

Cluster Server Agent for WebSphere MQ Queue Manager WebSphere MQ Installation and Configuration Guide

AIX, Linux, Solaris

8.0.2

Veritas InfoScale™ Availability Agents

Last updated: 2024-01-10

Legal Notice

Copyright © 2024 Veritas Technologies LLC. All rights reserved.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

This product may contain third-party software for which Veritas is required to provide attribution to the third party ("Third Party Programs"). Some of the Third-Party Programs are available under open source or free software licenses. The License Agreement accompanying the Software does not alter any rights or obligations you may have under those open source or free software licenses. Refer to the third-party legal notices document accompanying this Veritas product or available at:

<https://www.veritas.com/about/legal/license-agreements>

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 Veritas Technologies LLC 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. VERITAS TECHNOLOGIES LLC 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, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas 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.

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054

<http://www.veritas.com>

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

<https://www.veritas.com/support>

You can manage your Veritas account information at the following URL:

<https://my.veritas.com>

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)

CustomerCare@veritas.com

Japan

CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

<https://sort.veritas.com/documents>

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

<http://www.veritas.com/community/>

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Chapter 1	Introducing the agent for WebSphere MQ	8
	About the Cluster Server agent for WebSphere MQ	8
	Supported software	9
	How the agent supports intelligent resource monitoring	9
	WebSphere MQ agent functions	9
	Online	9
	Offline	10
	Monitor	11
	Clean	12
 Chapter 2	 Installing, upgrading, and removing the agent for WebSphere MQ	 14
	Before you install the Cluster Server agent for WebSphere MQ	14
	Prerequisites for enabling i18n support	15
	About the ACC library	15
	Installing the ACC library	16
	Installing the ACC library IPS package on Oracle Solaris 11 systems	17
	Installing the ACC library package on Solaris brand non-global zones	18
	Installing the agent in a VCS environment	19
	Installing the agent manually	19
	Removing the agent in a VCS environment	23
	Removing the agent manually	23
	Removing the ACC library	24
	Upgrading the agent in a VCS environment	25
 Chapter 3	 Configuring the agent for WebSphere MQ	 27
	About configuring the Cluster Server agent for WebSphere MQ	27
	Importing the agent types file in a VCS environment	27
	Importing the agent types file using Cluster Manager (Java Console)	28
	Importing the agent types file using the CLI	28
	WebSphere MQ agent attributes	29

Chapter 4

Executing a customized monitoring program	39
Configuring the service group for WebSphere MQ using the Veritas High Availability Configuration wizard	40
Installing the agent for WebSphere MQ	41
Typical VCS cluster configuration in a virtual environment	41
About configuring application monitoring using the Veritas High Availability solution for VMware	42
Getting ready to configure VCS service groups using the wizard	43
Before configuring application monitoring	44
Launching the High Availability Configuration wizard	45
Configuring the WebSphere MQ queue manager for high availability	47
Understanding service group configurations	52
Resource dependency	52
Service group dependency	53
Infrastructure service groups	53
Understanding configuration scenarios	53
Configuring a single instance/multiple instances in VCS	54
Configuring multiple WebSphere MQ Queue Manager instances in VCS using multiple runs of the wizard	54
Configuring multiple applications	55
Veritas High Availability Configuration wizard limitations	55
Troubleshooting	56
Veritas High Availability Configuration wizard displays blank panels	56
The Veritas High Availability Configuration wizard displays the "hadiscover is not recognized as an internal or external command" error	56
Running the 'hastop -all' command detaches virtual disks	56
Log files	57
Sample configurations	57
Sample VCS configuration file for single WebSphere MQ Queue Manager instance (VxVM)	57
Sample VCS configuration file for single WebSphere MQ Queue Manager instance (LVM)	62
Sample Configuration files at Primary and Secondary cluster for IBM WebSphereMQ - RDQM HA/DR	67

Chapter 5	Enabling the agent for WebSphere MQ to support IMF	85
	About Intelligent Monitoring Framework	85
	Benefits of IMF	86
	Agent functions for the IMF functionality	86
	imf_init	86
	imf_getnotification	86
	imf_register	87
	Attributes that enable IMF	87
	IMF	87
	IMFRegList	88
	Before you enable the agent to support IMF	88
	Enabling the agent to support IMF	89
	If VCS is in a running state	89
	If VCS is not in a running state	91
	Disabling intelligent resource monitoring	92
	Sample IMF configurations	92
Chapter 6	Configuring the service groups for WebSphere MQ using the CLI	94
	Before configuring the service groups for WebSphere MQ	94
	Configuring service groups for WebSphere MQ queue managers	95
	Configuring a WebSphere MQ resource	95
	Configuring a WebSphere MQ listener	96
	Creating service groups for WebSphere MQ under Solaris non-global zones	97
Chapter 7	Troubleshooting the agent for WebSphere MQ	99
	Using the correct software and operating system versions	99
	Meeting prerequisites	100
	Configuring WebSphere MQ Queue Manager resources	100
	Starting the WebSphere MQ Queue Manager instance outside a cluster	100
	Monitoring WebSphere MQ queue manager processes	101
	Stopping WebSphere MQ queue manager processes forcefully	102
	Reviewing error log files	104
	Using WebSphere MQ log files	104
	Reviewing cluster log files	104
	Using trace level logging	104
	Troubleshooting the configuration for IMF	105

	Known issues	108
Appendix A	Sample Configurations	109
	About sample configurations for the agents for WebSphere MQ	109
	Sample agent type definition for WebSphere MQ	109
	Sample configuration in a VCS environment	110
	Sample configuration in a VCS virtual environment with VMware	112
	Sample configuration for Multi-Instance mode	121
	Sample service group configurations	123

Introducing the agent for WebSphere MQ

This chapter includes the following topics:

- [About the Cluster Server agent for WebSphere MQ](#)
- [Supported software](#)
- [How the agent supports intelligent resource monitoring](#)
- [WebSphere MQ agent functions](#)

About the Cluster Server agent for WebSphere MQ

Cluster Server (VCS) agents monitor specific resources within an enterprise application. They determine the status of resources and start or stop them according to external events.

The Cluster Server agent for WebSphere MQ provides high availability for all WebSphere MQ Queue Managers in a cluster. The agent can bring a specific WebSphere MQ Queue Manager online and monitor the state of the Queue Manager. The agent can also detect failures and shut down the Queue Manager in case of a failure.

For the latest updates or software issues for this agent, see the *Cluster Server Agent Pack Release Notes*.

Supported software

For information on the software versions that the Cluster Server agent for WebSphere MQ supports, see the Veritas Services and Operations Readiness Tools (SORT) site: <https://sort.veritas.com/agents>.

How the agent supports intelligent resource monitoring

With Intelligent Monitoring Framework (IMF), VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring.

When an IMF-enabled agent starts up, the agent initializes the Asynchronous Monitoring Framework (AMF) kernel driver. After the resource is in a steady state, the agent registers with the AMF kernel driver, the details of the resource that are required to monitor the resource.

For example, the agent for WebSphere MQ registers the PIDs of the WebSphere MQ Queue Manager processes with the AMF kernel driver.

The `imf_getnotification` function of the agent waits for any resource state changes. When the AMF kernel driver module notifies the `imf_getnotification` function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS, which then takes appropriate action.

For more information, see the *Cluster Server Administrator's Guide*.

WebSphere MQ agent functions

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

Online

The online function involves the following tasks:

- Verifies that the WebSphere MQ instance is not already online. If the instance is online, the online operation exits immediately.
- If the partial set of WebSphere MQ processes are running, the agent performs a process level clean up before starting the queue manager.

- Uses an IBM provided start script to start the WebSphere MQ using the StartOptions and the name of the queue manager.
You can also configure the online function to source a shell script or a program that the EnvFile attribute specifies. This script or program ensures that the required shell environment variables are properly set before executing the start script.
- Ensures that the WebSphere MQ queue manager is up and running successfully. The operation uses the wait period that the OnlineTimeout attribute specifies to enable the queue manager to initialize fully before allowing the monitor function to probe the resource.
- If the MQInstallationPath attribute is configured, the agent runs WebSphere MQ commands from the specified WebSphere MQ installation path.
- Executes the service file that is specified in the ServiceName attribute if SystemD is supported. If that attribute value is empty, the agent executes the default service file.

In a multi-instance configuration, the online function can be performed at the service group level to start both, the Active instance and the Standby instance.

Offline

The offline function involves the following tasks:

- Verifies that the WebSphere MQ instance is not already offline. If the instance is offline, the offline operation exits immediately.
- Depending on whether the -s option is passed in the StopOptions attribute, takes the following action:

Is the -s option passed?	Is the Standby instance running?	What does the agent do?
Yes	No	Eliminates the option from the attribute
No	Yes	Issues an offline command to the standby instance
Yes	Yes	Sleeps for 30 seconds and checks the status of the standby instance again

- Uses an IBM-provided stop script to stop the WebSphere MQ queue manager using the name of the queue manager and the StopOptions attribute for an active instance.

If the instance is in an active mode and the `-s` option is not passed in the `StopOptions` attribute, the agent first runs the offline command to the standby instance and then proceeds with the offline operation on the active instance. You can also configure the offline function to source a shell script or a program that the `EnvFile` attribute specifies. This script or program ensures that the required shell environment variables are properly set before executing the stop script.

- Ensures that the WebSphere MQ queue manager is given enough time to go offline successfully. The operation uses a wait period that the `OfflineTimeout` attribute specifies, to allow the WebSphere MQ queue manager to complete the offline sequence before allowing further probing of the resource. If the processes are found running even after the wait period, then these processes are killed.
- If the `MQInstallationPath` attribute is configured, the agent runs WebSphere MQ commands from the specified WebSphere MQ installation path.

In a multi-instance configuration, the offline function can be performed at the service group level to stop both, the Active instance and the Standby instance.

Monitor

The monitor function monitors the states of the WebSphere MQ queue managers running on all nodes within the cluster.

The monitor function can monitor the following WebSphereMQ queue manager components:

- Queue manager
- Channel Initiator
- Command Server (if the `CommandServer` attribute is set to 1)

The function performs the following tasks:

- The first-level check searches for all system processes that must be running for a WebSphere MQ queue manager instance. If the first-level check does not find these processes running on the node, the check exits immediately, and reports the queue manager as offline. For a multi-instance configuration, the agent runs the `dspmq` command to detect the Active and the Standby modes of configuration. If the output of `dspmq` command is **Starting**, it means that the queue manager is in transition. In this case, the agent reports the UNKNOWN state. The agent performs the check for a certain number of monitor cycles, which are specified in the `TransitionWaitLimit` attribute. After these many monitor cycles have lapsed, if the output of the `dspmq` command is still **Starting**, the agent calls the Clean function.

The agent for WebSphere MQ also supports Intelligent Monitoring Framework (IMF) in the first level check. IMF enables intelligent resource monitoring. The agent for WebSphere MQ is IMF-aware and uses the asynchronous monitoring framework (AMF) kernel driver for resource state change notifications. See [“How the agent supports intelligent resource monitoring”](#) on page 9.

You can use the MonitorFreq key of the IMF attribute to specify the frequency at which the agent invokes the monitor function. See [“MonitorFreq”](#) on page 88.

- The second-level check, if configured, determines the status of the WebSphereMQ queue manager.

For an active instance, the second-level check runs the `runmqsc` command and pings the queue manager to see if the manager is up and running. This check ensures that the processes are truly available for MQ Queue processing.

For a standby instance, the second-level check runs the `dspmq` command to check if the standby instance is running.

Note: The attribute used to configure the second level check and its frequency depends on the software versions of VCS and WebSphere MQ agent you have installed: For VCS 5.1 SP1 with WebSphereMQ agent version 5.1.9.0, use the LevelTwoMonitorFreq attribute. For VCS 5.1 or earlier with WebSphereMQ agent 5.1.8.0 or earlier, use the SecondLevelMonitor attribute

- Depending upon the MonitorProgram attribute, the monitor function can perform a customized check using a user-supplied monitoring utility. For details about executing a custom monitor program:
See [“Executing a customized monitoring program”](#) on page 39.
- When the WebSphere MQ resource is offline and the agent detects the queue manager processes as running, but the second level monitor check fails, the agent cleans these processes.
- If the MQInstallationPath attribute is configured, the agent runs WebSphere MQ commands from the specified WebSphere MQ installation path.

Clean

In case of a failure or after an unsuccessful attempt to online or offline WebSphere MQ queue manager, the clean function removes any queue manager processes remaining in the system.

The function performs the following tasks:

- Attempts to gracefully shut down the WebSphere MQ queue manager.

- If a graceful shutdown fails, the clean function looks for all the processes running for the WebSphere MQ queue manager, and cleans the processes.
- The clean function executes the IBM supplied utility, amqcln to clean the IPC resources that are associated with the WebSphere MQ queue manager.
- If the CommandServer attribute is set to 1 for WebSphere MQ version 6.0 or later, the clean function kills the Command Server processes associated with the WebSphere MQ queue manager.
- If the MQInstallationPath attribute is configured, the agent runs WebSphere MQ commands from the specified WebSphere MQ installation path.

Note: For information about the additional functions of the agent for WebSphere MQ when IMF is enabled: See [“Agent functions for the IMF functionality”](#) on page 86.

Installing, upgrading, and removing the agent for WebSphere MQ

This chapter includes the following topics:

- [Before you install the Cluster Server agent for WebSphere MQ](#)
- [About the ACC library](#)
- [Installing the ACC library](#)
- [Installing the agent in a VCS environment](#)
- [Removing the agent in a VCS environment](#)
- [Removing the ACC library](#)
- [Upgrading the agent in a VCS environment](#)

Before you install the Cluster Server agent for WebSphere MQ

You must install the Cluster Server agent for WebSphere MQ on all the systems that will host WebSphere MQ Queue Manager service groups.

Before you install the agent for WebSphere MQ, ensure that the following prerequisites are met.

For VCS, do the following:

- Install and configure Cluster Server.

For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.

- Remove any previous version of this agent.
- Install the latest version of ACC Library.
To install or update the ACC Library package, locate the library and related documentation in the Agent Pack tarball.
See [“About the ACC library”](#) on page 15.

Note: All non-global zones must be booted and in a running state at the time of package installation or un-installation. If the non-global zones are not booted, you may need to reinstall the package manually after booting the non-global zones.

Prerequisites for enabling i18n support

Perform the following steps to enable i18n support to the agent:

- Install ACCLib version 5.1.2.0 or later.
See [“Installing the ACC library”](#) on page 16.
- For VCS 5.0 and earlier releases, copy the latest `ag_i18n_inc.pm` module from the following location on the agent pack disc.

Note: Review the `readme.txt` for instructions to copy this module.

VCS 5.0	<code>cd1/platform/arch_dist/vcs/application/i18n_support/5.0</code>
VCS 4.1	<code>cd1/platform/arch_dist/vcs/application/i18n_support/4.1</code>
VCS 4.0	<code>cd1/platform/arch_dist/vcs/application/i18n_support/4.0</code>

where *arch_dist* takes the following values:

'sol_sparc' for Solaris SPARC
'generic' for Linux

Note: *arch_dist* is not applicable to AIX.

About the ACC library

The operations of a Cluster Server agent depend on a set of Perl modules known as the ACC library. The library must be installed on each system in the cluster that

runs the agent. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

Instructions to install or remove the ACC library on a single system in the cluster are given in the following sections. The instructions assume that the ACCLib tar file has already been extracted.

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

Installing the ACC library

Install the ACC library on each system in the cluster that runs an agent that depends on the ACC library.

To install the ACC library

- 1 Log in as a superuser.
- 2 Download ACC Library.

You can download either the complete Agent Pack tar file or the individual ACCLib tar file from the Veritas Services and Operations Readiness Tools (SORT) site (<https://sort.veritas.com/agents>).

- 3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX	<code>cd1/aix/vcs/application/acc_library/version_library/pkg</code>
Linux	<code>cd1/linux/generic/vcs/application/acc_library/version_library/rpms</code>
Solaris	<code>cd1/solaris/dist_arch/vcs/application/acc_library/version_library/pkg</code>

where *dist_arch* is *sol_sparc*.

- 4 If you downloaded the individual ACCLib tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).
- 5 Install the package. Enter **Yes**, if asked to confirm overwriting of files in the existing package.

AIX # installp -ac -d VRTSacclib.bff VRTSacclib

Linux # rpm -i \
 VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm

Solaris # pkgadd -d VRTSacclib.pkg

See [“Installing the ACC library IPS package on Oracle Solaris 11 systems”](#) on page 17.

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

Installing the ACC library IPS package on Oracle Solaris 11 systems

Install the ACC library IPS package on an Oracle Solaris 11 system.

To install the ACC library IPS package on Oracle Solaris 11 systems

- 1 Copy the VRTSacclib.p5p package from the pkgs directory to the system in the /tmp/install directory.

- 2 Disable the publishers that are not reachable as package install may fail, if any, of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

- 3 Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSacclib.p5p Veritas
```

- 4 Install the package.

```
# pkg install --accept VRTSacclib
```

- 5 Remove the publisher from the system.

```
# pkg unset-publisher Veritas
```

- 6 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher name>
```

Installing the ACC library package on Solaris brand non-global zones

With Oracle Solaris 11, you must install the agent package inside non-global zones. The native non-global zones are called Solaris brand zones.

To install the ACC library package on Solaris brand non-global zones

- 1 Ensure that the SMF services,
`svc:/application/pkg/system-repository:default` and
`svc:/application/pkg/zones-proxyd:default`, are online on the global zone.

```
# svcctl svc:/application/pkg/system-repository:default
# svcctl svc:/application/pkg/zones-proxyd:default
```
- 2 Log on to the non-global zone as a superuser.
- 3 Ensure that the SMF service
`svc:/application/pkg/zones-proxy-client:default` is online inside non-global zone:

```
# svcctl svc:/application/pkg/zones-proxy-client:default
```
- 4 Copy the `VRTSacclib.p5p` package from the `pkgs` directory to the non-global zone (for example, at the `/tmp/install` directory).
- 5 Disable the publishers that are not reachable, as package install may fail, if any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```
- 6 Add a file-based repository in the non-global zone.

```
# pkg set-publisher -g/tmp/install/VRTSacclib.p5p Veritas
```
- 7 Install the package.

```
# pkg install --accept VRTSacclib
```
- 8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Veritas
```
- 9 Clear the state of the SMF service, as setting the file-based repository causes the SMF service `svc:/application/pkg/system-repository:default` to go into the maintenance state.

```
# svcadm clear svc:/application/pkg/system-repository:default
```
- 10 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher>
```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent in a VCS environment

Depending upon the operating system and VCS version installed on the cluster nodes, you can install the agent using the following methods:

Table 2-1 Agent installation methods in a VCS environment

Method	VCS version	Operating system
Manual installation using native commands See “Installing the agent manually” on page 19.	4.x, 5.x, and 6.x	AIX, Linux, Solaris

Note: On successful installation of the WebSphere MQ agent version 5.1.14.0 or later, the agent runs the `/opt/VRTSagents/ha/bin/WebSphereMQ/MQ_update.pl` script. This script changes any existing resources of type WebSphereMQ6 to type WebSphereMQ in the VCS configuration, and deletes the WebSphereMQ6 agent type.

Installing the agent manually

Install the agent for WebSphere MQ on each node in the cluster.

To install the agent manually in a VCS environment

- 1 Download the agent from the Veritas Services and Operations Readiness Tools (SORT) site: <https://sort.veritas.com/agents>.

You can download either the complete Agent Pack tar file or an individual agent tar file.

- 2 Uncompress the file to a temporary location, say `/tmp`.

- 3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

```
AIX      cd1/aix/vcs/application/webspheremq_agent/  
        vcs_version/version_agent/pkggs  
  
Linux    cd1/linux/generic/vcs/application/webspheremq_agent/  
        vcs_version/version_agent/rpms  
  
Solaris  cd1/solaris/dist_arch/vcs/application/webspheremq_agent/  
        vcs_version/version_agent/pkggs  
        where, dist_arch is sol_sparc
```

If you downloaded the individual agent tar file, navigate to the pkggs directory (for AIX and Solaris), or rpms directory (for Linux).

- 4 Log in as superuser.
- 5 Install the package.

```
AIX      # installp -ac -d VRTSmq6.rte.bff VRTSmq6.rte  
  
Linux    # rpm -ihv \  
        VRTSmq6-AgentVersion-GA_GENERIC.noarch.rpm  
  
Solaris  # pkgadd -d . VRTSmq6
```

Note: The agent package VRTSmq6 includes the Cluster Server agents for WebSphere MQ and WebSphere MQ FTE. So, the following procedure to remove the agent for WebSphere MQ removes the agent for WebSphere MQ FTE also. However, the agent for WebSphere MQ FTE does not support the Veritas High Availability Configuration wizard.

Installing the agent IPS package on Oracle Solaris 11 systems

To install the agent IPS package on an Oracle Solaris 11 system

- 1 Copy the VRTSmq6.p5p package from the `pkgs` directory to the system in the `/tmp/install` directory.

- 2 Disable the publishers that are not reachable as package install may fail, if any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```

where the publisher name is obtained using the `pkg publisher` command.

- 3 Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTSmq6.p5p Veritas
```

- 4 Install the package.

```
# pkg install --accept VRTSmq6
```

- 5 Remove the publisher from the system.

```
# pkg unset-publisher Veritas
```

- 6 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher name>
```

Installing agent packages on Solaris brand non-global zones

To install the agent package on Solaris brand non-global zones

- 1 Ensure that the SMF services, `svc:/application/pkg/system-repository:default` and `svc:/application/pkg/zones-proxyd:default`, are online on the global zone.

```
# svcs svc:/application/pkg/system-repository:default
```

```
# svcs svc:/application/pkg/zones-proxyd:default
```

- 2 Log on to the non-global zone as a superuser.

- 3 Ensure that the SMF service `svc:/application/pkg/zones-proxy-client:default` is online inside non-global zone:

```
# svcs svc:/application/pkg/zones-proxy-client:default
```

- 4 Copy the `VRTSsapnw.p5p` package from the `pkgs` directory to the non-global zone (for example, at the `/tmp/install` directory).
- 5 Disable the publishers that are not reachable, as package install may fail, if any of the already added repositories are unreachable.

```
# pkg set-publisher --disable <publisher name>
```
- 6 Add a file-based repository in the non-global zone.

```
# pkg set-publisher -g/tmp/install/VRTSsapnw.p5p Veritas
```
- 7 Install the package.

```
# pkg install --accept VRTSsapnw
```
- 8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Veritas
```
- 9 Clear the state of the SMF service, as setting the file-based repository causes the SMF service `svc:/application/pkg/system-repository:default` to go into the maintenance state.

```
# svcadm clear svc:/application/pkg/system-repository:default
```
- 10 Enable the publishers that were disabled earlier.

```
# pkg set-publisher --enable <publisher>
```

Note: Perform steps 2 through 10 on each non-global zone.

Installing the agent in a Solaris 10 brand zone

To install the WebSphere MQ agent in a brand zone on Solaris 10:

- Ensure that the `ACCLibrary` package, `VRTSaclib`, is installed in the non-global zone.
To install `VRTSaclib` in the non-global zone, run the following command from the global zone:

```
# pkgadd -R /zones/zone1/root -d VRTSaclib.pkg
```
- To install the agent package in the non-global zone, run the following command from the global zone:

```
# pkgadd -R zone-root/root -d . VRTSsapnw
```


For example:

```
# pkgadd -R /zones/zone1/root -d . VRTSsapnw
```

Note: You can ignore the following messages that might appear:

```
## Executing postinstall script.

ln: cannot create
/opt/VRTSagents/ha/bin/WebSphereMQ/imf_getnotification: File exists

ln: cannot create /opt/VRTSagents/ha/bin/WebSphereMQ/imf_register:
File exists

or ## Executing postinstall script.

ln: cannot create
/opt/VRTSagents/ha/bin/WebSphereMQ/imf_getnotification: No such file
or directory

ln: cannot create /opt/VRTSagents/ha/bin/WebSphereMQ/imf_register:
No such file or directory
```

Removing the agent in a VCS environment

Depending upon the operating system and VCS version installed on the cluster nodes, you can uninstall the agent using the following methods:

Table 2-2 Agent uninstallation methods in a VCS environment

Method	VCS version	Operating system
Manual uninstallation using native commands See “Removing the agent manually” on page 23.	4.x, 5.x, and 6.x	AIX, Linux, Solaris

Removing the agent manually

You must uninstall the agent for WebSphere MQ from a cluster while the cluster is active.

Warning: The agent package VRTSmq6 includes the Cluster Server agents for WebSphere MQ and WebSphere MQ FTE. So, the following procedure to remove the agent for WebSphere MQ removes the agent for WebSphere MQ FTE also.

To uninstall the agent in a VCS environment

- 1 Log in as a superuser.
- 2 Set the cluster configuration mode to read/write by typing the following command from any node in the cluster:

```
# haconf -makerw
```

- 3 Remove all WebSphere MQ Queue Manager resources from the cluster. Use the following command to verify that all resources have been removed:

```
# hares -list Type=WebSphereMQ
```

- 4 Remove the agent type from the cluster configuration by typing the following command from any node in the cluster:

```
# hatype -delete WebSphereMQ
```

Removing the agent's type file from the cluster removes the include statement for the agent from the `main.cf` file, but the agent's type file is not removed from the cluster configuration directory. You can remove the agent's type file later from the cluster configuration directory.

- 5 Save these changes. Then set the cluster configuration mode to read-only by typing the following command from any node in the cluster:

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

- 6 Use the platform's native software management program to remove the agent for WebSphere MQ from each node in the cluster.

Execute the following command to uninstall the agent:

```
AIX # installp -u VRTSmq6.rte
```

```
Linux # rpm -e VRTSmq6
```

```
Solaris # pkgrm VRTSmq6
```

Note: To uninstall the agent IPS package on a Solaris 11 system:

```
# pkg uninstall VRTSmq6
```

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

- 1 Ensure that all agents that use ACC library are removed.
- 2 Run the following command to remove the ACC library package:

AIX # installp -u VRTSaccLib

Linux # rpm -e VRTSaccLib

Solaris # pkgrm VRTSaccLib

Note: To uninstall the ACCLib IPS package on a Solaris 11 system, run the following command:

pkg uninstall VRTSaccLib

Upgrading the agent in a VCS environment

The VRTSmq6 package for the WebSphere MQ agent version 5.1.14.0 contains a script named `MQ_update.pl`.

This script does the following:

- Updates the WebSphereMQ6 agent type to WebSphereMQ.
- Reconfigures all the existing resources of the WebSphereMQ6 agent type with the WebSphereMQ agent type.
- Removes the WebSphereMQ6 agent type.

Perform the following steps to upgrade the agent with minimal disruption, in a VCS environment.

To upgrade the agent

- 1 If VCS is not running, proceed to step 2.

If VCS is running, stop the WebSphereMQ and WebSphereMQFTE agents on all the nodes.

- Check if the agent is running.

If the agent is WebSphereMQ

```
# haagent -display WebSphereMQ | grep Running
```

The output resembles:

```
WebSphereMQ Running Yes
```

If the agent is WebSphereMQFTE

```
# haagent -display WebSphereMQFTE | grep Running
```

The output resembles:

```
WebSphereMQFTE Running Yes
```

- If the agent is running, stop the agent.

If the agent is WebSphereMQ

```
# haagent -stop WebSphereMQ -force -sys hostname
```

If the agent is WebSphereMQFTE

```
# haagent -stop WebSphereMQFTE -force -sys hostname
```

2 Upgrade the agent.

- Linux # `rpm -Uvh VRTSmq6-agentVersion-GENERIC.noarch.rpm`

- For AIX and Solaris, uninstall and then install the agent:

See [“Removing the agent in a VCS environment”](#) on page 23.

See [“Installing the agent in a VCS environment”](#) on page 19.

The `MQ_update.pl` script runs automatically on successful installation of the agent.

- When the agent is successfully upgraded to version 6.2.0.0, the following message is displayed:

```
You have successfully replaced agent type [WebSphereMQ6]
with the new agent type [WebSphereMQ]
For details, see the following log: [/var/VRTSvcs/log/
UpdateWebSphereMQType_4251/MQ_update.log]
Installation of <VRTSmq6> was successful.
```

- Import the agent type configuration file.

See [“Importing the agent types file in a VCS environment”](#) on page 27.

3 If the agent is stopped, start the agent.

If agent is WebSphereMQ, use

```
# haagent -start WebSphereMQ -sys hostname
```

If agent is WebSphereMQFTE, use

```
# haagent -start WebSphereMQFTE -sys hostname
```

Configuring the agent for WebSphere MQ

This chapter includes the following topics:

- [About configuring the Cluster Server agent for WebSphere MQ](#)
- [Importing the agent types file in a VCS environment](#)
- [WebSphere MQ agent attributes](#)
- [Executing a customized monitoring program](#)

About configuring the Cluster Server agent for WebSphere MQ

After installing the Cluster Server agent for WebSphere MQ, you must import the agent type configuration file. After importing this file, review the attributes table that describes the resource type and its attributes, and then create and configure WebSphere MQ Queue Manager resources.

To view the sample agent type definition and service groups configuration:

See [“About sample configurations for the agents for WebSphere MQ”](#) on page 109.

Importing the agent types file in a VCS environment

To use the agent for WebSphere MQ, you must import the agent types file into the cluster.

You can import the agent types file using Cluster Manager (Java Console) or using the command line interface.

Importing the agent types file using Cluster Manager (Java Console)

To import the agent types file using Cluster Manager (Java Console)

- 1** Start the Cluster Manager (Java Console) and connect to the cluster on which the agent is installed.
- 2** Click **File > Import Types**.
- 3** In the **Import Types** dialog box, select the following file:

VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample/WebSphereMQ/
	■ Linux	WebSphereMQTypes.cf
	■ Solaris	
VCS 5.x or later	■ AIX	/etc/VRTSagents/ha/conf/WebSphereMQ/
	■ Linux	WebSphereMQTypes.cf
VCS 5.0	Solaris SPARC	/etc/VRTSagents/ha/conf/WebSphereMQ/ WebSphereMQTypes50.cf
VCS 5.1 or later	Solaris SPARC	/etc/VRTSagents/ha/conf/WebSphereMQ/ WebSphereMQTypes51.cf

- 4** Click **Import**.
- 5** Save the VCS configuration.

The WebSphere MQ Queue Manager agent type is now imported to the VCS engine.

You can now create WebSphere MQ Queue Manager resources.

Importing the agent types file using the CLI

To import the agent types file using the command line interface (CLI):

- 1** If VCS is running, run the
`/etc/VRTSagents/ha/conf/WebSphereMQ/WebSphereMQTypes.cmd` file from the command line.
- 2** If VCS is not running, perform the following steps:

1. Copy the agent types file from the
`/etc/VRTSagents/ha/conf/<AgentTypes_file>` directory to the
`/etc/VRTSvcS/conf/config` directory.

Where, `<AgentTypes_file>` is chosen according to the following table:

VCS 4.x	■ AIX	<code>/etc/VRTSvcS/conf/sample_WebSphereMQ/</code>
	■ Linux	<code>WebSphereMQTypes.cf</code>
	■ Solaris	
VCS 5.x or later	■ AIX	<code>/etc/VRTSagents/ha/conf/WebSphereMQ/</code>
	■ Linux	<code>WebSphereMQTypes.cf</code>
VCS 5.0	Solaris SPARC	<code>/etc/VRTSagents/ha/conf/WebSphereMQ/</code>
		<code>WebSphereMQTypes50.cf</code>
VCS 5.1 or later	Solaris SPARC	<code>/etc/VRTSagents/ha/conf/WebSphereMQ/</code>
		<code>WebSphereMQTypes51.cf</code>

2. Include the agent types file in the `main.cf` file.

3. Start HAD.

WebSphere MQ agent attributes

Refer to the required and optional attributes while configuring the agent for WebSphere MQ queue manager.

[Table 3-1](#) shows the required attributes for configuring a WebSphere MQ queue manager.

Table 3-1 Required attributes

Required attributes	Description
CleanIPC	<p>When enabled, the agent runs the <code>amqclien</code> utility to clean up the residual MQ IPC resources. It is not recommended to run <code>amqclien</code>.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>

Table 3-1 Required attributes (*continued*)

Required attributes	Description
CommandServer	<p>Decides whether the monitor function must monitor the command server process. This attribute is applicable to WebSphere 6.0 and later.</p> <p>If this attribute is set to 1, the agent for WebSphere MQ monitors the command server process, <code>amqpcsea</code>. If this process faults, the agent for WebSphere MQ restarts the process.</p> <p>If you set this attribute to 0, the agent for WebSphere MQ does not monitor the <code>amqpcsea</code> process.</p> <p>Type and dimension: boolean-scalar</p> <p>Default: 0</p> <p>Example: 1</p>
MQUser	<p>UNIX user name of the owner of the WebSphere MQ directories and executables. The agent functions use this name to execute all WebSphere MQ commands. This user name also owns the WebSphere MQ processes.</p> <p>This user name does not have to be unique within a cluster. The login shell for this user must be Bourne, Korn, or C-shell.</p> <p>Type and dimension: string-scalar</p> <p>Default: mqm</p> <p>Example: mqusr1</p>
MQVer	<p>Version of the WebSphere MQ queue manager. Valid values are 5.3, 6.0, 7.0, 7.1, 7.5, and 8.x, and 9.x, where x is a supported minor version.</p> <p>Type and dimension: string-scalar</p> <p>Default: 6.0</p> <p>Example: 9.1</p>

Table 3-1 Required attributes (*continued*)

Required attributes	Description
QueueManager	<p>Name of the WebSphere MQ queue manager that the cluster server manages.</p> <p>You must uniquely define this attribute for each queue manager within the cluster. This attribute also uniquely identifies the processes running for a specific WebSphere MQ queue manager.</p> <p>Type and dimension: string-scalar</p> <p>Default: No default value</p> <p>Example: venus.queue.manager</p>
TransitionWaitLimit	<p>Number of monitor cycles that the agent waits while the queue manager is in the Starting state. During this time, the agent reports the status of the resource as UNKNOWN. After the wait limit defined by this attribute has lapsed, if the transition is not complete, the agent faults the resource.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 10</p> <p>Example: 15</p>

Table 3-1 Required attributes (*continued*)

Required attributes	Description
ResLogLevel	<p>The logging detail performed by the agent for the resource. Valid values are:</p> <p>ERROR: Only logs error messages.</p> <p>WARN : Logs above plus warning messages.</p> <p>INFO: Logs above plus informational messages.</p> <p>TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic functions.</p> <p>Type and dimension: string-scalar</p> <p>Default: INFO</p> <p>Example: TRACE</p> <p>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.</p>

[Table 3-2](#) shows the optional attributes for configuring a WebSphere MQ queue manager.

Table 3-2 Optional attributes

Optional attribute	Description
LogDbg	<p>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 and the VCS version is 6.2 or later.</p> <p>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 WebSphereMQ resources in the cluster. If debug logs must be enabled for a specific WebSphereMQ resource, override the LogDbg attribute.</p> <p>Type and dimension: string-keylist</p> <p>Default: {} (none)</p> <p>For more information on how to use the LogDbg attribute, refer to the <i>Cluster Server Administrator's Guide</i>.</p>
EnvFile	<p>The complete path of the file name to source to set the environment prior to executing WebSphere MQ programs. Veritas recommends storing the file on the shared disk. 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.</p> <p>Type and dimension: string-scalar</p> <p>Default: No default value</p> <p>Example: /MQ/setEnv.sh</p>
MonitorProgram	<p>Absolute path name of an external, user-supplied monitor executable. For information about setting this attribute:</p> <p>See "Executing a customized monitoring program" on page 39.</p> <p>Type and dimension: string-scalar</p> <p>Default: No default value</p> <p>Example 1: /ibm/mq/myMonitor.sh</p> <p>Example 2: /ibm/mq/myMonitor.sh arg1 arg2</p>

Table 3-2 Optional attributes (*continued*)

Optional attribute	Description
SecondLevelMonitor	<p>Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the WebSphere MQ queue manager. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval, and so on.</p> <p>Note: Exercise caution while setting SecondLevelMonitor to large numbers. For example, if the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then the runmqsc command is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor.</p> <p>Note: The SecondLevelMonitor attribute is applicable to VCS versions earlier than VCS 5.1 SP1 with WebSphereMQ agent versions earlier than 5.1.9.0. From VCS version 5.1 SP1 with WebSphere MQ agent version 5.1.9.0 onwards, the SecondLevelMonitor attribute of the WebSphereMQ agent is deprecated. Instead, a resource type level attribute LevelTwoMonitorFreq should be used to specify the frequency of in-depth monitoring.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p> <p>Example: 1</p>

Table 3-2 Optional attributes (*continued*)

Optional attribute	Description
LevelTwoMonitorFreq	<p>Specifies the frequency at which the agent for this resource type must perform second-level or detailed monitoring. You can also override the value of this attribute at the resource level.</p> <p>The value indicates the number of monitor cycles after which the agent will monitor the WebSphere MQ queue manager in detail. For example, the value 5 indicates that the agent will monitor the WebSphere MQ queue manager in detail after every five online monitor intervals.</p> <p>Note: This attribute is applicable to VCS version 5.1 SP1 with WebSphere MQ agent version 5.1.9.0 or later. If the VCS version is earlier than VCS 5.1 SP1 and the WebSphere MQ agent version is earlier than 5.1.9.0, the SecondLevelMonitor attribute should be used.</p> <p>If you upgraded the VCS version to VCS 5.1 SP1 and the WebSphereMQ agent version to 5.1.9.0 (or later), and if you had enabled detail monitoring in the previous version, then do the following:</p> <ul style="list-style-type: none"> Set the value of the LevelTwoMonitorFreq attribute to the same value as that of the SecondLevelMonitor attribute. <p>Type and dimension: integer-scalar</p> <p>Default: 0</p>

Table 3-2 Optional attributes (*continued*)

Optional attribute	Description
MonitorListener	<p>Decides whether the monitor and clean functions must monitor and clean MQ listener processes (runmqslsr).</p> <p>If this attribute is set to a value greater than zero, the agent monitors MQ listener processes. The integer you specify in this attribute denotes the number of MQ listener processes that the agent must monitor.</p> <p>If you set this attribute to 0 (default value), the agent does not monitor the WebSphere MQ listener process.</p> <p>If a listener cannot be started or stopped as part of the queue manager startup or shutdown, then you must create a separate resource for that WebSphereMQ listener, using the Application agent.</p> <p>Type and dimension: integer-scalar</p> <p>Default: 0</p> <p>Example: 1</p>
MQInstallationPath	<p>When more than one copy of WebSphere MQ is installed on a system, each queue manager is associated with a particular installation. Specify the location of the WebSphere MQ installation to which this queue manager is associated.</p> <p>For WebSphere MQ versions earlier than 7.1, the default installation path is <code>/opt/mqm</code> or <code>/usr/mqm</code> for AIX platforms .</p> <p>Specify a custom installation path for WebSphere MQ 7.1 or later.</p> <p>If you change the installation that is associated with a queue manager, ensure that you update this attribute to specify the newly-associated installation path.</p> <p>Example: <code>/opt/customLocation</code></p> <p>Default: No default value</p>

Table 3-2 Optional attributes (*continued*)

Optional attribute	Description
DisabledProcesses	<p>Specifies a list of queue manager processes that the agent must bypass or disregard while performing a monitor operation.</p> <p>Typically, when the agent for WebSphere MQ performs the monitor operation, the agent searches for all system processes that must be running for a queue manager to function. These are processes that are expected to start by default when a queue manager is started on the system. However, in certain circumstances, such as for security purposes, a few of these processes might be disabled on the system. You can specify such disabled processes in the DisabledProcesses attribute. When the agent performs a monitor operation, the agent bypasses or disregards the specified processes.</p> <p>For example, the Object Authority Manager (OAM) process (amqzfuma) is expected to start by default when queue manager starts on a system. Therefore, the WebSphere MQ agent assumes that the amqzfuma process is running. However, it is possible for this process to be disabled. In such a case, you can specify the amqzfuma process in the DisabledProcesses attribute, so that the agent does not consider (bypasses) this process.</p> <p>The disabled process name must be specified as the binary name, as seen in the MQ installation path.</p> <p>Type and dimension: string-vector</p> <p>Example: amqzfuma</p> <p>Default: No default value</p>
StartOptions	<p>The options passed to the <code>strmqm</code> command to bring the resource online.</p> <p>Type and dimension: string-scalar</p> <p>Default: No default value</p>

Table 3-2 Optional attributes (*continued*)

Optional attribute	Description
StopOptions	<p>The options passed to the <code>endmqm</code> command to take the resource offline.</p> <p>This attribute must be configured only for an active instance. For a standby instance, the <code>-x</code> option is passed by default.</p> <p>Depending on whether the <code>-s</code> option is passed and whether the standby instance that corresponds to the active instance is running, the Offline function takes the following action:</p> <ul style="list-style-type: none"> ■ If the option is passed and the standby instance is <i>not</i> running, it eliminates the option from this attribute. ■ If the option is not passed and the standby instance is running, it issues an offline command to the standby instance. <p>Type and dimension: string-scalar</p> <p>Default: <code>-i</code></p>
ServiceName	<p>To start the application in <code>system.slice</code>, provide the SystemD service file for the application as the value of this attribute. The agent runs the specified file. If this attribute value is empty, the agent creates and uses the default SystemD service file.</p> <p>Type and dimension: string-scalar</p> <p>Default: No default value</p> <p>Example: <code>mq.service</code></p>
SystemDEnvFile	<p>Full path of the SystemD-compliant environment file. The agent uses this value in the WebSphereMQ service file.</p> <p>This file is in addition to the one that is specified in the <code>EnvFile</code> attribute. If you do not specify a value, the agent uses the value of the <code>EnvFile</code> attribute in the service file.</p> <p>Note: On SystemD-enabled platforms, you must specify a SystemD-compliant environment file in this attribute.</p> <p>Type and dimension: string-scalar</p> <p>Example: <code>/var/mqm/mq_sysd.env</code></p> <p>Default: No default value</p>

Note: The agent for WebSphereMQ takes into consideration some additional attributes when IMF is enabled.

See [“Attributes that enable IMF”](#) on page 87.

Executing a customized monitoring program

- The monitor function can execute a customized monitoring utility to perform an additional WebSphere MQ Queue Manager server state check.
- The monitor function executes the utility specified in the MonitorProgram attribute if the following conditions are satisfied:
- The specified utility is a valid executable file.
 - The first-level process check indicates that the WebSphere MQ Queue Manager instance is online.
 - The second-level monitor check indicates that the WebSphere MQ Queue Manager instance is online.
 - The second-level monitor check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

110 or 0	WebSphere MQ Queue Manager server instance is online
100 or 1	WebSphere MQ Queue Manager server instance is offline
99	WebSphere MQ Queue Manager server instance is unknown
Any other value	WebSphere MQ Queue Manager server instance is unknown

To ensure that the customized utility is always available to the agent, Veritas recommends storing the file in a shared directory that is available on an online node.

Configuring the service group for WebSphere MQ using the Veritas High Availability Configuration wizard

This chapter includes the following topics:

- [Installing the agent for WebSphere MQ](#)
- [Typical VCS cluster configuration in a virtual environment](#)
- [About configuring application monitoring using the Veritas High Availability solution for VMware](#)
- [Getting ready to configure VCS service groups using the wizard](#)
- [Before configuring application monitoring](#)
- [Launching the High Availability Configuration wizard](#)
- [Configuring the WebSphere MQ queue manager for high availability](#)
- [Understanding service group configurations](#)
- [Understanding configuration scenarios](#)
- [Veritas High Availability Configuration wizard limitations](#)
- [Troubleshooting](#)

- [Sample configurations](#)

Installing the agent for WebSphere MQ

You can install the agent for WebSphere MQ in the following way:

- Using the command line interface (CLI).
Use this method to install the agent in a physical or virtual environment.
For more details, refer to the Cluster Server installation guide.

Note: The installation of the agent using vSphere client is not supported on InfoScale 8.0.2 and later versions.

Typical VCS cluster configuration in a virtual environment

A typical VCS cluster configuration for WebSphere MQ Queue Manager, in a VMware virtual environment involves two or more virtual machines. The virtual machine on which the application is active, accesses a non-shared VMware VMDK or RDM disk that resides on a VMware datastore.

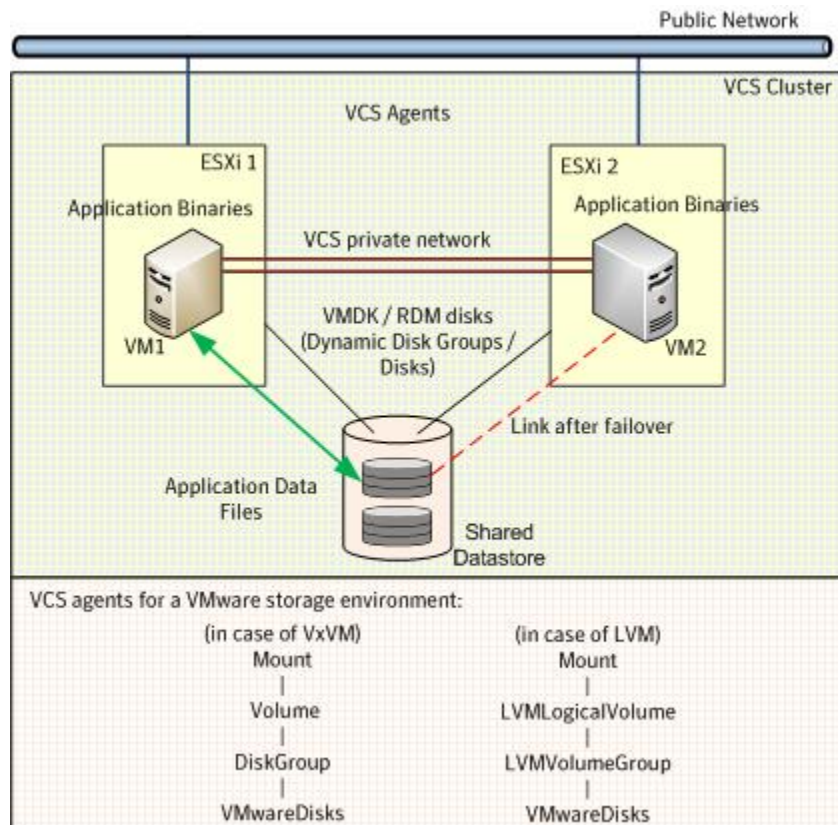
The virtual machines involved in the VCS cluster configuration may belong to a single ESX host or could reside on separate ESX hosts. If the virtual machines reside on separate ESX hosts, the datastore on which the VMware VMDK or RDM disks (on which the application data is stored) reside must be accessible to each of these ESX hosts.

The application binaries are installed on the virtual machines and the data files are installed on the VMware disk drive. The VCS agents monitor the application components and services, and the storage and network components that the application uses.

In a site recovery environment, High Availability solution additionally provides script files for the following tasks. These files are invoked when the SRM recovery plan is executed.

- Set up communication between the vCenter Server and the SRM Server at the recovery site and the virtual machines at the protected site.
- Assign a SiteID to both the sites.
- Specify attribute values for the application components at the respective site.
- Retrieve the application status in the SRM recovery report, after the virtual machine is started at the recovery site.

Figure 4-1 Typical WebSphere MQ Queue Manager cluster configuration in a VMware virtual environment



During a failover, the VCS storage agents move the VMware disks to the new system. The VCS network agents bring the network components online, and the application specific agents then start application services on the new system.

About configuring application monitoring using the Veritas High Availability solution for VMware

Consider the following before you proceed:

- You can configure application monitoring on a virtual machine using the Veritas High Availability Configuration wizard for VMware. The wizard is launched when you click **Configure application for high availability** on the Veritas High Availability tab in VMware vSphere Client.

Note: The configuration of application monitoring using vSphere client is not supported on InfoScale 8.0.2 and later versions.

- Apart from the Veritas High Availability Configuration wizard, you can also configure application monitoring using the VCS commands. For more information, refer to the *Cluster Server Administrator's Guide*.
- Veritas recommends that you first configure application monitoring using the wizard before using VCS commands to add additional components or modify the existing configuration.
Apart from configuring application availability, the wizard also sets up the other components required for successful application monitoring.
- You must not suspend a system if an application is currently online on that machine. If you suspend a system, VCS moves the disks along with the application to another system. Later, when you try to restore the suspended system, VMware does not allow the operation because the disks that were attached before the system was suspended are no longer with the system. To suspend a virtual machine, ensure that the application being monitored is not online on that system.

Note: For details about deploying, configuring, and administering the Veritas High Availability solution, refer to the *Veritas High Availability Solution Guide for VMware*.

Getting ready to configure VCS service groups using the wizard

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

- Install and enable VMware Tools on the virtual machine, where you want to monitor applications with VCS. Install a version that is compatible with the VMware ESX server.
- Install High Availability console on a Windows system in your data center and register the High Availability plug-in with the vCenter server.
- Assign Configure Application Monitoring (Admin) privileges to the logged-on user on the virtual machine where you want to configure application monitoring.
- Install the Cluster Server component.
- Install the application and the associated components that you want to monitor on the virtual machine.

- If you have configured a firewall, ensure that your firewall settings allow access to ports used by the High Availability installer, wizards, and services. Refer to the respective application configuration guide for a list of ports and services used.

Before configuring application monitoring

Note the following prerequisites before configuring application monitoring on a virtual machine:

- The Veritas High Availability Configuration wizard discovers the disks which are attached and the storage which is currently mounted. Ensure that the shared storage used by the application is mounted before you invoke the wizard.
- For all the WebSphere MQ Queue Managers that you want to configure, the DataPath, LogPath, and InstallationPath directories must be accessible from the node from where you invoke the Veritas High Availability Configuration wizard.
- The latest `VRTSmq6` and `VRTSacclib` packages must be installed on the node on which you want to monitor WebSphere MQ Queue Managers.
- You must not restore a snapshot on a virtual machine where an application is currently online, if the snapshot was taken when the application was offline on that virtual machine. Doing this may cause an unwanted failover. This also applies in the reverse scenario; you should not restore a snapshot where the application was online on a virtual machine, where the application is currently offline. This may lead to a misconfiguration where the application is online on multiple systems simultaneously.
- While creating a VCS cluster in a virtual environment, you must configure the cluster communication link over a public network in addition to private adapters. The link using the public adapter should be assigned as a low-priority link. This helps in case the private network adapters fail, leading to a condition where the systems are unable to connect to each other, consider that the other system has faulted, and then try to gain access to the disks, thereby leading to an application fault.
- You must not select teamed network adapters for cluster communication. If your configuration contains teamed network adapters, the wizard groups them as "NIC Group #N" where "N" is a number assigned to the teamed network adapters. A teamed network adapter is a logical NIC, formed by grouping several physical NICs together. All NICs in a team have an identical MAC address, due to which you may experience the following issues:
 - SSO configuration failure.

- The wizard may fail to discover the specified network adapters.
- The wizard may fail to discover/validate the specified system name.
- Verify that the boot sequence of the virtual machine is such that the boot disk (OS hard disk) is placed before the removable disks. If the sequence places the removable disks before the boot disk, the virtual machine may not reboot after an application failover. The reboot may halt with an "OS not found" error. This issue occurs because during the application failover the removable disks are detached from the current virtual machine and are attached on the failover target system.
- Verify that the disks used by the application that you want to monitor are attached to non-shared controllers so that they can be deported from the system and imported to another system.
- If multiple types of SCSI controllers are attached to the virtual machines, then storage dependencies of the application cannot be determined and configured.
- The term 'shared storage' refers to the removable disks attached to the virtual machine. It does not refer to disks attached to the shared controllers of the virtual machine.
- If you want to configure the storage dependencies of the application through the wizard, the LVM volumes or VxVM volumes used by the application should not be mounted on more than one mount point path.
- The host name of the system must be resolvable through the DNS server or, locally, using `/etc/hosts` file entries.

Launching the High Availability Configuration wizard

You can launch the High Availability Configuration wizard from:

- VMware vSphere Client: See [To launch the wizard from the VMware vSphere Client](#).

Note: Launching the wizard from the VMware vSphere Client is not supported on InfoScale 8.0.2 and later versions.

- A browser window: See [To launch the wizard from a browser window](#).

You must launch the High Availability Configuration wizard from the system where the disk residing on the shared datastore is attached.

To launch the wizard from the VMware vSphere Client

- 1 Launch the VMware vSphere Client and connect to the VMware vCenter Server that hosts the virtual machine.
- 2 From the vSphere Client's Inventory view in the left pane, select the virtual machine where you want to configure application monitoring.
- 3 Skip this step if you have already configured single sign-on during guest installation.

Select the Veritas High Availability tab and in the Veritas High Availability View page, specify the credentials of a user account that has administrative privileges on the virtual machine and click **Configure**.

The High Availability console sets up a permanent authentication for the user account on that virtual machine.

- 4 Depending on your setup, use one of the following options to launch the wizard:
 - If you have not configured a cluster, click the **Configure a VCS Cluster** link.
 - If you have already configured a cluster, click **Actions > Configure application for high availability** or the **Configure application for high availability** link.
 - If you have already configured a cluster and configured an application for monitoring, click **Actions > Configure application for high availability**.

To launch the wizard from a browser window

- 1 Open a browser window and enter the following URL:

`https://<VMNameorIP>:5634/vcs/admin/application_health.html`

<VMNameorIP> is the virtual machine name or IP address of the system on which you want to configure application monitoring.

- 2 In the Authentication dialog box, enter the username and password of the user who has administrative privileges.
- 3 Depending on your setup, use one of the following options to launch the wizard:
 - If you have not configured a cluster, click the **Configure a VCS Cluster** link.
 - If you have already configured a cluster, click **Actions > Configure application for high availability** or the **Configure application for high availability** link.
 - If you have already configured a cluster and configured an application for monitoring, click **Actions > Configure application for high availability**.

Configuring the WebSphere MQ queue manager for high availability

Perform the following steps to configure the WebSphere MQ queue manager for high availability on a virtual machine.

To configure the WebSphere MQ queue manager for high availability

- 1 Launch the Veritas High Availability Configuration wizard. See [“Launching the High Availability Configuration wizard”](#) on page 45.
- 2 Review the information on the Welcome panel and click **Next**.
- 3 On the Application Selection panel, select **WebSphere MQ** from the Supported Applications list and click **Next**.

You can use the Search box to search for the WebSphere MQ application.

- 4 On the Application Inputs panel, from the Queue Manager list, select the WebSphere MQ queue manager instances that you want to monitor.
- 5 Skip this step if you do not want to specify environment file for the queue manager.

In the **Environment File** field, enter the path where the environment file is located. The environment file must be accessible on the node from which you invoke the wizard.

- 6 Skip this step if you do not want to monitor the related queue manager listener.

If you want to monitor the related queue manager listener, check the **Monitor Queue Manager Listener** check box and select one of the following options to proceed:

- Select the **Along with Queue Manager** radio button to monitor the queue manager listeners along with WebSphere MQ queue manager instances. The `MonitorListener` attribute of the WebSphere MQ resource is enabled.
- Select the **Independent of Queue Manager** radio button to monitor the queue manager listeners separately from queue manager instances. In the **Listener Port** field, enter the listener port number. A separate resource for the listener is created using the Application agent.

Note: The Veritas High Availability Configuration wizard supports only one listener associated with each queue manager.

- 7 Click **Next**.

- 8
- Skip this step if you do not want to select mount points for the queue manager instances that you want to monitor.

On the Storage Selection panel, select the mount points that you want to associate with each queue manager instance. Resource-level dependencies are created between queue managers and their corresponding mount resources. If the same mount point is associated with two or more queue managers, the queue managers are configured as part of the same service group.

Note: The Veritas High Availability Configuration wizard does not automatically associate mount points with queue manager instances.

- 9
- Click **Next**.
- 10
- On the Configuration Inputs panel, use the Edit icon to specify the user name and password of the systems for VCS cluster operations.

Cluster systems lists the systems included in the cluster configuration. **Application failover targets** lists the systems to which the application can fail over. Move the required systems to the Application failover targets list. Use the up and down arrow keys to define the priority order of the failover systems. The local system is selected by default for both, the cluster operations and as a failover target.

- 11
- Click **Next**.
- 12
- Skip this step if you do not want to add more systems to your cluster.

To add a system to the VCS cluster, click **Add System**. In the Add System dialog box, specify the following details of the system that you want to add to the VCS cluster and click **OK**.

System Name or IP address	Specify the name or IP address of the system that you want to add to the VCS cluster.
User name	Specify the user account for the system. Typically, this is the root user. The root user should have the necessary privileges.
Password	Specify the password for the user account mentioned.
Use the specified user account on all systems	Select to use the specified user account on all the cluster systems that have the same user name and password.

13 If you are configuring a cluster and if you want to modify the security settings for the cluster, click **Advanced Settings**. In the Advanced settings dialog box, specify the following details and click **OK**.

- Use Single Sign-on

Select to configure single sign-on using VCS Authentication Service for cluster communication.

This option is enabled by default.
- Use VCS user privileges

Select to configure a user with administrative privileges to the cluster.

Specify the username and password and click **OK**.

Note: The **Advanced Settings** link is not visible if the cluster is already created.

14 Skip this step if the cluster is already configured. By default, the links are configured over Ethernet.

On the Network Details panel, select the type of network protocol to configure the VCS cluster network links and then specify the adapters for network communication.

The wizard configures the VCS cluster communication links using these adapters. You must select a minimum of two adapters per system.

Select **Use MAC address for cluster communication (LLT over Ethernet)** or **Use IP address for cluster communication (LLT over UDP)**, depending on the IP protocol that you want to use and then specify the required details to configure the VCS cluster communication network links. You must specify these details for each cluster system.

- To configure LLT over Ethernet, select the adapter for each network communication link. You must select a different network adapter for each communication link.
- To configure LLT over UDP, select the type of IP protocol and then specify the required details for each communication link.
Depending on the IP protocol, specify the following:

- Network Adapter

Select a network adapter for the communication links.

You must select a different network adapter for each communication link.
- IP Address

Displays the IP address.

Port	Specify a unique port number for each link. For IPv4 and IPv6, the port range is from 49152 to 65535. A specified port for a link is used for all the cluster systems on that link.
Subnet mask (IPv4)	Displays the subnet mask details.
Prefix (IPv6)	Displays the prefix details.

By default, one of the links is configured as a low-priority link on a public network interface. The second link is configured as a high-priority link. To change a high-priority link to a low-priority link, click **Modify**. In the Modify low-priority link dialog box, select the link and click **OK**.

Note: Veritas recommends that you configure one of the links on a public network interface. You can assign the link on the public network interface as a low-priority link for minimal VCS cluster communication over the link.

15 Skip this step if the application does not use virtual IP address.

In the Virtual Network Details panel, specify the IP protocol and virtual IP address for the application.

Depending on the IP protocol, specify the following:

Virtual IP address	Specify a unique virtual IP address.
Subnet Mask (IPv4)	Specify the subnet mask details.
Prefix (IPv6)	Select the prefix from the drop-down list.
Network Adapter	Select the network adapter that will host the virtual IP.

If you want to add another virtual IP address for your application, click **Add virtual IP address**.

If you select multiple instances for the same virtual IP address, those instances are configured in the same service group.

16 Click **Next**.

17 Skip this step if you did not select mount points.

On the Target ESX Details panel, specify all the ESX hosts to which the virtual machines can fail over. Each ESX host must be able to access the required shared datastores that contain visible disks. Enter the administrative user account details for each ESX host and click **Next**.

To specify the ESX hosts, click **Add ESX Host** and in the Add ESX Host dialog box, specify the following details:

ESX hostname or IP address Specify the target ESX hostname or IP address.

The virtual machines can fail over on this ESX host during vMotion. All the additional ESX hosts should have access to the datastore on which the disks used by the application reside.

User name Specify a user account for the ESX host. The user account must have administrator privileges on the specified ESX host.

Password Specify the password for the user account provided in the User name text box.

The wizard validates the user account and the storage details on the specified ESX hosts.

18 On the Summary panel, review the VCS cluster configuration summary and then click **Next** to proceed with the configuration.

If the network contains multiple clusters, the wizard verifies the cluster ID with the IDs assigned to all the accessible clusters in the network. The wizard does not validate the assigned ID with the clusters that are not accessible during the validation. Veritas recommends you to validate the uniqueness of the assigned ID in the existing network. If the assigned ID is not unique or if you want to modify the cluster name or cluster ID, click **Edit**. In the Edit Cluster Details dialog box, modify the details as necessary and click **OK**.

- 19** On the Implementation panel, the wizard creates the VCS cluster, configures the application for monitoring, and creates cluster communication links.

The wizard displays the status of each task. After all the tasks are complete, click **Next**.

If the configuration task fails, click **View Logs** to check the details of the failure. Rectify the cause of the failure and run the wizard again to configure application monitoring.

- 20** On the Finish panel, click **Finish** to complete the wizard workflow.

This completes the application monitoring configuration.

If the application status shows as not running, click **Start** to start the configured components on the system.

Understanding service group configurations

One or more WebSphere MQ Queue Manager instances can be discovered on a virtual machine. These WebSphere MQ Queue Manager instances may or may not share the same mount points, disks, disk groups, volume, or virtual IP address. The WebSphere MQ Queue Manager listeners that do not share any of these forms a separate service group.

Resource dependency

Following are the resource dependencies:

- When the listeners are monitored using the Independent of Queue Manager option, the listener resources associated with a WebSphere MQ Queue Manager instance depends on the WebSphere MQ resource.
- Listener resources also depend on the configured IP resources.
- When you associate a mount point with a queue manager using the Veritas High Availability Configuration wizard, a dependency is created between the corresponding WebSphere MQ resource and the mount resource.
- Mount point resources depend on either LVM (logical volume) or VxVM volume.
 - VxVM volume depends on DiskGroup resources.
 - LVM depends on LVM volume group.
- DiskGroup and LVM volume group resources depend on the shared disks which are configured as VMwareDisks resources.

Service group dependency

The Veritas High Availability Configuration wizard does not create service group dependency for WebSphere MQ.

Infrastructure service groups

As part of configuring the application, the Veritas High Availability Configuration wizard:

- Configures application specific service groups and resources.
- Configures the VCS infrastructure service group (VCSInfraSG).

VCSInfraSG includes a resource called VCSNotifySinkRes. The type of this resource is Process. VCSNotifySinkRes configures and administers the notify_sink process on the guest. The notify_sink process sends the details about service groups and its attributes to the Veritas High Availability Console. This information is used for reporting purpose and is displayed on the Dashboard.

Note: VCSInfraSG is an internal service group. You must not add or delete resources from this service group.

The following are the VCSInfraSG notes:

- Before you configure the application for monitoring, ensure that SSO is configured between the Veritas High Availability Console and the guest. If SSO is not configured, VCSInfraSG fails to come online.
- If VCSInfraSG or VCSNotifySinkRes faults, ensure that SSO is configured between the Veritas High Availability Console and the guest. Clear the faults and bring the resource online again.
- VCSInfraSG or VCSNotifySinkRes must not be taken offline because it affects the information displayed on the Dashboard.

Understanding configuration scenarios

You can configure WebSphere MQ Queue Manager instances in different ways using the Veritas High Availability Configuration wizard.

Table 4-1 WebSphere MQ Queue Manager configurations

Configuration Type	Reference
Configuring a single instance/multiple instances in VCS	See Configuring a single instance/multiple instances in VCS
Configuring multiple WebSphere MQ Queue Manager instances in VCS using multiple runs of the wizard	See Configuring multiple WebSphere MQ Queue Manager instances in VCS using multiple runs of the wizard
Configuring multiple applications	See Configuring multiple applications

Configuring a single instance/multiple instances in VCS

Use the Veritas High Availability Configuration wizard to configure one or more WebSphere MQ Queue Manager instances in a single run.

In the Application Inputs panel, select the WebSphere MQ Queue Manager instances from the Queue Manager list. For each instance, you can specify the following optional parameters:

- Environment file
- Listener related to the specific Queue Manager

Configuring multiple WebSphere MQ Queue Manager instances in VCS using multiple runs of the wizard

If you are configuring the first WebSphere MQ Queue Manager instance on a machine where more than one Queue Manager instance is present, configure it by following the steps in the [Configuring a single instance/multiple instances in VCS](#) section.

The Veritas High Availability Configuration wizard will not allow you to configure the next WebSphere MQ Queue Manager instance if any of the mandatory dependent resources such as mount point, disk group, and disk are already configured in VCS.

- If existing resources are part of the WebSphere MQ service group, unconfigure the existing service group and then reconfigure the new instance along with the old instances/listeners which were part of the pre-existing service group.
- If existing resources are part of an application service group other than WebSphere MQ, the wizard does not support configuring multiple applications. You can configure these applications through CLI or Veritas InfoScale Operations Manager.

Configuring multiple applications

If you run the Veritas High Availability Configuration wizard multiple times, you can configure multiple applications of different types.

If you are configuring the first application on a machine where more than one application is running, you can configure it by following the steps in the [Configuring a single instance/multiple instances in VCS](#) section.

The Veritas High Availability Configuration wizard will not allow you to configure the next application if any of the mandatory dependent resources such as mount point, disk group, and disk are already configured in VCS.

Veritas High Availability Configuration wizard limitations

Following are the Veritas High Availability Configuration wizard limitations:

- If the WebSphere MQ Queue Manager instance is already configured, that instance is not shown in the list of queue managers and it will not be available for configuration. In such a scenario, no error message will be displayed. For example, if there are two WebSphere MQ Queue Manager instances running on the system with the same mount point and assuming that the first instance is configured, if you run the wizard to configure the next instance, the following error message is displayed:

```
The wizard has failed to discover WebSphere MQ Queue Manager
on the system
```

However, if the WebSphere MQ queue managers do not share any resources (storage and network), you can configure the WebSphere MQ Queue Manager.

- The wizard supports discovery of only LVM or VxVM type of storage.
- The wizard will not discover the disks used by the application if the controllers attached to the virtual machine are of different type. To correctly discover and identify the association of mount points to the virtual disks, all the controllers attached to the virtual machine must be of same type.
- The wizard will not discover disks which are attached to the virtual machine in shared mode.
- Although the WebSphereMQ agent allows the MonitorListener attribute to be set to any positive integer, the wizard allows the MonitorListener attribute to be set to 0 or 1. To specify multiple listeners for monitoring, use HA commands or Veritas InfoScale Operations Manager.

- The wizard does not support the configuration of the DisabledProcesses attribute. To configure this attribute, use HA commands or Veritas Infoscale Operations Manager.

Troubleshooting

This section lists common troubleshooting scenarios that you may encounter while or after configuring application monitoring.

Veritas High Availability Configuration wizard displays blank panels

The Veritas High Availability Configuration wizard may fail to display the wizard panels. The window may appear blank.

Workaround:

Verify that the Veritas ApplicationHA Service is running on the Veritas High Availability Console host and then launch the wizard again.

The Veritas High Availability Configuration wizard displays the "hadiscover is not recognized as an internal or external command" error

While configuring application monitoring the Veritas High Availability Configuration wizard may display the "hadiscover is not recognized as an internal or external command" error, after you click Next on the Application Selection panel.

This issue occurs if you launch the wizard from a system where you have reinstalled the Veritas High Availability guest components.

Workaround:

Close the wizard, restart the Veritas Storage Foundation Messaging Service, and then re-run the wizard.

Running the 'hastop -all' command detaches virtual disks

The `hastop -all` command takes offline all the components and component groups of a configured application, and then stops the VCS cluster. In the process, the command detaches the virtual disks from the VCS cluster nodes.

Workaround:

If you want to stop the VCS cluster (and not the applications running on cluster nodes), instead of the `hastop -all` command, use the following command:


```
hastop -all -force
```

This command stops the cluster without affecting the virtual disks attached to the VCS cluster nodes.

Log files

The log files are stored in the virtual machine on which you configured application monitoring.

The `healthview_A.log` file contains the steps performed by the back-end to configure the application. To check the file, you must access:

```
/var/VRTSvcs/log/healthview_A.log
```

The `WebSphereMQ_A.log` file contains the actions performed by the agent. To check the file, you must access:

```
/var/VRTSvcs/log/WebSphereMQ_A.log
```

The `engine_A.log` file contains the actions performed by the VCS cluster. To check the file, you must access:

```
/var/VRTSvcs/log/engine_A.log
```

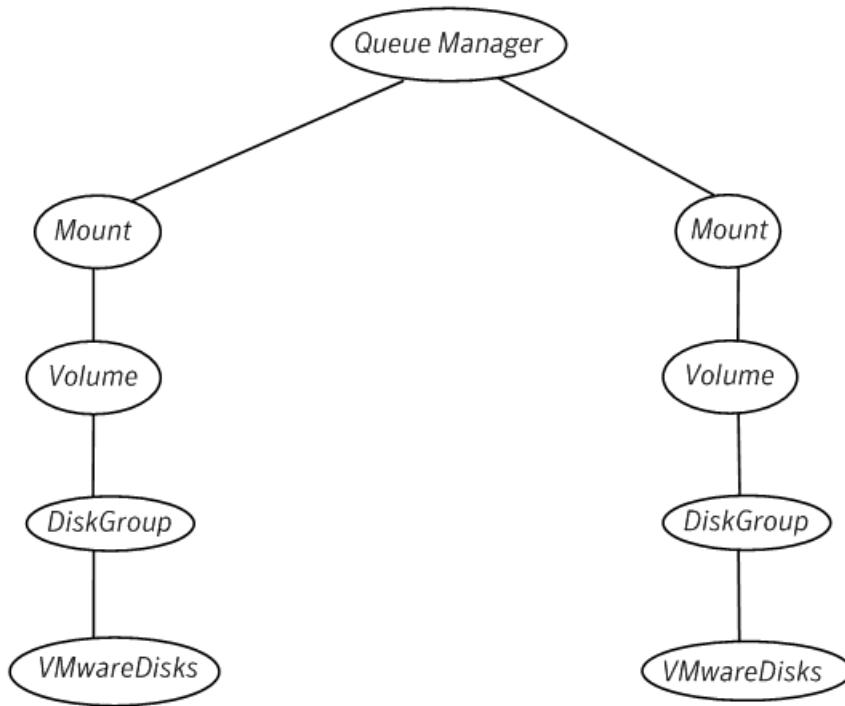
Sample configurations

The sample configurations includes description for typical service groups that are configured using the Veritas High Availability Configuration wizard.

Sample VCS configuration file for single WebSphere MQ Queue Manager instance (VxVM)

[Figure 4-2](#) shows a typical service group configured to monitor the state of a WebSphere MQ Queue Manager instance. In this example, the WebSphere MQ Queue Manager instance uses VxVM volume as storage.

Figure 4-2 Dependency graph for single WebSphere MQ Queue Manager instance (VxVM)



Review the sample configuration with a resource of type WebSphere MQ Queue Manager that is configured as follows in main.cf file.

```

include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"
include "WebSphereMQTypes.cf"

cluster Cluster_32625 (
    SecureClus = 1
)

system rhel5u8mq1 (

```

```

group MQ_1 (
  SystemList = { rhel5u8mq1 = 0 }
  UserAssoc = { Type = WebSphereMQ,
    Name = "WebSphere MQ: Queue manager" }
)

DiskGroup dg_QM1Log (
  DiskGroup = QM1Log
)

DiskGroup dg_QM1Qmgrs (
  DiskGroup = QM1Qmgrs
)

Mount mnt_QM71_2 (
  MountPoint = "/usr/queueManager/qmgrs/QM71"
  BlockDevice = "/dev/vx/dsk/QM1Qmgrs/QM1Qmgrs_Vol"
  FSType = vxfs
  MountOpt = "rw,delaylog,largefiles,ioerror=mwdisable"
  FsckOpt = "-y"
)

Mount mnt_QM71_3 (
  MountPoint = "/usr/queueManager/logs/QM71"
  BlockDevice = "/dev/vx/dsk/QM1Log/QM1Log_Vol"
  FSType = vxfs
  MountOpt = "rw,delaylog,largefiles,ioerror=mwdisable"
  FsckOpt = "-y"
)

VMwareDisks VMwareDisk_1 (
  ESXDetails = {
    "vcs1x658.domain.com" = "root=HVNTtKVkPHnINjNK" }
  DiskPaths = {
    "6000C293-f663-218a-7ac7-16175cde2daa:[rhel5MQ1]
    QM1qmgrs.vmdk" = "0:1" }
  VMRegisterWait = 5
)

VMwareDisks VMwareDisk_2 (
  ESXDetails = {
    "vcs1x658.domain.com" = "root=HVNTtKVkPHnINjNK" }

```

```

DiskPaths = {
    "6000C290-f3f0-77ff-49b1-74ef05b87f8b:[rhe15MQ1]
QM1log.vmdk" = "0:2" }
VMRegisterWait = 5
)

Volume vol_QM1Log_Vol_QM1Log (
    DiskGroup = QM1Log
    Volume = QM1Log_Vol
)

Volume vol_QM1Qmgrs_Vol_QM1Qmgrs (
    DiskGroup = QM1Qmgrs
    Volume = QM1Qmgrs_Vol
)

WebSphereMQ QM71 (
    QueueManager = QM71
    MQVer = "7.1"
    MonitorListener = 1
    MQInstallationPath = "/opt/MQ"
)

QM71 requires mnt_QM71_2
QM71 requires mnt_QM71_3
dg_QM1Log requires VMwareDisk_2
dg_QM1Qmgrs requires VMwareDisk_1
mnt_QM71_2 requires vol_QM1Qmgrs_Vol_QM1Qmgrs
mnt_QM71_3 requires vol_QM1Log_Vol_QM1Log
vol_QM1Log_Vol_QM1Log requires dg_QM1Log
vol_QM1Qmgrs_Vol_QM1Qmgrs requires dg_QM1Qmgrs

// resource dependency tree
//
// group MQ_1
// {
//   WebSphereMQ QM71
//   {
//     Mount mnt_QM71_2
//     {
//       Volume vol_QM1Qmgrs_Vol_QM1Qmgrs
//       {

```

```
//          DiskGroup dg_QM1Qmgrs
//          {
//          VMwareDisks VMwareDisk_1
//          }
//      }
//  }
//      Mount mnt_QM71_3
//      {
//          Volume vol_QM1Log_Vol_QM1Log
//          {
//          DiskGroup dg_QM1Log
//          {
//          VMwareDisks VMwareDisk_2
//          }
//          }
//      }
//  }
//  }
```

```
group VCSInfraSG (
    SystemList = { rhel5u8mq1 = 0 }
    UserAssoc = { Type = "vcs internal",
        Name = "VCS Infrastructure service group" }
    Parallel = 1
    AutoStartList = { rhel5u8mq1 }
    OnlineRetryLimit = 5
)

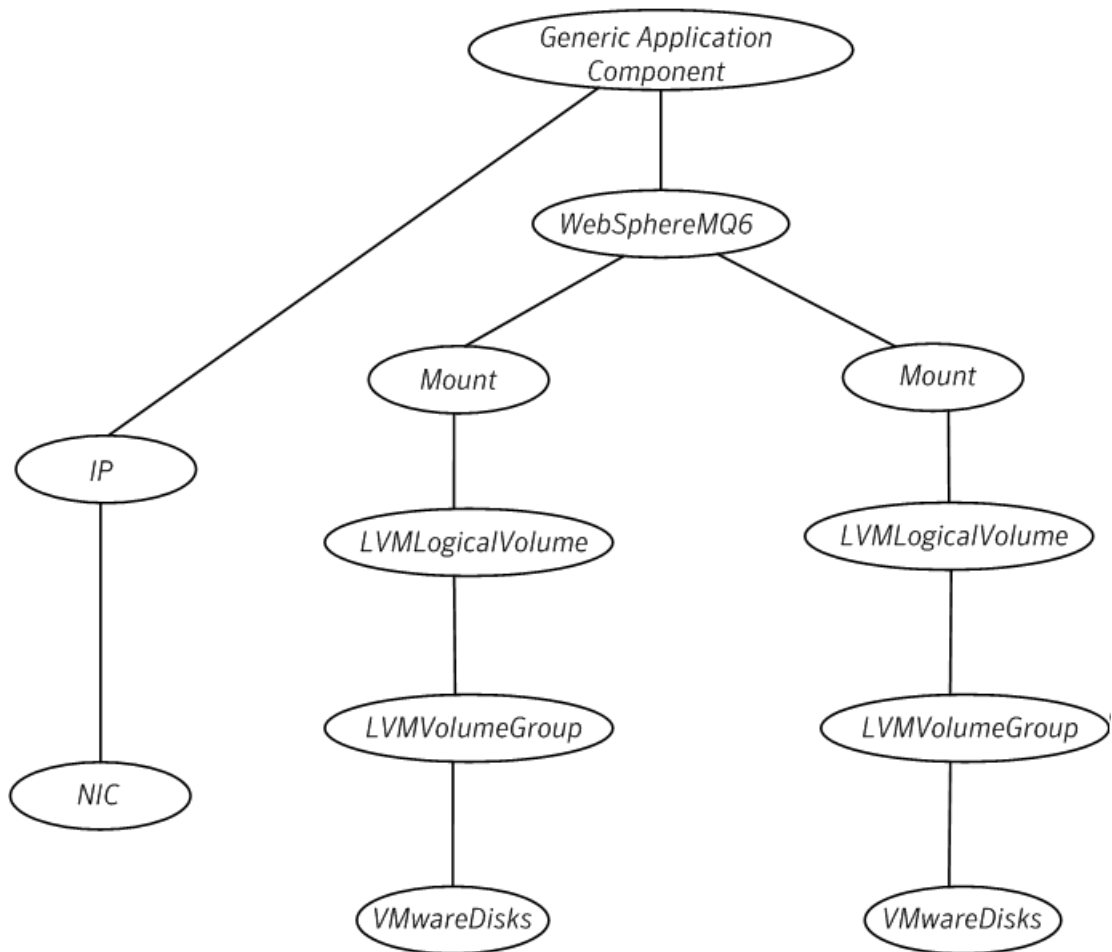
Process VCSNotifySinkRes (
    PathName = "/opt/VRTSvcs/portal/admin/notify_sink"
)

// resource dependency tree
//
// group VCSInfraSG
// {
// Process VCSNotifySinkRes
// }
```

Sample VCS configuration file for single WebSphere MQ Queue Manager instance (LVM)

Figure 4-3 shows a typical service group configured to monitor the state of a WebSphere MQ Queue Manager instance. In this example, the WebSphere MQ Queue Manager instance uses LVM volume as storage.

Figure 4-3 Dependency graph for single WebSphere MQ Queue Manager instance (LVM)



Review the sample configuration with a resource of type WebSphere MQ Queue Manager that is configured as follows in main.cf file.

```

include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"
include "WebSphereMQTypes.cf"

cluster Cluster_3175 (
    SecureClus = 1
)

system sles11sp2mq1 (
)

group MQ_1 (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = WebSphereMQ,
        Name = "WebSphere MQ: Queue manager" }
)

Application QM1_Listener_res (
    User = mqm
    StartProgram = "/opt/mqm/bin/runmqslsr -t tcp -i 10.209.70.205
        -p 1490 -m QM1 &"
    StopProgram = "/opt/mqm/bin/endmqslsr -m QM1"
    MonitorProcesses = {
        "/opt/mqm/bin/runmqslsr -t tcp -I 10.209.70.205 -p 1490 -m QM1" }
)

IP IP_10-209-70-205 (
    Device @sles11sp2mq1 = eth0
    Address = "10.209.70.205"
    NetMask = "255.255.252.0"
)

LVMLLogicalVolume lvol_QM1QmgrVol_QM1Qmgrs (
    LogicalVolume = QM1QmgrVol
    VolumeGroup = QM1Qmgrs
)

LVMLLogicalVolume lvol_QM1TmpVol_QM1Tmp (
    LogicalVolume = QM1TmpVol
    VolumeGroup = QM1Tmp
)

```

```

    )

    LVMVolumeGroup volg_QM1Qmgrs (
        VolumeGroup = QM1Qmgrs
    )

    LVMVolumeGroup volg_QM1Tmp (
        VolumeGroup = QM1Tmp
    )

    Mount mnt_QM1_2 (
        MountPoint = "/var/mqm/qmgrs/QM1"
        BlockDevice = "/dev/mapper/QM1Qmgrs-QM1QmgrVol"
        FSType = ext3
        MountOpt = rw
        FsckOpt = "-y"
    )

    Mount mnt_QM1_3 (
        MountPoint = "/var/mqm/log/QM1"
        BlockDevice = "/dev/mapper/QM1Tmp-QM1TmpVol"
        FSType = ext3
        MountOpt = rw
        FsckOpt = "-y"
    )

    Proxy NICProxy_10-209-70-205 (
        TargetResName @sles11sp2mq1 = NIC_sles11sp2mq1_eth0
    )

    VMwareDisks VMwareDisk_1 (
        ESXDetails = {
            "vcs1x658.domain.com" = "root=drjPgrGldJejFjg" }
        DiskPaths = {
            "6000C29f-3b70-a4e1-9933-0b882e6fd051:[SAPMQ]"
            slesQM2Tmp.vmdk" = "0:1" }
        VMRegisterWait = 5
    )

    VMwareDisks VMwareDisk_2 (
        ESXDetails = {
            "vcs1x658.domain.com" = "root=drjPgrGldJejFjg" }
        DiskPaths = {

```



```

        "6000C29c-cfd8-3914-fe12-a5b7461fbd9e:[SAPMQ]
slesQM2Qmgrs.vmdk" = "0:2" }
VMRegisterWait = 5
)

WebSphereMQ QM1 (
    QueueManager = QM1
    MQVer = "7.0"
)

IP_10-209-70-205 requires NICProxy_10-209-70-205
QM1 requires mnt_QM1_2
QM1 requires mnt_QM1_3
QM1_Listener_res requires IP_10-209-70-205
QM1_Listener_res requires QM1
lvol_QM1QmgrVol_QM1Qmgrs requires volg_QM1Qmgrs
lvol_QM1TmpVol_QM1Tmp requires volg_QM1Tmp
mnt_QM1_2 requires lvol_QM1QmgrVol_QM1Qmgrs
mnt_QM1_3 requires lvol_QM1TmpVol_QM1Tmp
volg_QM1Qmgrs requires VMwareDisk_2
volg_QM1Tmp requires VMwareDisk_1

// resource dependency tree
//
// group MQ_1
// {
//     Application QM1_Listener_res
//     {
//         IP IP_10-209-70-205
//         {
//             Proxy NICProxy_10-209-70-205
//         }
//         WebSphereMQ QM1
//         {
//             Mount mnt_QM1_2
//             {
//                 LVMLogicalVolume lvol_QM1QmgrVol_QM1Qmgrs
//                 {
//                     LVMVolumeGroup volg_QM1Qmgrs
//                     {
//                         VMwareDisks VMwareDisk_2
//                     }
//                 }
//             }
//         }
//     }
// }

```

```

//          }
//      }
//      Mount mnt_QM1_3
//      {
//          LVMLogicalVolume lvol_QM1TmpVol_QM1Tmp
//          {
//              LVMVolumeGroup volg_QM1Tmp
//              {
//                  VMwareDisks VMwareDisk_1
//              }
//          }
//      }
//  }
//  }
//  }
//  }

```

```

group VCSInfraSG (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = "vcs internal",
        Name = "VCS Infrastructure service group" }
    Parallel = 1
    AutoStartList = { sles11sp2mq1 }
    OnlineRetryLimit = 5
)

Process VCSNotifySinkRes (
    PathName = "/opt/VRTSvcs/portal/admin/notify_sink"
)

```

```

// resource dependency tree
//
// group VCSInfraSG
// {
//     Process VCSNotifySinkRes
// }

```

```

group sles11sp2mq1_NIC_SG (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = "vcs internal", Name = "NIC service group" }

```

```

)

NIC NIC_sles11sp2mq1_eth0 (
    Device @sles11sp2mq1 = eth0
    Mii = 0
)

Phantom Phantom_NIC_SGsles11sp2mq1 (

// resource dependency tree
//
// group sles11sp2mq1_NIC_SG
// {
// NIC NIC_sles11sp2mq1_eth0
// Phantom Phantom_NIC_SGsles11sp2mq1
// }

```

Sample Configuration files at Primary and Secondary cluster for IBM WebSphereMQ - RDQM HA/DR

An excerpt from the `main.cf` file for IBM WebSphereMQ - RDQM HA/DR at Primary and Secondary cluster is as follows:

main.cf file at primary cluster:

```

include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"
include "RestServerTypes.cf"
include "SybaseTypes.cf"
include "WebSphereMQTypes.cf"

cluster clsrdqm1 (

```

```

SecInfo256 = "51:de:f1:71:60:01:2b:18:c3:df:52:a9:94:70:da:
b7:3f:78:57:0d:ee:9d:09:c1:b2:e4:8d:53:6c:ef:48:09"
ClusterAddress = "10.113.55.93"
ProtocolNumber = 11000
HacliUserLevel = COMMANDROOT
IV256List = {
    "2b:c0:cf:f1:c4:46:f0:57:41:3e:20:9a:75:96:a0:16:
    25:ea:00:d4:e0:f0:3c:a8:4f:17:98:1d:dd:bb:b8:ad:
    a6:9f:6a:04:d6:21:7a:d9:df:2c:ad:40:1b:22:bf:47:
    e4:0e:f9:5b:e8:94:99:9c:96:bb:12:b4:13:17:b1:
    ff" = "4c:40:11:91:8a:50:0c:85:00:00:00:00:00:00:
    00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
    00:00:00",
    "8b:9a:80:d1:8f:bb:98:64:a5:d8:bd:16:d0:32:20:19:
    9a:04:11:6d:71:81:d6:99:56:93:4e:80:b7:7b:8c:70:
    56:da:78:8a:45:3f:17:2e:4a:5c:4d:38:d1:aa:89:87:
    7f:fe:5b:fd:fa:4f:1d:9c:9b:4d:b0:a9:e9:4f:e1:
    51" = "73:ff:56:07:2d:81:04:0f:00:00:00:00:00:00:
    00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
    00:00:00" }
)

remoteclass ClsRDQMGC (
    ClusterAddress = "10.113.55.94"
)

heartbeat Icmp (
    ClusterList = { ClsRDQMGC }
    Arguments @ClsRDQMGC = { "10.113.55.94" }
)

system RDQM-Vijay-RHL9-VM1 (
)

system RDQM-Vijay-RHL9-VM2 (
)

system RDQM-Vijay-RHL9-VM3 (
)

group Azure-auth-grp (
    SystemList = { RDQM-Vijay-RHL9-VM1 = 0, RDQM-Vijay-RHL9-VM2 = 1,
RDQM-Vijay-RHL9-VM3 = 2 }
)

```

```
Parallel = 1
)

AzureAuth azure-auth-res (
    SubscriptionId = 6940a326-xxxx-xxxx-xxxx-c1e9bbdf1d63
    ClientId = 941832d6-b890-xxxx-xxx-6934336e0595
    SecretKey = "8b:9a:80:d1:8f:bb:98:64:a5:d8:bd:16:d0:32:
20:19:9a:04:11:6d:71:81:d6:99:56:93:4e:80:b7:7b:8c:70:
56:da:78:8a:45:3f:17:2e:4a:5c:4d:38:d1:aa:89:87:7f:fe:
5b:fd:fa:4f:1d:9c:9b:4d:b0:a9:e9:4f:e1:51"
    TenantId = fc8e13c0-xxxx-xxxx-xxxx-ca318e6cac32
)

Phantom phatom-res (
)

// resource dependency tree
//
//      group Azure-auth-grp
//      {
//      AzureAuth azure-auth-res
//      Phantom phatom-res
//      }

group ClusterService (
    SystemList = { RDQM-Vijay-RHL9-VM1 = 0, RDQM-Vijay-RHL9-VM2 = 1,
RDQM-Vijay-RHL9-VM3 = 2 }
    AutoStartList = { RDQM-Vijay-RHL9-VM1, RDQM-Vijay-RHL9-VM2,
RDQM-Vijay-RHL9-VM3 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120
)

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
    RestartLimit = 3
)
```

```
AzureIP ip-gco-res-azure (
    PrivateIP = "10.113.55.93"
    NICDevice = eth0
    AzureAuthResName = azure-auth-res
)
```

```
IP gcoip (
    Device = eth0
    Address = "10.113.55.93"
    NetMask = "255.255.255.128"
)
```

```
NIC gconic (
    Device = eth0
)
```

```
gcoip requires ip-gco-res-azure
ip-gco-res-azure requires gconic
wac requires gcoip
```

```
// resource dependency tree
//
//      group ClusterService
//      {
//      Application wac
//      {
//      IP gcoip
//      {
//      AzureIP ip-gco-res-azure
//      {
//      NIC gconic
//      }
//      }
//      }
//      }
//      }
```

```
group MQGrp (
    SystemList = { RDQM-Vijay-RHL9-VM1 = 0, RDQM-Vijay-RHL9-VM2 = 1,
                  RDQM-Vijay-RHL9-VM3 = 2 }
    ClusterList = { clsrdqm1 = 0, ClsRDQMGC0 = 1 }
    AutoStartList = { RDQM-Vijay-RHL9-VM1, RDQM-Vijay-RHL9-VM2,
```

```

        RDQM-Vijay-RHL9-VM3 }
ClusterFailOverPolicy = Auto
)

CFSMount MQ_log_mnt (
    Critical = 0
    MountPoint = "/mqmnt/log"
    BlockDevice = "/dev/vx/dsk/mqdg/mq_log"
    MountOpt = "mincache=direct"
)

CFSMount MQ_qmgrs_mnt (
    Critical = 0
    MountPoint = "/mqmnt/qmgrs"
    BlockDevice = "/dev/vx/dsk/mqdg/mq_qmgrs"
    MountOpt = "mincache=direct"
)

RVGPrimary MQ_RvgPri (
    RvgResourceName = MQ_Rvg
    AutoResync = 1
)

WebSphereMQ MQ_QM1 (
    QueueManager = QM1
    MQVer = "9.3"
    MQInstallationPath = "/opt/mqm"
)

requires group RVGgroup online local firm
MQ_QM1 requires MQ_log_mnt
MQ_QM1 requires MQ_qmgrs_mnt
MQ_log_mnt requires MQ_RvgPri
MQ_qmgrs_mnt requires MQ_RvgPri

// resource dependency tree
//
//      group MQGrp
//      {
//      WebSphereMQ MQ_QM1
//      {
//      CFSMount MQ_log_mnt

```

```
//          {
//          RVGPrimary MQ_RvgPri
//          }
//          CFSMount MQ_qmgrs_mnt
//          {
//          RVGPrimary MQ_RvgPri
//          }
//          }
//          }

group RVGgroup (
    SystemList = { RDQM-Vijay-RHL9-VM1 = 0, RDQM-Vijay-RHL9-VM2 = 1,
                  RDQM-Vijay-RHL9-VM3 = 2 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { RDQM-Vijay-RHL9-VM1, RDQM-Vijay-RHL9-VM2,
                     RDQM-Vijay-RHL9-VM3 }
)

CVMVolDg log_voldg (
    Critical = 0
    CVMDiskGroup = mqdg
    CVMVolume = { mq_log }
    CVMActivation = sw
)

CVMVolDg qmgrs_voldg (
    Critical = 0
    CVMDiskGroup = mqdg
    CVMVolume = { mq_qmgrs }
    CVMActivation = sw
)

DiskGroup MQ_Dg (
    DiskGroup = mqdg
)

RVGShared MQ_Rvg (
    RVG = mq_rvg
    DiskGroup = mqdg
)
```



```
requires group cvm online local firm
MQ_Rvg requires MQ_Dg
MQ_Rvg requires log_voldg
MQ_Rvg requires qmgrs_voldg

// resource dependency tree
//
//      group RVGgroup
//      {
//      RVGShared MQ_Rvg
//      {
//      DiskGroup MQ_Dg
//      CVMVolDg log_voldg
//      CVMVolDg qmgrs_voldg
//      }
//      }

group VVRGrp (
    SystemList = { RDQM-Vijay-RHL9-VM1 = 0, RDQM-Vijay-RHL9-VM2 = 1,
                  RDQM-Vijay-RHL9-VM3 = 2 }
    AutoStartList = { RDQM-Vijay-RHL9-VM1, RDQM-Vijay-RHL9-VM2,
                     RDQM-Vijay-RHL9-VM3 }
)

AzureIP ip-vvr-res-azure (
    PrivateIP = "10.113.55.92"
    NICDevice = eth0
    AzureAuthResName = azure-auth-res
)

IP ip-vvr-res (
    Device = eth0
    Address = "10.113.55.92"
    NetMask = "255.255.255.128"
)

NIC nic1 (
    Device = eth0
    NetworkHosts = { "10.113.55.92" }
)
```

```

Proxy proxyres (
    TargetResName = azure-auth-res
)

RVGLogowner logowner (
    RVG = mq_rvg
    DiskGroup = mqdg
)

requires group RVGgroup online local firm
ip-vvr-res requires ip-vvr-res-azure
ip-vvr-res-azure requires nic1
logowner requires ip-vvr-res

// resource dependency tree
//
//      group VVRGrp
//      {
//          RVGLogowner logowner
//          {
//              IP ip-vvr-res
//              {
//                  AzureIP ip-vvr-res-azure
//                  {
//                      NIC nic1
//                  }
//              }
//          }
//          Proxy proxyres
//      }

group cvm (
    SystemList = { RDQM-Vijay-RHL9-VM1 = 0, RDQM-Vijay-RHL9-VM2 = 1,
                  RDQM-Vijay-RHL9-VM3 = 2 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { RDQM-Vijay-RHL9-VM1, RDQM-Vijay-RHL9-VM2,
                     RDQM-Vijay-RHL9-VM3 }
)

CFSfsckd vxfsckd (

```

```

    )

CVMCluster cvm_clus (
    CVMClustName = clsrdqml
    CVMNodeId = { RDQM-Vijay-RHL9-VM1 = 0,
                  RDQM-Vijay-RHL9-VM2 = 1,
                  RDQM-Vijay-RHL9-VM3 = 2 }
    CVMTransport = gab
    CVMTimeout = 200
)

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
)

ProcessOnOnly vxattachd (
    Critical = 0
    PathName = "/bin/sh"
    Arguments = "- /usr/lib/vxvm/bin/vxattachd root"
    RestartLimit = 3
)

cvm_clus requires cvm_vxconfigd
vxfsckd requires cvm_clus

// resource dependency tree
//
//      group cvm
//      {
//      ProcessOnOnly vxattachd
//      CFSfsckd vxfsckd
//      {
//      CVMCluster cvm_clus
//      {
//      CVMVxconfigd cvm_vxconfigd
//      }
//      }
//      }
//      }

```

main.cf file at secondary cluster:

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"
include "RestServerTypes.cf"
include "SybaseTypes.cf"
include "WebSphereMQTypes.cf"

cluster ClsRDQMGO (
    SecInfo256 = "16:11:fe:b6:57:fb:32:b2:36:cf:cc:09:03:80:e4:04:
bc:6e:f1:a6:3b:d7:f9:2f:4f:a2:c3:a8:11:dc:43:84"
    ClusterAddress = "10.113.55.94"
    ProtocolNumber = 11000
    HacliUserLevel = COMMANDROOT
    IV256List = {
        "5f:d3:91:d8:6d:94:c9:db:e8:ff:aa:36:d9:7c:95:7a:f9:b7:
a9:1f:e2:be:b3:8d:a3:c4:13:bd:3f:f7:04:ac:2a:17:57:13:
b3:88:f3:41:08:f3:4d:81:ad:1f:01:e0:67:66:51:cd:0b:5b:
01:ed:93:b6:59:a6:47:c7:06:20" = "9c:f6:4d:45:de:28:
94:74:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
00:00:00:00:00:00:00:00" }
    )

remoteclass clsrqdm1 (
    ClusterAddress = "10.113.55.93"
)

heartbeat Icmp (
    ClusterList = { clsrqdm1 }
    Arguments @clsrqdm1 = { "10.113.55.93" }
)

system RDQM-Vijay-RHL9-VM4 (
)
```

```

system RDQM-Vijay-RHL9-VM5 (
)

system RDQM-Vijay-RHL9-VM6 (
)

group Azure-auth-grp (
    SystemList = { RDQM-Vijay-RHL9-VM4 = 0, RDQM-Vijay-RHL9-VM5 = 1,
                  RDQM-Vijay-RHL9-VM6 = 2 }
    Parallel = 1
)

AzureAuth azure-auth-res (
    SubscriptionId = 6940a326-xxxx-xxxx-xxxx-c1e9bbdf1d63
    ClientId = 941832d6-xxxx-xxxx-xxxx-6934336e0595
    SecretKey = "5f:d3:91:d8:6d:94:c9:db:e8:ff:aa:36:d9:7c:
95:7a:f9:b7:a9:1f:e2:be:b3:8d:a3:c4:13:bd:3f:f7:04:ac:
2a:17:57:13:b3:88:f3:41:08:f3:4d:81:ad:1f:01:e0:67:66:
51:cd:0b:5b:01:ed:93:b6:59:a6:47:c7:06:20"
    TenantId = fc8e13c0-xxxx-xxxx-xxxx-ca318e6cac32
)

Phantom phatom-res (
)

// resource dependency tree
//
//      group Azure-auth-grp
//      {
//      AzureAuth azure-auth-res
//      Phantom phatom-res
//      }

group ClusterService (
    SystemList = { RDQM-Vijay-RHL9-VM4 = 0, RDQM-Vijay-RHL9-VM5 = 1,
                  RDQM-Vijay-RHL9-VM6 = 2 }
    AutoStartList = { RDQM-Vijay-RHL9-VM4, RDQM-Vijay-RHL9-VM5,
                     RDQM-Vijay-RHL9-VM6 }
    OnlineRetryLimit = 3
    OnlineRetryInterval = 120

```

```

)

Application wac (
    StartProgram = "/opt/VRTSvcs/bin/wacstart"
    StopProgram = "/opt/VRTSvcs/bin/wacstop"
    MonitorProcesses = { "/opt/VRTSvcs/bin/wac" }
    RestartLimit = 3
)

```

```

AzureIP ip-gco-res-azure (
    PrivateIP = "10.113.55.94"
    NICDevice = eth0
    AzureAuthResName = azure-auth-res
)

```

```

IP gcoip (
    Device = eth0
    Address = "10.113.55.94"
    NetMask = "255.255.255.128"
)

```

```

NIC gconic (
    Device = eth0
)

```

```

gcoip requires ip-gco-res-azure
ip-gco-res-azure requires gconic
wac requires gcoip

```

```

// resource dependency tree
//
//      group ClusterService
//      {
//      Application wac
//      {
//      IP gcoip
//      {
//      AzureIP ip-gco-res-azure
//      {
//      NIC gconic
//      }
//      }
//      }
//      }

```

```
//      }
//      }
```

```
group MQGrp (
    SystemList = { RDQM-Vijay-RHL9-VM4 = 0, RDQM-Vijay-RHL9-VM5 = 1,
                  RDQM-Vijay-RHL9-VM6 = 2 }
    ClusterList = { clsrdqm1 = 0, ClsRDQMGC0 = 1 }
    AutoStartList = { RDQM-Vijay-RHL9-VM4, RDQM-Vijay-RHL9-VM5,
                    RDQM-Vijay-RHL9-VM6 }
    ClusterFailOverPolicy = Auto
)
```

```
CFSMount MQ_log_mnt (
    Critical = 0
    MountPoint = "/mqmnt/log"
    BlockDevice = "/dev/vx/dsk/mqdg/mq_log"
    MountOpt = "mincache=direct"
)
```

```
CFSMount MQ_qmgrs_mnt (
    Critical = 0
    MountPoint = "/mqmnt/qmgrs"
    BlockDevice = "/dev/vx/dsk/mqdg/mq_qmgrs"
    MountOpt = "mincache=direct"
)
```

```
RVGPrimary MQ_RvgPri (
    RvgResourceName = MQ_Rvg
    AutoResync = 1
)
```

```
WebSphereMQ MQ_QM1 (
    QueueManager = QM1
    MQVer = "9.3"
    MQInstallationPath = "/opt/mqm"
)
```

```
requires group RVGgroup online local firm
MQ_QM1 requires MQ_log_mnt
MQ_QM1 requires MQ_qmgrs_mnt
MQ_log_mnt requires MQ_RvgPri
MQ_qmgrs_mnt requires MQ_RvgPri
```

```
// resource dependency tree
//
//      group MQGrp
//      {
//      WebSphereMQ MQ_QM1
//      {
//      CFSMount MQ_log_mnt
//      {
//      RVGPrimary MQ_RvgPri
//      }
//      CFSMount MQ_qmgrs_mnt
//      {
//      RVGPrimary MQ_RvgPri
//      }
//      }
//      }

group RVGgroup (
    SystemList = { RDQM-Vijay-RHL9-VM4 = 0, RDQM-Vijay-RHL9-VM5 = 1,
                  RDQM-Vijay-RHL9-VM6 = 2 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { RDQM-Vijay-RHL9-VM4, RDQM-Vijay-RHL9-VM5,
                     RDQM-Vijay-RHL9-VM6 }
)

CVMVolDg log_voldg (
    Critical = 0
    CVMDiskGroup = mqdg
    CVMVolume = { mq_log }
    CVMActivation = sw
)

CVMVolDg qmgrs_voldg (
    Critical = 0
    CVMDiskGroup = mqdg
    CVMVolume = { mq_qmgrs }
    CVMActivation = sw
)
```



```
DiskGroup MQ_Dg (
    DiskGroup = mqdg
)

RVGShared MQ_Rvg (
    RVG = mq_rvg
    DiskGroup = mqdg
)

requires group cvm online local firm
MQ_Rvg requires MQ_Dg
MQ_Rvg requires log_voldg
MQ_Rvg requires qmgrs_voldg

// resource dependency tree
//
//      group RVGgroup
//      {
//      RVGShared MQ_Rvg
//          {
//          DiskGroup MQ_Dg
//          CVMVolDg log_voldg
//          CVMVolDg qmgrs_voldg
//          }
//      }

group VVRGrp (
    SystemList = { RDQM-Vijay-RHL9-VM4 = 0, RDQM-Vijay-RHL9-VM5 = 1,
                  RDQM-Vijay-RHL9-VM6 = 2 }
    AutoStartList = { RDQM-Vijay-RHL9-VM4, RDQM-Vijay-RHL9-VM5,
                     RDQM-Vijay-RHL9-VM6 }
)

AzureIP ip-vvr-res-azure (
    PrivateIP = "10.113.55.96"
    NICDevice = eth0
    AzureAuthResName = azure-auth-res
)

IP ip-vvr-res (
    Device = eth0
```

```

        Address = "10.113.55.96"
        NetMask = "255.255.255.128"
    )

    NIC nic1 (
        Device = eth0
        NetworkHosts = { "10.113.55.96" }
    )

    Proxy proxyres (
        TargetResName = azure-auth-res
    )

    RVGLogowner logowner (
        RVG = mq_rvg
        DiskGroup = mqdg
    )

    requires group RVGgroup online local firm
    ip-vvr-res requires ip-vvr-res-azure
    ip-vvr-res-azure requires nic1
    logowner requires ip-vvr-res

    // resource dependency tree
    //
    //     group VVRGrp
    //     {
    //         RVGLogowner logowner
    //         {
    //             IP ip-vvr-res
    //             {
    //                 AzureIP ip-vvr-res-azure
    //                 {
    //                     NIC nic1
    //                 }
    //             }
    //         }
    //         Proxy proxyres
    //     }

    group cvm (

```

```
SystemList = { RDQM-Vijay-RHL9-VM4 = 0, RDQM-Vijay-RHL9-VM5 = 1,
               RDQM-Vijay-RHL9-VM6 = 2 }
AutoFailOver = 0
Parallel = 1
AutoStartList = { RDQM-Vijay-RHL9-VM4, RDQM-Vijay-RHL9-VM5,
                  RDQM-Vijay-RHL9-VM6 }
)

CFSfsckd vxfsckd (
)

CVMCluster cvm_clus (
    CVMClustName = ClsRDQMGC0
    CVMNodeId = { RDQM-Vijay-RHL9-VM4 = 0,
                  RDQM-Vijay-RHL9-VM5 = 1,
                  RDQM-Vijay-RHL9-VM6 = 2 }
    CVMTransport = gab
    CVMTimeout = 200
)

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
)

ProcessOnOnly vxattachd (
    Critical = 0
    PathName = "/bin/sh"
    Arguments = "- /usr/lib/vxvm/bin/vxattachd root"
    RestartLimit = 3
)

cvm_clus requires cvm_vxconfigd
vxfsckd requires cvm_clus

// resource dependency tree
//
//      group cvm
//      {
//          ProcessOnOnly vxattachd
//          CFSfsckd vxfsckd
//          {
```

```
//          CVMCluster cvm_clus
//          {
//          CVMVxconfigd cvm_vxconfigd
//          }
//      }
//  }
```

Enabling the agent for WebSphere MQ to support IMF

This chapter includes the following topics:

- [About Intelligent Monitoring Framework](#)
- [Agent functions for the IMF functionality](#)
- [Attributes that enable IMF](#)
- [Before you enable the agent to support IMF](#)
- [Enabling the agent to support IMF](#)
- [Disabling intelligent resource monitoring](#)
- [Sample IMF configurations](#)

About Intelligent Monitoring Framework

With the IMF feature, VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring. You can enable or disable the intelligent resource monitoring functionality of the WebSphere MQ agent.

VCS process and mount-based agents use the AMF kernel driver that provides asynchronous event notifications to the agents that are enabled for IMF.

You can enable the WebSphere MQ agent for IMF, provided the following software versions are installed:

- Cluster Server (VCS) 5.1 SP1 or later
- Cluster Server agent for WebSphere MQ version 5.1.0.0 or later

Refer to the *Cluster Server Administrator's Guide* for more information about IMF notification module functions and administering the AMF kernel driver.

Benefits of IMF

IMF offers the following benefits:

- Performance
Enhances performance by reducing the monitoring of each resource at a default of 60 seconds for online resources, and 300 seconds for offline resources. IMF enables the agent to monitor a large number of resources with a minimal effect on performance.
- Faster detection
Asynchronous notifications would detect a change in the resource state as soon as it happens. Immediate notification enables the agent to take action at the time of the event.

Agent functions for the IMF functionality

If the WebSphere MQ agent is enabled for IMF support, the agent supports the following functions, in addition to the functions mentioned in the WebSphere MQ agent functions topic.

See [“WebSphere MQ agent functions”](#) on page 9.

imf_init

This function initializes the WebSphere MQ agent to interface with the AMF kernel driver, which is the IMF notification module for the agent for WebSphere MQ. This function runs when the agent starts up.

imf_getnotification

This function gets notifications about resource state changes. This function runs after the agent initializes with the AMF kernel module. This function continuously waits for notification and takes action on the resource upon notification.

imf_register

This function registers or unregisters resource entities with the AMF kernel module. This function runs for each resource after the resource goes into a steady state—online or offline.

Attributes that enable IMF

If the agent for WebSphere MQ is enabled for IMF support, the agent uses type-level attributes in addition to the agent-specific attributes.

See [“WebSphere MQ agent attributes”](#) on page 29.

IMF

This resource type-level attribute determines whether the WebSphere MQ agent must perform intelligent resource monitoring. You can also override the value of this attribute at the resource level.

This attribute includes the following keys:

Mode

Define this attribute to enable or disable intelligent resource monitoring.

The valid values are as follows:

- 0—Does not perform intelligent resource monitoring
- 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources
- 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources
- 3—Performs intelligent resource monitoring for both online and for offline resources.

Note: The agent supports intelligent resource monitoring for online resources only. Hence, Mode should be set to either 0 or 2.

Type and dimension: integer-association

Default: 0 for VCS 5.1 SP1, 3 for VCS 6.0 and later.

MonitorFreq

This key value specifies the frequency at which the agent invokes the monitor agent function. The value of this key is an integer.

Default: 1

You can set this key to a non-zero value for cases where the agent requires to perform both poll-based and intelligent resource monitoring.

If the value is 0, the agent does not perform poll-based process check monitoring.

After the resource registers with the AMF kernel driver, the agent calls the monitor agent function as follows:

- After every (MonitorFreq x MonitorInterval) number of seconds for online resources
- After every (MonitorFreq x OfflineMonitorInterval) number of seconds for offline resources

RegisterRetryLimit

If you enable intelligent resource monitoring, the agent invokes the `imf_register` agent function to register the resource with the AMF kernel driver.

The value of the `RegisterRetryLimit` key determines the number of times the agent must retry registration for a resource. If the agent cannot register the resource within the limit that is specified, then intelligent monitoring is disabled until the resource state changes or the value of the `Mode` key changes.

Default: 3.

IMFRegList

An ordered list of attributes whose values are registered with the IMF notification module.

Type and dimension: string-vector

Default: No default value

Note: The attribute values can be overridden at the resource level.

Before you enable the agent to support IMF

Before you enable the WebSphere MQ agent to support IMF, ensure that the AMF kernel module is loaded and AMF is configured. For details, refer to the

'Administering the AMF kernel driver' section of the *Cluster Server Administrator's Guide*. For details about the commands you can configure AMF using the `amfconfig -h` command.

Enabling the agent to support IMF

In order to enable the WebSphere MQ agent to support IMF, you must make the following configuration changes to the attributes of the agent:

- **AgentFile:** Set the AgentFile attribute to **Script51Agent** or **Script60Agent** as appropriate for your agent version
- **IMF Mode:** Set the IMF Mode attribute to **2**
- **IMFRegList:** Update the IMFRegList attribute

The following sections provide more information about the commands you can use to make these configuration changes, depending on whether VCS is in a running state or not.

Note: If you have upgraded VCS from an earlier version to version 5.1 SP1 or later, and you already have version 5.1.00 or later of the agent installed, ensure that you run the following commands to create appropriate symbolic links:

```
# cd /opt/VRTSagents/ha/bin/<resourceType>
# ln -s /opt/VRTSamf/imf/imf_getnotification imf_getnotification
# ln -s /opt/VRTSagents/ha/bin/<resourceType>/monitor imf_register
```

If VCS is in a running state

To enable the resource for IMF when VCS is in a running state

- 1 Make the VCS configuration writable.

```
# haconf -makerw
```

- 2 Run the following command to update the AgentFile attribute.

```
# hatype -modify <resourceType> AgentFile\
/opt/VRTSvcS/bin/Script51Agent
```

- 3 For VCS version 6.0 or later, run the following commands to add the IMF attributes:

```
# haattr -add -static <resourceType> IMF -integer -assoc Mode 0 \
MonitorFreq 1 RegisterRetryLimit 3

# haattr -add -static <resourceType> IMFRegList -string -vector
```

Note: Run these commands only once after you first enable IMF support for the agent.

- 4 Run the following command to update the IMF attribute.

```
# hatype -modify <resourceType> IMF Mode num MonitorFreq num
RegisterRetryLimit num
```

For example, to enable intelligent monitoring of online resources, with the MonitorFreq key set to 5, and the RegisterRetryLimit key is set to 3, run the following command:

```
# hatype -modify <resourceType> IMF Mode 2 MonitorFreq 5 \
RegisterRetryLimit 3
```

Note: The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

- 5 Run the following command to update the IMFRegList attribute:

```
# hatype -modify WebSphereMQ IMFRegList QueueManager MQUser\
MQVer MonitorListener
```

- 6 Save the VCS configuration.

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

- 7 If the agent is running, restart the agent.

For information on the commands you can use to restart the agent, see [Restarting the agent](#).

Restarting the agent

To restart the agent:

- 1 Run the following command to stop the agent forcefully:

```
# haagent -stop <resourceType> -force -sys <systemName>
```

Note: Stopping the agent forcefully eliminates the need to take the resource offline.

- 2 Run the following command to start the agent:

```
# haagent -start <resourceType> -sys <systemName>
```

If VCS is not in a running state

To change the agent type definition file when VCS is not in a running state

- 1 Update the AgentFile attribute.

```
static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
```

- 2 Update the IMF attribute.

The valid values for the Mode key of the IMF attribute are 0 (disabled) and 2 (online monitoring).

```
static int IMF{} = { Mode=num, MonitorFreq=num,  
RegisterRetryLimit=num }
```

For example, to update the IMF attribute such that the Mode key is set to 2, the MonitorFreq key is set to 5, and the RegisterRetryLimit key is set to 3:

```
static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3  
}
```

- 3 Update the IMFRegList attribute.

```
static str IMFRegList[] = { QueueManager, MQUser, MQVer,  
MonitorListener }
```

Disabling intelligent resource monitoring

To disable intelligent resource monitoring

- 1 Make the VCS configuration writable.

```
# haconf -makerw
```
- 2 To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

```
# haconf -makerw
```

```
# hatype -modify WebSphereMQ IMF -update Mode 0
```

- 3 To disable intelligent resource monitoring for a specific resource, run the following command:

```
# hares -override resource name IMF
```

```
# hares -modify resource name IMF -update Mode 0
```

- #### 4 Save the VCS configuration.

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

Sample IMF configurations

An example of a type definition file for an IMF-enabled WebSphere MQ agent is as follows.

In this example, the IMF-related attributes are set to the following values:

AgentFile	opt/VRTSvcs/bin/Script51Agent
-----------	-------------------------------

IMF{ Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }

IMFRegList[] { QueueManager, MQUser, MQVer, MonitorListener }

LevelTwoMonitorFreq	25
---------------------	----

[illegible]

```
        DisabledProcesses}

static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
static str IMFRegList[] = { QueueManager, MQUser, MQVer,
                           MonitorListener }

str ResLogLevel = INFO
str QueueManager
boolean CommandServer = 0
str MQUser = mqm
str MQVer = "6.0"
str EnvFile
int SecondLevelMonitor
str MonitorProgram
int MonitorListener = 0
str MQInstallationPath
keylist DisabledProcesses
)

```

A sample resource configuration from the `/etc/VRTSvcs/conf/config/main.cf` file is as follows:

```
WebSphereMQ Queue1 (
    QueueManager = Queue1
    CommandServer = 1
    MQVer = "7.0"
    MonitorListener = 1
)

```

Configuring the service groups for WebSphere MQ using the CLI

This chapter includes the following topics:

- [Before configuring the service groups for WebSphere MQ](#)
- [Configuring service groups for WebSphere MQ queue managers](#)
- [Creating service groups for WebSphere MQ under Solaris non-global zones](#)

Before configuring the service groups for WebSphere MQ

Before you configure the WebSphere MQ Queue Manager service group, you must:

- Verify that the Cluster Server components are installed and configured on all nodes in the cluster where you will configure the service group.
For more information on installing the components, refer to the *InfoScale Availability Installation Guide*.
- Verify that the Cluster Server agent for WebSphere MQ is installed on all nodes in the cluster.
See [“Installing the agent in a VCS environment”](#) on page 19.

Configuring service groups for WebSphere MQ queue managers

You can cluster WebSphere MQ queue managers in a clustered environment, and you can use the Cluster Server agent for WebSphere MQ to manage these components.

Configuring a WebSphere MQ resource

In a clustered environment, you can configure a WebSphere MQ resource using the following methods:

- [Active-passive configuration](#)
The active-passive configuration is an easier method of configuration. This method limits the configuration to one service group running a WebSphere MQ queue manager on a particular node at one time.
- [Active-active configuration](#)
The active-active configuration allows multiple service groups running WebSphere MQ queue managers on a particular node simultaneously. This configuration incurs additional complexity in configuration and maintenance.

Active-passive configuration

Use this configuration only where you need WebSphere MQ queue managers in a clustered environment.

On the node that hosts the service group, perform the following steps:

To configure a WebSphere MQ queue manager using active-passive configuration

- 1 Ensure that a file system is located on a shared disk.
This file system must be in the same service group in which the WebSphere MQ is to be created.
- 2 If required, copy the WebSphere MQ default files from the local copy in to the `/var/mqm` directory. This directory is a WebSphere MQ configuration item that is not changeable.
- 3 Mount the file system at the `/var/mqm` directory.

- 4 Use the WebSphere MQ tools to create the WebSphere MQ queue manager. Refer to the WebSphere MQ documentation for details.
- 5 Define this WebSphere MQ queue manager as a resource in the service group. See [“Sample service group configurations”](#) on page 123.

You can now create additional queue managers on the same node on which the service group is currently online.

Ensure that you always define the additional queue manager as a cluster server resource in the same service group where other queue managers are defined.

Active-active configuration

In an active-active configuration, you can configure each WebSphere MQ queue manager in a separate service group and each queue manager can fail over independent of each other.

This configuration provides the flexibility that some applications may require.

On the node that hosts the service group to which the WebSphere MQ queue manager belongs, perform the following steps:

To configure a WebSphere MQ queue manager using active-active configuration

- 1 Use the WebSphere MQ tools to create the WebSphere MQ queue managers that you require. Refer to the WebSphere MQ documentation for details.
- 2 Create a file system for each WebSphere MQ on the separate shared disk. Add each file system to a separate service group.
- 3 Add the queue manager resource to the service group that contains the storage resources used by this queue manager.
- 4 If the listener for the queue manager uses a virtual IP, add the networking resources such as IP and NIC to the same service group as that of corresponding queue manager.

See [“Sample configuration in a VCS virtual environment with VMware”](#) on page 112.

Configuring a WebSphere MQ listener

A WebSphere MQ queue manager uses a Listener to listen for requests on a specific IP address. You can configure a Listener resource in the cluster using a bundled application agent. An example listener resource configuration is shown as follows. In this example, the virtual IP address is set to 1.2.3.4 and the queue manager name is venus.veritas.

See [“WebSphere MQ agent attributes”](#) on page 29.

You can enable the MonitorListener attribute, if the listener is configured to start automatically when the queue manager starts.

You can enable the MonitorListener attribute, to start Listener when WebSphere MQ resource is online.

You can replace these values with the virtual IP address and queue manager name defined within the cluster.

```
Application was4WSMQ_listen
(
  User          = mqm
  StartProgram  = "/opt/mqm/bin/runmqslr -t tcp
                  -i 1.2.3.4 -m venus.veritas &"
  StopProgram   = "/opt/mqm/bin/endmqslr -m venus.veritas"
  MonitorProcesses = {"/opt/mqm/bin/runmqslr -t tcp
                      -i 1.2.3.4 -m venus.veritas" }
```

For details about the WebSphere MQ listener, refer to the WebSphere MQ documentation.

Creating service groups for WebSphere MQ under Solaris non-global zones

To configure zones on each cluster node

1 Set up the non-global zone configuration.

```
hazonesetup servicegroup_name zoneres_name zone_name password
systems
```

For example:

```
hazonesetup -g servicegroup_name -r zoneres_name -z zone_name
-p password -s systems
```

2 Verify the non-global zone configuration.

```
hazoneverify servicegroup_name
```

3 Whenever you make a change that affects the zone configuration, run the hazonesetup command to reconfigure the zones in VCS.

4 Make sure that the zone configuration files are consistent on all nodes at all times. The file is located at /etc/zones/zone_name.xml.

- 5** Make sure that the application is identical on all nodes. If you update the application configuration on one node, apply the same updates to all nodes.
- 6** Configure the service groups for WebSphere MQ.

Troubleshooting the agent for WebSphere MQ

This chapter includes the following topics:

- [Using the correct software and operating system versions](#)
- [Meeting prerequisites](#)
- [Configuring WebSphere MQ Queue Manager resources](#)
- [Starting the WebSphere MQ Queue Manager instance outside a cluster](#)
- [Monitoring WebSphere MQ queue manager processes](#)
- [Stopping WebSphere MQ queue manager processes forcefully](#)
- [Reviewing error log files](#)
- [Troubleshooting the configuration for IMF](#)

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

For information on the software versions that the agent for WebSphere MQ supports, see the Veritas Services and Operations Readiness Tools (SORT) site:

<https://sort.veritas.com/agents>.

Meeting prerequisites

Before installing the agent for WebSphere MQ, ensure that the following prerequisites are met.

For example, you must install the ACC library on VCS before installing the agent for WebSphere MQ.

See [“Before you install the Cluster Server agent for WebSphere MQ”](#) on page 14.

Note: For information about the prerequisites for IMF and for other IMF-related troubleshooting information: See [“Troubleshooting the configuration for IMF”](#) on page 105.

Configuring WebSphere MQ Queue Manager resources

Before using WebSphere MQ Queue Manager resources, ensure that you configure the resources properly. For a list of attributes used to configure all WebSphere MQ Queue Manager resources, refer to the agent attributes.

Starting the WebSphere MQ Queue Manager instance outside a cluster

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 WebSphere MQ Queue Manager instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the WebSphere MQ Queue Manager instance outside the cluster framework.

Note: Use the same parameters that the resource attributes define within the cluster framework while restarting the resource outside the cluster framework.

A sample procedure to start a WebSphere MQ instance outside the cluster framework, is described as follows.

To restart the WebSphere MQ Queue Manager outside the framework

- 1 Log in to the WebSphere MQ Queue Manager as an MQUser.

```
# su - MQUser
```

- 2 Start the WebSphere MQ Queue Manager.

```
# strmqm QueueManagerName
```

If the WebSphere MQ Queue Manager works properly outside the cluster framework, you can then attempt to implement the Queue Manager within the cluster framework.

Monitoring WebSphere MQ queue manager processes

The agent for WebSphere MQ monitors the following processes:

MQ 5.3

```
"amqhasmx X_QUEUE_MANAGER_X( |\\$) ",
"amqzllp0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzlaa0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqrrmfa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqchi .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzdmaa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzfuma .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzxma0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
```

MQ 6.0

```
"amqrrmfa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqchi .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzdmaa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzfuma .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzxma0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmuc0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmur0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
```

MQ 7.0

```
"amqrrmfa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqchi .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzdmaa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzfuma .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzxma0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmuc0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmur0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqslr .*-m *X_QUEUE_MANAGER_X( |\\$) ",
```

MQ 8.0 and later	<pre>"amqrrmfa .*-m *X_QUEUE_MANAGER_X(\\\$) ", "runmqchi .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzfuma .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzxma0 .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzmuc0 .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzmur0 .*-m *X_QUEUE_MANAGER_X(\\\$) ", "runmqlsr .*-m *X_QUEUE_MANAGER_X(\\\$) ",</pre>
---------------------	---

Stopping WebSphere MQ queue manager processes forcefully

As per IBM recommendations, when an attempt to gracefully stop the queue manager fails, the agent for WebSphere MQ kills the processes in the following order:

MQ 5.3	<pre>"amqhasmx X_QUEUE_MANAGER_X(\\\$) ", "amqzllp0 .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzlaa0 .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqrrmfa .*-m *X_QUEUE_MANAGER_X(\\\$) ", "runmqchi .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzdmaa .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzfuma .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqzxma0 .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqpcsea .*-m *X_QUEUE_MANAGER_X(\\\$) ", "amqharmx .*-m *X_QUEUE_MANAGER_X(\\\$) ", "runmqlsr .*-m *X_QUEUE_MANAGER_X(\\\$) ",</pre>
--------	---

MQ 6.0

```
"amqzmuc0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzxma0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzfuma .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzlaa0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzlsa0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmgr0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmur0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqrmppa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqrmfa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzdmaa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqpcsea *X_QUEUE_MANAGER_X( |\\$) ",
"amqhasmx X_QUEUE_MANAGER_X( |\\$) ",
"amqzllp0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqchi .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqharmx .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqfqpub .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqfcxba .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqcrsta .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqsc *X_QUEUE_MANAGER_X( |\\$) ",
"runmqlsr .*-m *X_QUEUE_MANAGER_X( |\\$) ",
```

MQ 7.0 and later

```
"amqzmuc0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzxma0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzfuma .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzlaa0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzlsa0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmuf0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmur0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzmgr0 .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqfqpub .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqfcxba .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqrmppa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqcrsta .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqcrs6b .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqrmfa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqzdmaa .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqpcsea *X_QUEUE_MANAGER_X( |\\$) ",
"runmqtrm .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqdlq .*X_QUEUE_MANAGER_X( |\\$) ",
"runmqchi .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqlsr .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqxssvn .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"amqztrcn .*-m *X_QUEUE_MANAGER_X( |\\$) ",
"runmqsc *X_QUEUE_MANAGER_X( |\\$) ",
```

Reviewing error log files

If you face problems while using WebSphere MQ Queue Manager or the agent for WebSphere MQ, use the log files described in this section to investigate the problems.

Using WebSphere MQ log files

If a WebSphere MQ Queue Manager is facing problems, you can access the server log files to further diagnose the problem. The WebSphere MQ Queue Manager log files are located in the *<Queue Manager Home>/qmgrs/<Queue Manager Name>/errors* directory.

Reviewing cluster log files

In case of problems while using the agent for WebSphere MQ, you can also access the engine log file for more information about a particular resource.

The VCS engine log file is at `/var/VRTSvcS/log/engine_A.log`.

Using trace level logging

The `ResLogLevel` attribute controls the level of logging that is written in a cluster log file for each WebSphere MQ Queue Manager 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. Veritas recommends that you localize the `ResLogLevel` attribute for a particular resource.

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 resource for which you want to enable detailed logging.
- 2 Localize the `ResLogLevel` attribute for the identified resource:

```
# hares -local Resource_Name ResLogLevel
```

- 3 Set the `ResLogLevel` attribute to `TRACE` for the identified resource:

```
# hares -modify Resource_Name ResLogLevel TRACE -sys SysA
```

- 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:

```
# hares -modify Resource_Name ResLogLevel INFO -sys SysA
```

- 8 Save the configuration changes.

```
# haconf -dump
```

- 9 Review the contents of the log file.

Use the time noted in the previous steps to diagnose the problem.

You can also contact Veritas support for more help.

To enable debug logs for all resources of type WebSphereMQ

- ◆ Enable the debug log.

```
# hatype -modify WebSphereMQ LogDbg DBG_5
```

To override the LogDbg attribute at resource level

- ◆ Override the LogDbg attribute at the resource level and enable the debug logs for the specific resource.

```
# hares -override WebSphereMQ LogDbg
# hares -modify WebSphereMQ LogDbg DBG_5
```

Troubleshooting the configuration for IMF

If you face problems with the IMF configuration or functionality, consider the following:

- Ensure that the following attributes are configured with appropriate values.

- AgentFile
- IMF
- IMFRegList

If IMFRegList is not configured correctly, the WebSphere MQ resources that have been registered for IMF get unregistered every time the monitor function is run.

- If you have configured the required attributes to enable the WebSphere MQ agent for IMF, but the agent is still not IMF-enabled, restart the agent. The `imf_init` function runs only when the agent starts up, so when you restart the agent, `imf_init` runs and initializes the WebSphere MQ agent to interface with the AMF kernel driver.

- You can run the following command to check the value of the `MonitorMethod` attribute and to verify that a resource is registered for IMF.

```
# hares -value resource MonitorMethod system
```

The `MonitorMethod` attribute specifies the monitoring method that the agent uses to monitor the resource:

- Traditional—Poll-based resource monitoring
- IMF—Intelligent resource monitoring

- You can use the `amfstat` to see a list of registered PIDs for a WebSphereMQ resource.

Following is a sample output for the Queue Manager 'testQM'.

The `ps -ef` command output shows the Queue Manager process.

```
# ps -ef | grep -i testQM
mqm 10085 10075 1 13:38:30 ? 0:00 amqzlaa0 -mTestQM
                                -fip0
mqm 10080 10075 1 13:38:28 ? 0:00 amqzmur0 -m TestQM
mqm 10083 10075 1 13:38:28 ? 0:00 /opt/mqm/bin/
                                amqzdmaa -m TestQM
mqm 10076 10075 1 13:38:27 ? 0:00 /opt/mqm/bin/
                                amqzfuma -m TestQM
mqm 10088 10081 1 13:38:30 ? 0:00 amqfqpub -mTestQM
mqm 10086 10084 0 13:38:30 ? 0:00 /opt/mqm/bin/
                                runmqchi -m TestQM
                                -q SYSTEM.CHANNEL.
                                INITQ -r
mqm 10089 10088 1 13:38:30 ? 0:00 amqfcxba -m TestQM
mqm 10077 10075 1 13:38:27 ? 0:00 amqzmuc0 -m TestQM
mqm 10082 10075 0 13:38:28 ? 0:00 /opt/mqm/bin/
                                amqgrmf0 -m TestQM
                                -t2332800 -s2592000
                                -p2592000 -g5184000
                                -c3600
mqm 10084 10075 1 13:38:29 ? 0:00 /opt/mqm/bin/
                                amqzmgr0 -m TestQM
mqm 10075 1 1 13:38:27 ? 0:00 amqzxma0 -m TestQM
mqm 10081 10075 1 13:38:28 ? 0:00 amqzmuf0 -m TestQM
```

```
mqm 10087 10084 1 13:38:30 ? 0:00 /opt/mqm/bin/
                                     amqpcsea TestQM
```

The `amfstat` command shows the Queue Manager PIDs monitored by the WebSphereMQ agent.

```
# amfstat
AMF Status Report

Registered Reapers (3):
=====
      RID      PID      MONITOR TRIGG      REAPER
163      16417      7          0          WebSphereMQ

Process ONLINE Monitors (7):
=====
      RID      R_RID      PID      GROUP
165      163      10082      TestQM
166      163      10086      TestQM
167      163      10083      TestQM
168      163      10076      TestQM
169      163      10075      TestQM
170      163      10077      TestQM
171      163      10080      TestQM
```

- Run the following command to set the `ResLogLevel` attribute to `TRACE`. When you set `ResLogLevel` to `TRACE`, the agent logs messages in the `WebSphereMQ_A.log` file.

```
# hares -modify ResourceName ResLogLevel TRACE
```

For more information about the `ResLogLevel` attribute, See [Table 3-1](#) on page 29.

- Run the following command to view the content of the AMF in-memory trace buffer.
- If you have upgraded to VCS version 5.1 SP1, from an earlier VCS version, and if you already have WebSphere MQ agent version 5.1.9.0 or later installed, ensure that the appropriate symbolic links have been created. You can run the following commands to create appropriate symbolic links:

```
# cd /opt/VRTSagents/ha/bin/WebSphereMQ
# ln -s /opt/VRTSamf/imf/imf_getnotification imf_getnotification
# ln -s /opt/VRTSagents/ha/bin/WebSphereMQ/monitor imf_register
```

Known issues

This release of the agent for WebSphere MQ has the following known issues:

Problem

An error message might appear when you run the `hares -offline` command to take a resource offline.

Description

When a resource is taken offline, it is unregistered from the AMF module. However, the `imf_register` function attempts to unregister the resource again.

This results in the following error message from the engine log.

```
VCS ERROR V-16-2-13710 Resource(Queue1) - imf_register entry point
failed with exit code(1)
```

The following message is logged in the agent log:

```
V-16-55000-10209 Commandline [/opt/VRTSsamf/bin/amfregister -u
-rWebSphereMQ -g Queue1 ] provided a non-zero exit code --This does
not necessarily indicate a problem ... (Perl's OS error variable
prior to the command-pipe close was [], and after the close was [])
) VCSagentFW:messageEngineLog:[AMF amfregisterNOTICEIgnoring the
group unregister request; group named \"Queue1\" not found]
```

Workaround

It is safe to ignore this error message.

Sample Configurations

This appendix includes the following topics:

- [About sample configurations for the agents for WebSphere MQ](#)
- [Sample agent type definition for WebSphere MQ](#)
- [Sample configuration in a VCS environment](#)
- [Sample configuration in a VCS virtual environment with VMware](#)
- [Sample configuration for Multi-Instance mode](#)
- [Sample service group configurations](#)

About sample configurations for the agents for WebSphere MQ

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for WebSphere MQ. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition for WebSphere MQ

After importing the agent types into the cluster, if you save the configuration on your system disk using the `haconf -dump` command, you can find the `WebSphereMQTypes.cf` file in the `/etc/VRTSvcs/conf/config` cluster configuration directory.

Examples of agent type definition files for different versions of VCS are as follows:

For VCS 5.x and VCS 6.0

```
type WebSphereMQ (
    static boolean AEPTIMEOUT = 1
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/WebSphereMQ"
    static str ArgList[] = { ResLogLevel, State, IState, QueueManager,
        CommandServer, MQUser, MQVer, StartOptions, StopOptions, EnvFile,
        SecondLevelMonitor, MonitorProgram, MonitorListener, CleanIPC,
        MQInstallationPath, TransitionWaitLimit, DisabledProcesses,
        ServiceName, SystemDEnvFile }
    str ResLogLevel = INFO
    str QueueManager
    boolean CommandServer = 0
    str MQUser = mqm
    str MQVer = "6.0"
    str StartOptions
    str StopOptions = "-i"
    str EnvFile
    int SecondLevelMonitor
    str MonitorProgram
    boolean CleanIPC = 0
    int MonitorListener = 0
    str MQInstallationPath
    keylist DisabledProcesses
    int TransitionWaitLimit = 10
    str ServiceName
    str SystemDEnvFile
)
```

Sample configuration in a VCS environment

An excerpt from the `main.cf` file that includes a WebSphere MQ resource follows.

```
group WASMQ_Solaris (
    SystemList = { system_A = 0, system_B = 1 }
)
    DiskGroup DG_OPT (
        DiskGroup = WAS
    )
    DiskGroup DG_VAR (
        DiskGroup = WAS
    )
)
```

```

    )

Mount Mount_OPT (
    MountPoint = "/opt/mqm"
    BlockDevice = "/dev/vx/dsk/WAS/MQ_Opt"
    FSType = vxfs
    FsckOpt = "-y"
)

Mount Mount_VAR (
    MountPoint = "/var/mqm"
    BlockDevice = "/dev/vx/dsk/WAS/MQ_Vol"
    FSType = vxfs
    FsckOpt = "-y"
)

Volume Volum_OPT (
    Volume = MQ_Opt
    DiskGroup = WAS
)

Volume Volume_VAR (
    Volume = MQ_Vol
    DiskGroup = WAS
)

WebSphereMQ WASMQ (
    QueueManager = MQ1
    CommandServer = 1
    MQVer = "7.0"
    SecondLevelMonitor = 5
    MonitorProgram = "/ibm/mq/myMonitor.sh"
    MonitorListener = 1
    MQInstallationPath = "/opt/customLocation"
)

Mount_OPT requires Volum_OPT

Mount_VAR requires Volume_VAR

Volum_OPT requires DG_OPT

Volume_VAR requires DG_VAR

```

```
WASMQ requires Mount_OPT

WASMQ requires Mount_VAR


// resource dependency tree
//      group WASMQ_Solaris
//      {
//      WebSphereMQ WASMQ
//      {
//      Mount Mount_OPT
//      {
//      Volume Volum_OPT
//      {
//      DiskGroup DG_OPT
//      }
//      }
//      Mount Mount_VAR
//      {
//      Volume Volume_VAR
//      {
//      DiskGroup DG_VAR
//      }
//      }
//      }
//      }
```

Sample configuration in a VCS virtual environment with VMware

An excerpt from the `main.cf` file in a scenario where the WebSphere MQ queue manager is configured using an active-active configuration is as follows:

```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"
include "WebSphereMQTypes.cf"

cluster Cluster_62228 (
```



```
SecureClus = 1
)

system sles11sp2mq1 (
)

group MQ_1 (
  SystemList = { sles11sp2mq1 = 0 }
  UserAssoc = { Type = WebSphereMQ,
    Name = "WebSphere MQ: Queue manager" }
  AutoStartList = { sles11sp2mq1 }
)

LVMLogicalVolume lvol_log1Vol_Qname1_log (
  LogicalVolume = log1Vol
  VolumeGroup = Qname1_log
)

LVMLogicalVolume lvol_qmgr1Vol_Qname1_qmgr (
  LogicalVolume = qmgr1Vol
  VolumeGroup = Qname1_qmgr
)

LVMVolumeGroup volg_Qname1_log (
  VolumeGroup = Qname1_log
)

LVMVolumeGroup volg_Qname1_qmgr (
  VolumeGroup = Qname1_qmgr
)

Mount mnt_sym_vcs_qml_2 (
  MountPoint = "/var/mqm/qmgrs/sym!vcs!qml"
  BlockDevice = "/dev/mapper/Qname1_qmgr-qmgr1Vol"
  FSType = ext3
  MountOpt = rw
  FsckOpt = "-y"
)

Mount mnt_sym_vcs_qml_3 (
  MountPoint = "/var/mqm/log/sym!vcs!qml"
  BlockDevice = "/dev/mapper/Qname1_log-log1Vol"
  FSType = ext3
```

```
MountOpt = rw
FsckOpt = "-y"
)

VMwareDisks VMwareDisk_1 (
  ESXDetails = { vcs1x658 = "root=BPHnEPeJBhCHdHE" }
  DiskPaths = {
    "6000C29c-491e-0d0f-835d-a25f9bbf685f:[SAPMQ] sles11sp2mq1
    /sles11sp2mq1_1.vmdk" = "0:3" }
  VMRegisterWait = 5
)

WebSphereMQ sym_vcs_qm1 (
  QueueManager = "sym!vcs!qm1"
  MQVer = "7.0"
  MonitorListener = 1
)

lvol_log1Vol_Qname1_log requires volg_Qname1_log
lvol_qmgr1Vol_Qname1_qmgr requires volg_Qname1_qmgr
mnt_sym_vcs_qm1_2 requires lvol_qmgr1Vol_Qname1_qmgr
mnt_sym_vcs_qm1_3 requires lvol_log1Vol_Qname1_log
sym_vcs_qm1 requires mnt_sym_vcs_qm1_2
sym_vcs_qm1 requires mnt_sym_vcs_qm1_3
volg_Qname1_qmgr requires VMwareDisk_1

// resource dependency tree
//
// group MQ_1
// {
//   WebSphereMQ sym_vcs_qm1
//   {
//     Mount mnt_sym_vcs_qm1_2
//     {
//       LVMLLogicalVolume lvol_qmgr1Vol_Qname1_qmgr
//       {
//         LVMVolumeGroup volg_Qname1_qmgr
//         {
//           VMwareDisks VMwareDisk_1
//         }
//       }
//     }
//   }
//   Mount mnt_sym_vcs_qm1_3
```

```
//      {
//      LVMLogicalVolume lvol_log1Vol_Qname1_log
//      {
//      LVMVolumeGroup volg_Qname1_log
//      }
//      }
//  }
// }

group MQ_2 (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = WebSphereMQ,
        Name = "WebSphere MQ: Queue manager" }
    AutoStartList = { sles11sp2mq1 }
)

LVMLogicalVolume lvol_log2Vol_Qname2_log (
    LogicalVolume = log2Vol
    VolumeGroup = Qname2_log
)

LVMLogicalVolume lvol_qmgr2Vol_Qname2_qmgr (
    LogicalVolume = qmgr2Vol
    VolumeGroup = Qname2_qmgr
)

LVMVolumeGroup volg_Qname2_log (
    VolumeGroup = Qname2_log
)

LVMVolumeGroup volg_Qname2_qmgr (
    VolumeGroup = Qname2_qmgr
)

Mount mnt_sym_vcs_qm2_2 (
    MountPoint = "/var/mqm/qmgrs/sym!vcs!qm2"
    BlockDevice = "/dev/mapper/Qname2_qmgr-qmgr2Vol"
    FSType = ext3
    MountOpt = rw
    FsckOpt = "-y"
)

Mount mnt_sym_vcs_qm2_3 (
```

```

MountPoint = "/var/mqm/log/sym!vcs!qm2"
BlockDevice = "/dev/mapper/Qname2_log-log2Vol"
FSType = ext3
MountOpt = rw
FsckOpt = "-y"
)

VMwareDisks VMwareDisk_2 (
  ESXDetails = { vcs1x658 = "root=BPHEPeJBhCHdHE" }
  DiskPaths = {
    "6000C29d-1034-f3dc-abbcd9b166e4cf38:[SAPMQ] sles11sp2mq1
    /sles11sp2mq1_2.vmdk" = "0:5" }
  VMRegisterWait = 5
)

WebSphereMQ sym_vcs_qm2 (
  QueueManager = "sym!vcs!qm2"
  MQVer = "7.0"
)

lvol_log2Vol_Qname2_log requires volg_Qname2_log
lvol_qmgr2Vol_Qname2_qmgr requires volg_Qname2_qmgr
mnt_sym_vcs_qm2_2 requires lvol_qmgr2Vol_Qname2_qmgr
mnt_sym_vcs_qm2_3 requires lvol_log2Vol_Qname2_log
sym_vcs_qm2 requires mnt_sym_vcs_qm2_2
sym_vcs_qm2 requires mnt_sym_vcs_qm2_3
volg_Qname2_qmgr requires VMwareDisk_2

// resource dependency tree
//
// group MQ_2
// {
//   WebSphereMQ sym_vcs_qm2
//   {
//     Mount mnt_sym_vcs_qm2_2
//     {
//       LVMLLogicalVolume lvol_qmgr2Vol_Qname2_qmgr
//       {
//         LVMVolumeGroup volg_Qname2_qmgr
//         {
//           VMwareDisks VMwareDisk_2
//         }
//       }
//     }
//   }
// }

```

```
//      }
//      Mount mnt_sym_vcs_qm2_3
//      {
//          LVMLLogicalVolume lvol_log2Vol_Qname2_log
//          {
//              LVMVolumeGroup volg_Qname2_log
//          }
//      }
//  }
// }

group MQ_3 (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = WebSphereMQ,
        Name = "WebSphere MQ: Queue manager" }
    AutoStartList = { sles11sp2mq1 }
)

LVMLLogicalVolume lvol_log3Vol_Qname3_log (
    LogicalVolume = log3Vol
    VolumeGroup = Qname3_log
)

LVMLLogicalVolume lvol_qmgr3Vol_Qname3_qmgr (
    LogicalVolume = qmgr3Vol
    VolumeGroup = Qname3_qmgr
)

LVMVolumeGroup volg_Qname3_log (
    VolumeGroup = Qname3_log
)

LVMVolumeGroup volg_Qname3_qmgr (
    VolumeGroup = Qname3_qmgr
)

Mount mnt_sym_vcs_qm3_2 (
    MountPoint = "/var/mqm/qmgrs/sym!vcs!qm3"
    BlockDevice = "/dev/mapper/Qname3_qmgr-qmgr3Vol"
    FSType = ext3
    MountOpt = rw
    FsckOpt = "-y"
)
```

```
Mount mnt_sym_vcs_qm3_3 (
    MountPoint = "/var/mqm/log/sym!vcs!qm3"
    BlockDevice = "/dev/mapper/Qname3_log-log3Vol"
    FSType = ext3
    MountOpt = rw
    FsckOpt = "-y"
)

VMwareDisks VMwareDisk_3 (
    ESXDetails = { vcs!x658 = "root=BPHnEPeJBhChdHE" }
    DiskPaths = {
        "6000C297-14a0-42b4-ffca-9a00727b7952:[SAPMQ] sles11sp2mq1
        /sles11sp2mq1_3.vmdk" = "0:6" }
    VMRegisterWait = 5
)

WebSphereMQ sym_vcs_qm3 (
    QueueManager = "sym!vcs!qm3"
    MQVer = "7.0"
)

lvol_log3Vol_Qname3_log requires volg_Qname3_log
lvol_qmgr3Vol_Qname3_qmgr requires volg_Qname3_qmgr
mnt_sym_vcs_qm3_2 requires lvol_qmgr3Vol_Qname3_qmgr
mnt_sym_vcs_qm3_3 requires lvol_log3Vol_Qname3_log
sym_vcs_qm3 requires mnt_sym_vcs_qm3_2
sym_vcs_qm3 requires mnt_sym_vcs_qm3_3
volg_Qname3_qmgr requires VMwareDisk_3

// resource dependency tree
//
// group MQ_3
// {
//     WebSphereMQ sym_vcs_qm3
//     {
//         Mount mnt_sym_vcs_qm3_2
//         {
//             LVMLogicalVolume lvol_qmgr3Vol_Qname3_qmgr
//             {
//                 LVMVolumeGroup volg_Qname3_qmgr
//                 {
//                     VMwareDisks VMwareDisk_3
```

```
//          }
//      }
//  }
//      Mount mnt_sym_vcs_qm3_3
//      {
//          LVMLogicalVolume lvol_log3Vol_Qname3_log
//          {
//              LVMVolumeGroup volg_Qname3_log
//          }
//      }
//  }
// }

group MQ_4 (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = WebSphereMQ,
        Name = "WebSphere MQ: Queue manager" }
    AutoStartList = { sles11sp2mq1 }
)

LVMLogicalVolume lvol_log4Vol_Qname4_log (
    LogicalVolume = log4Vol
    VolumeGroup = Qname4_log
)

LVMLogicalVolume lvol_qmgr4Vol_Qname4_qmgr (
    LogicalVolume = qmgr4Vol
    VolumeGroup = Qname4_qmgr
)

LVMVolumeGroup volg_Qname4_log (
    VolumeGroup = Qname4_log
)

LVMVolumeGroup volg_Qname4_qmgr (
    VolumeGroup = Qname4_qmgr
)

Mount mnt_sym_vcs_qm4_2 (
    MountPoint = "/var/mqm/qmgrs/sym!vcs!qm4"
    BlockDevice = "/dev/mapper/Qname4_qmgr-qmgr4Vol"
    FSType = ext3
    MountOpt = rw
```

```
FsckOpt = "-y"
)

Mount mnt_sym_vcs_qm4_3 (
    MountPoint = "/var/mqm/log/sym!vcs!qm4"
    BlockDevice = "/dev/mapper/Qname4_log-log4Vol"
    FSType = ext3
    MountOpt = rw
    FsckOpt = "-y"
)

VMwareDisks VMwareDisk_4 (
    ESXDetails = { vcs!x658 = "root=BPHnEPeJBhCHdHE" }
    DiskPaths = {
        "6000C29e-cf9a-022e-83e3-eaba577357c1:[SAPMQ] sles11sp2mq1
        /sles11sp2mq1_4.vmdk" = "0:8" }
    VMRegisterWait = 5
)

WebSphereMQ sym_vcs_qm4 (
    QueueManager = "sym!vcs!qm4"
    MQVer = "7.0"
)

lvol_log4Vol_Qname4_log requires volg_Qname4_log
lvol_qmgr4Vol_Qname4_qmgr requires volg_Qname4_qmgr
mnt_sym_vcs_qm4_2 requires lvol_qmgr4Vol_Qname4_qmgr
mnt_sym_vcs_qm4_3 requires lvol_log4Vol_Qname4_log
sym_vcs_qm4 requires mnt_sym_vcs_qm4_2
sym_vcs_qm4 requires mnt_sym_vcs_qm4_3
volg_Qname4_qmgr requires VMwareDisk_4

// resource dependency tree
//
// group MQ_4
// {
//     WebSphereMQ sym_vcs_qm4
//     {
//         Mount mnt_sym_vcs_qm4_2
//         {
//             LVMLLogicalVolume lvol_qmgr4Vol_Qname4_qmgr
//             {
//                 LVMVolumeGroup volg_Qname4_qmgr
```



```

//          {
//          VMwareDisks VMwareDisk_4
//          }
//      }
//      }
//      Mount mnt_sym_vcs_qm4_3
//      {
//          LVMLogicalVolume lvol_log4Vol_Qname4_log
//          {
//          LVMVolumeGroup volg_Qname4_log
//          }
//      }
//      }
//      }
//  }

group VCSInfraSG (
    SystemList = { sles11sp2mq1 = 0 }
    UserAssoc = { Type = "vcs internal",
        Name = "VCS Infrastructure service group" }
    Parallel = 1
    AutoStartList = { sles11sp2mq1 }
    OnlineRetryLimit = 5
)

Process VCSNotifySinkRes (
    PathName = "/opt/VRTSvcs/portal/admin/notify_sink"
)

// resource dependency tree
//
// group VCSInfraSG
// {
// Process VCSNotifySinkRes
// }

```

Sample configuration for Multi-Instance mode

An excerpt from the `main.cf` file for a Multi-Instance configuration is as follows:

```

group WASMQ_Solaris (
    SystemList = { system_A = 0, system_B = 1 }
    Parallel = 1
    AutoStartList = { system_A }

```

```
)
DiskGroup DG_OPT (
    DiskGroup = WAS
)
DiskGroup DG_VAR (
    DiskGroup = WAS
)

Mount Mount_OPT (
    MountPoint = "/opt/mqm"
    BlockDevice = "/dev/vx/dsk/WAS/MQ_Opt"
    FSType = vxfs
    MountOpt = "-o cluster"
    FsckOpt = "-y"
)
Mount Mount_VAR (
    MountPoint = "/var/mqm"
    BlockDevice = "/dev/vx/dsk/WAS/MQ_Vol"
    FSType = vxfs
    MountOpt = "-o cluster"
    FsckOpt = "-y"
)

Volume Volum_OPT (
    Volume = MQ_Opt
    DiskGroup = WAS
)
Volume Volume_VAR (
    Volume = MQ_Vol
    DiskGroup = WAS
)

WebSphereMQ WASMQ (
    QueueManager = MQ1
    CommandServer = 1
    MQUser = mqm
    MQVer = "9.1"
    StartOptions = "-x"
    SecondLevelMonitor = 5
    EnvFile = "/MQ/setEnv.sh"
    MonitorProgram = "/ibm/mq/myMonitor.sh"
    MonitorListener = 1
    MQInstallationPath = "/opt/customLocation"
```

```

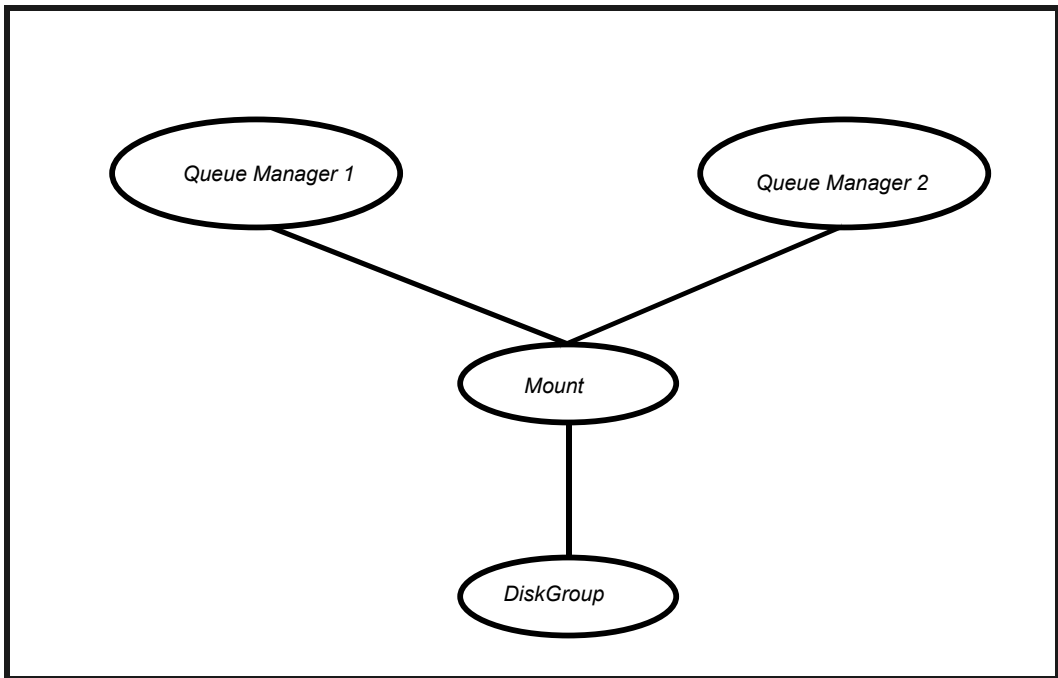
TransitionWaitLimit = 5
ServiceName = "mq.service"
SystemDEnvFile = "/var/mqm/mq_sysd.env"
)
Mount_OPT requires Volum_OPT
Mount_VAR requires Volume_VAR
Volum_OPT requires DG_OPT
Volume_VAR requires DG_VAR
WASMQ requires Mount_OPT
WASMQ requires Mount_VAR
// resource dependency tree
// group WASMQ_Solaris
// {
//   WebSphereMQ WASMQ
//   {
//     Mount Mount_OPT
//     {
//       Volume Volum_OPT
//       {
//         DiskGroup DG_OPT
//       }
//     }
//     Mount Mount_VAR
//     {
//       Volume Volume_VAR
//       {
//         DiskGroup DG_VAR
//       }
//     }
//   }
// }

```

Sample service group configurations

[Figure A-1](#) shows a sample service group that shows two WebSphere MQ queue manager resources.

This simple configuration also requires a Mount and a Disk Group resource.

Figure A-1 Sample Service group configuration

[Figure A-2](#) shows a sample service group that includes two WebSphere MQ queue manager resources with associated listeners.

In this example, each resource depends on a listener, which in turn depends on an IP and a Mount resource. This configuration applies to WebSphere MQ when listeners are used to provide remote services to application clients.

Figure A-2 Sample Service group configuration with listeners

