts to navigate to the virtual machine.
4. Right-click the required virtual machine, and then click **Manage ApplicationHA**.

   The Symantec High Availability view appears.

Configuring application monitoring for MySQL agent

Perform the following steps to configure monitoring for MySQL on a virtual machine.

To configure application monitoring for MySQL

1. In the Symantec High Availability view of theVeritas Operations Manager Management Server Console, click **Configure Application Monitoring**.

   This launches the Symantec ApplicationHA Configuration Wizard.
2. Review the information on the Welcome screen and then click **Next**.

   The wizard lists all the supported applications for the system.
3. Select **MySQL** and then click **Next**.

   The MySQL Server Configuration File Specification screen appears.

4 On the MySQL Server Configuration File Specification screen, enter the path where the MySQL Server Configuration File exists on the virtual machine and then click **Next**.

The MySQL Server Instance Details screen for MySQL instances appears.

5 Enter the appropriate values in the following fields.

MySQL Database Administrator	The administrative database user of the MySQL server with privileges to shutdown the database. Symantec recommends creating a dedicated account in the database, with shutdown privileges only.
MySQL Database AdministratorPassword	Password for the database administrator specified in the MySQLAdmin attribute. The password is encrypted using the VCS encrypt utility, vcsencrypt(1m).
MySQLUser	The dedicated OS login created while installing the MySQL server. The database server will be started as this user.
DataDir	The absolute path to the directory storing the database being managed by this instance of the server.
BaseDir	The installation path of the MySQL Database server.
HostName	Virtual hostname for this MySQL Database instance.
Port	Represents the port number dedicated to the MySQL Server.
EnvFile	Completes path of file name to source to set the environment prior to executing MySQL programs.

6 Click **Next**

7 The wizard performs the application monitoring configuration tasks. The ApplicationHA Configuration screen displays the status of each task.

8 After all the tasks are complete, click **Next**.

Note: If the configuration tasks fail, click **Diagnostic information** to check the details of the failure.

You then have to run the wizard again to configure application monitoring.

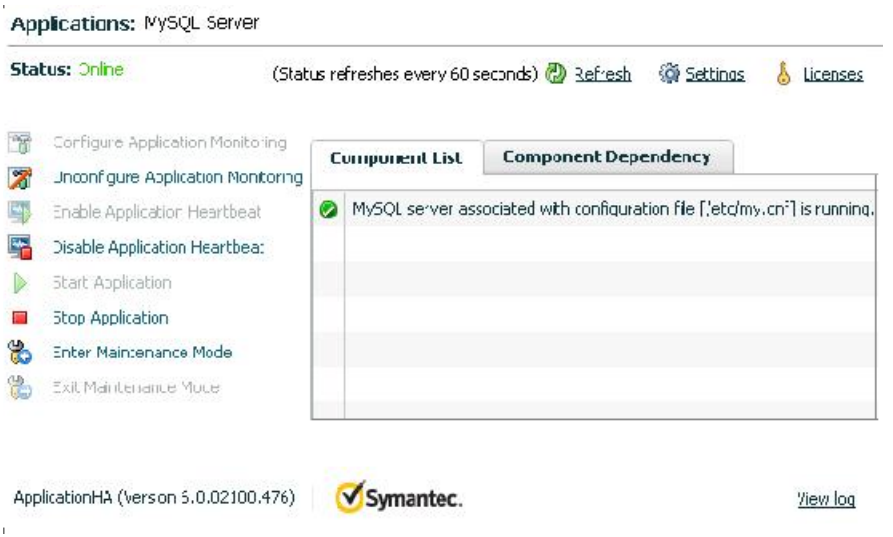
---

9 Click **Finish** to complete the wizard.

This completes the application monitoring configuration.

10 To view the status of the configured application on a virtual machine, on the Veritas Operations Manager Management Server console, right-click the appropriate virtual machine and then click **Manage ApplicationHA**.

The Symantec High Availability view appears.

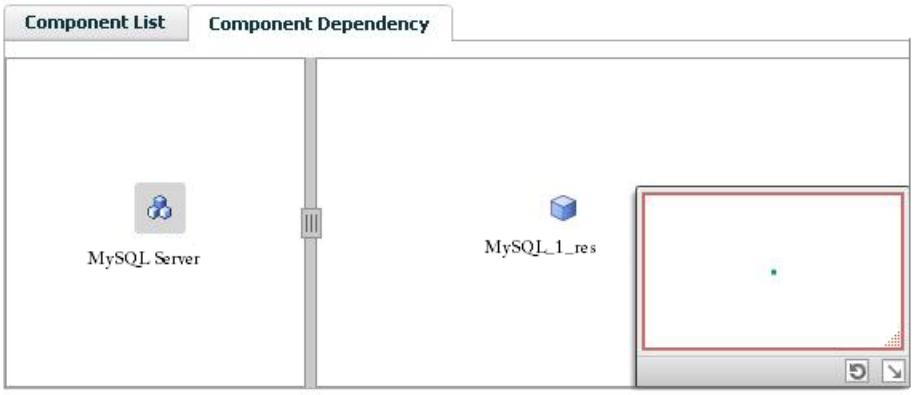


By default, the Component List tab appears. The tab lists each component of the configured application and the status description of each component.

For more information on viewing and administering applications through the Veritas Operations Manager, see the *Symantec ApplicationHA User's Guide*.

11 To view component dependency for the monitored application, click the **Component Dependency** tab.

The component dependency graph appears.



The graph illustrates the dependencies between a selected component group (an application or a group of inter-related components) and its components for the configured application. The left pane displays component groups, configured applications, or both. The right pane displays components of the selected component group or application.

For more information on viewing component dependency for any configured application, see the *Symantec ApplicationHA User's Guide*.

Chapter 4

Troubleshooting the agent for MySQL

This chapter includes the following topics:

- Starting the MySQL instance outside the Symantec ApplicationHA environment
- Reviewing error log files

Starting the MySQL instance outside the Symantec ApplicationHA environment

If you face problems while working with a resource, you must disable the resource within the cluster framework. A disabled resource is not under the control of the cluster framework, and so you can test the MySQL database server independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

Note: Use the same parameters that are defined for the resource attributes within the cluster framework while restarting the resource outside the framework, like the owner of the application, the environment file, and so on.

- Starting the MySQL server
  To start the MySQL server outside cluster, execute the following command:

  ```
  $ BaseDir/bin/mysqld_safe --defaults-file=MyCnf \
  --datadir=DataDir --user=MySQLUser &
  ```

- Stopping the MySQL server
  To stop the MySQL server outside cluster, execute the following command:

  ```
  $ BaseDir/bin/mysqladmin --user=MySQLAdmin \
  ```

```
--password=MySQLAdminPasswd shutdown
```

- Monitoring the MySQL server
  First verify that the MySQL processes are running as MySQLUser.
- The agent uses a connect(3c) method to check for the MySQL server to listen to the port defined by the Port attribute. Execute the following command:
  ```
  $ telnet HostName Port
  ```
- The agent then uses the following monitor command to verify that the MySQL server is up:
  ```
  $ BaseDir/bin/mysqladmin --user=MySQLAdmin --password=XXXXXX status
  Uptime: 2221700 Threads: 1 Questions: 35 Slow queries: 0 Opens:
  28 Flush tables: 1 Open tables: 4 Queries per second avg: 0.000
  Troubleshooting the agent for MySQL
  Verifying virtualization
  40
  $ echo $?
  0
  ```
  Here XXXXXX is the password for the MySQLAdmin database user. The command is executed in the context of the MySQLUser.

Reviewing error log files

If you face problems while using MySQL or the agent for MySQL, use the log files described in this section to investigate the problems. The common reasons for issues are as follows:

Insufficient privileges	Files that need to be created or written to may be created as MySQLUser. Verify if necessary privileges have been set.
Incorrect port, environment, or parameter settings	Verify that ports have been properly configured and declared. Typically, ports from 1 through 1024 are reserved for the superuser. Ensure that parameters to the agent are correctly defined.
Expired licenses	Check the application log files for any error messages related to expired licenses. Ensure that the license keys or files have been placed at the appropriate location, as needed by the application.

Broken symlinks, missing files, and libraries	Verify your installation. Make sure that nothing is broken, and all dependencies for the executables are met.
Insufficient disk space or system parameters	Ensure that the file-system has sufficient space for creation of temporary files that the application might need. Verify that the kernel has been tuned for sufficient IPC resources, file descriptors, and meets the hardware requirement. Consult your product documentation for these details.

Using MySQL log files

By default the MySQL agent writes error logs at `DataDir/HostName.err`, where HostName is the hostname of the node where the database is currently hosted.

Reviewing ApplicationHA log files

In case of problems while using the agent for MySQL, you can access the ApplicationHA log files at the following location:

`/var/VRTSvcs/log/MySQL_A.log`

Reviewing cluster log files

In case of problems while using the agent forMySQL, you can also access the engine log file for more information about a particular resource. The engine log files are located at the following location:

The VCS engine log file is /var/VRTSvcs/log/engine_A.log.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in an ApplicationHA log file for each MySQL resource. You can set this attribute to TRACE, which enables very detailed and verbose logging.

If you set ResLogLevel to TRACE, a very high volume of messages are produced. Symantec recommends that you localize the ResLogLevel attribute for a particular resource.

Note: The LogDbg attribute should be used to enable the debug logs for the ACCLib-based agents when the ACCLIB version is 6.2.0.0 or later and the VCS version is 6.2 or later.

To localize ResLogLevel attribute for a resource

1. Identify the MySQL resource for which you want to enable detailed logging.
2. Localize the ResLogLevel attribute for the identified resource:

```
# /opt/VRTS/bin/hares -local MySQL_<Instance_Number>_res \
 ResLogLevel
```

3. Set the ResLogLevel attribute to TRACE for the identified resource:

```
# /opt/VRTS/bin/hares -modify MySQL_<Instance_Number>_res \
 ResLogLevel TRACE -sys System_Name
```

4. Note the time before you begin to operate the identified resource.
5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.
7. Set the ResLogLevel attribute back to INFO for the identified resource:

```
# /opt/VRTS/bin/hares -modify MySQL_<Instance_Number>_res \
 ResLogLevel INFO -sys System_Name
```

8. Review the contents of the engine log file. Use the time noted in Step 4 and Step 6 to diagnose the problem.

Appendix A

Resource type definitions

This appendix includes the following topics:

- About the resource type and attribute definitions
- Resource type definition for MySQL agent

About the resource type and attribute definitions

The resource type represents the configuration definition of the agent and specifies how the agent is defined in the configuration file. The attribute definitions describe the attributes associated with the agent. The required attributes describe the attributes that must be configured for the agent to function.

Resource type definition for MySQL agent

Examples of agent type definition files are:

```
type MySQL (
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/MySQL"
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static int ToleranceLimit = 1
        static str ArgList[] = { ResLogLevel, State, IState,
        MySQLUser, MySQLAdmin, MySQLAdminPasswd, EnvFile,
        BaseDir, DataDir, MyCnf, HostName, Port,
        SecondLevelMonitor, MonitorProgram }
        static boolean AEPTimeout = 1
        str ResLogLevel = INFO
        str MySQLUser = mysql
        str MySQLAdmin = root
        str MySQLAdminPasswd
        str EnvFile
```

```
        str BaseDir
        str DataDir
        str MyCnf
        str HostName
        int Port = 3306
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

Attribute definition for MySQL agent

Refer to the following required and optional attributes while configuring the agent for MySQL.

Table A-1 Required attributes

Attribute	Description
ResLogLevel	Specifies the logging detail that the agent performs for the resource. The valid values are as follows: ■ ERROR: Only logs error messages. ■ WARN: Logs above plus warning messages. ■ INFO: Logs above plus informational messages ■ TRACE: Logs above plus trace messages. TRACE is very verbose and should be used only during initial configuration or for troubleshooting and diagnostic operations. Default Value: INFO Example: INFO **Note:** The use of the ResLogLevel attribute is deprecated from VCS version 6.2 onwards. You must use the LogDbg attribute instead of the ResLogLevel attribute to enable debug logs for the ACCLib-based agents, when the ACCLib version is 6.2.0.0 or later. The agent captures the first failure data of the unexpected events and automatically logs debug messages in their respective agent log files.

Table A-1 Required attributes *(continued)*

Attribute	Description
MySQLAdmin	The administrative database user of the MySQL server with privileges to shutdown the database. Symantec recommends creating a dedicated account in the database, with shutdown privileges only. Review the information for adding an administrative user for shutdown purposes. Default Value: root Example: admin
MySQLAdminPasswd	Password for the database administrator specified in the MySQLAdmin attribute. The password is encrypted using the VCS encrypt utility, vcsencrypt(1m). **Note:** You need not encrypt the password if you are using the VCS GUI to enter the password. VCS GUI automatically encrypts the password. Default Value: "" Example : jxmXkvVvkVnvWvsVx
MySQLUser	The dedicated OS login created while installing the MySQL server. The database server will be started as this user. This login has to be identical on all failover nodes. Default Value: mysql Example: mysql
DataDir	The absolute path to the directory storing the database being managed by this instance of the server. Symantec recommends storing this directory on shared storage so that the same copy is available on the failover node. The database directory should be owned by the user specified by the MySQLUser agent attribute. Default Value: "" Example: /var/lib/mysql
BaseDir	The installation path of the MySQL Database server. Default Value: "" Example: /usr/local/MySQL

Table A-2 Optional attributes

Attribute	Description
EnvFile	Complete path of file name to source to set the environment prior to executing MySQL programs. Symantec recommends storing the file on the shared disk where the database directory (DataDir) is located. This ensures that the same file is available on each failover node. Specifying this attribute is optional. The shell environments supported are ksh, sh, and csh. Default Value: "" Example: /db/bbmas/envfile
HostName	Virtual host name for this MySQL Database instance. The monitor agent function uses this attribute to determine if the server is responding to client requests. This attribute is required only if second level monitoring is enabled. Default Value: "" Example: mysql.sym.com
Port	Represents the port number dedicated to the MySQL server. The monitor agent function uses this value to determine if the server responds to client requests. This attribute is required only if second level monitoring is enabled. Default Value: 3306 Example: 3306
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. For information about setting this attribute: Default Value: "" Example 1.: /db/bbmas/myMonitor.pl Example 2.: /db/bbmas/myMonitor.sh arg1 arg2

Table A-2 Optional attributes *(continued)*

Attribute	Description
SecondLevelMonitor	Used to enable second-level monitoring and specify how often it is run. Second-level monitoring is a deeper, more thorough state check of the configured MySQL instance. The numeric value specifies how often that the second-level monitoring routines are run. Care should be taken when setting this attribute to large numbers. For example, if the MonitorInterval is set to 60 seconds, and the SecondLevelMonitor is set to 100, then the second level check would only get performed every 100 minutes, which may not be as often as intended. To provide maximum flexibility, the value set is not checked for an upper limit. You can set the second level check to occur once a month, if that is desired. Default Value: 0 Example: 1
MyCnf	Complete path to the MySQL configuration file to be used while starting the database. Symantec recommends storing the file on the shared disk where the database directory (DataDir) is located. This ensures that the same file is available on each failover node. Default Value: "" Example: /etc/my.cnf
LevelTwoMonitorFreq	Specifies the frequency at which the agent must perform second-level or detailed monitoring. You can also override the value of this attribute at the resource level. The value indicates the number of monitor cycles after which the agent will monitor MySQL in detail. For example, the value 5 indicates that the agent will monitor MySQL in detail after every five online monitor intervals. Type and dimension: integer-scalar Default: 0

Table A-2 Optional attributes *(continued)*

Attribute	Description
LogDbg	For ACCLib-based agents, you must use the LogDbg resource type attribute to enable the debug logs when the ACCLib version is 6.2.0.0 or later. Set the LogDbg attribute to DBG_5 to enable debug logs for ACCLIB based agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all the agent's resources in the cluster. If debug logs must be enabled for a specific resource, override the LogDbg attribute. Type and dimension: string-keylist Default: {} (none) For more information on how to use the LogDbg attribute, refer to the *Symantec Cluster Server Administrator's Guide*.

Appendix B

Detail monitoring

This appendix includes the following topics:

- Setting the PATH variable
- Setting up detail monitoring for ApplicationHA agent for MySQL Server

Setting the PATH variable

ApplicationHA commands reside in the `/opt/VRTS/bin` directory. Add this directory to your PATH environment variable.

To set the PATH variable

- Perform one of the following steps:

  For the Bourne Shell (sh or ksh), type:

  ```
  $ PATH=/opt/VRTS/bin:$PATH; export PATH
  ```

  For the C Shell (csh or tcsh), type:

  ```
  $ setenv PATH :/opt/VRTS/bin:$PATH
  ```

Setting up detail monitoring for ApplicationHA agent for MySQL Server

This section describes the procedure to enable and disable detail monitoring for MySQL.

To enable detail monitoring for MySQL

1. Make the ApplicationHA configuration writable:

   ```
   # haconf -makerw
   ```

2. Freeze the service group to avoid automated actions by ApplicationHA in case of an incomplete configuration:

   ```
   # hagrp -freeze MySQL_<Instance_Number>_SG
   ```

3. Enable detail monitoring for MySQL resources by using the following ApplicationHA commands:

   ```
   # hares -modify MySQL_<Instance_Number>_res LevelTwoMonitorFreq
   <frequency>
   ```

   Note: For more information on LevelTwoMonitorFreq attribute:

4. Save the configuration and unfreeze the service group.

   ```
   # hagrp -unfreeze MySQL_<Instance_Number>_SG
   ```

   ```
   # haconf -dump -makero
   ```

To disable detail monitoring for MySQL

1. Make the ApplicationHA configuration writable:

   ```
   # haconf -makerw
   ```

2. Freeze the service group to avoid automated actions by ApplicationHA in case of an incomplete configuration:

   ```
   # hagrp -freeze MySQL_<Instance_Number>_SG
   ```

3. Disable detail monitoring for MySQL resources by using the following ApplicationHA commands:

   ```
   # hares -modify MySQL_<Instance_Number>_res LevelTwoMonitorFreq
   0
   ```

4. Save the configuration and unfreeze the service group.

   ```
   # hagrp -unfreeze MySQL_<Instance_Number>_SG
   ```

   ```
   # haconf -dump -makero
   ```