

Cluster Server Agent for WebSphere MQ FTE Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

8.0

VERITAS™

Last updated: 2023-01-05

Legal Notice

Copyright © 2023 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 FTE	7
	7
	About the Cluster Server agent for WebSphere MQ FTE	7
	Supported software	7
	How the agent makes WebSphere MQ FTE highly available	8
	About WebSphere MQ FTE	8
	Component architecture	8
	Agent queue manager	9
	Command queue manager	9
	Coordination queue manager	10
	Deployment topologies	10
	WebSphere MQ FTE agent functions	10
	Online	11
	Offline	11
	Monitor	11
	Clean	12
Chapter 2	Installing, upgrading, and removing the agent for WebSphere MQ FTE	13
	13
	Before you install the Cluster Server agent for WebSphere MQ FTE	13
	Prerequisites for enabling i18n support	14
	About the ACC library	14
	Installing the ACC library	15
	Installing the agent in a VCS environment	16
	Installing the agent in VCS One environment	17
	Installing the agent packages using the installer	18
	Installing the agent package using the CLI	19
	Adding the agent resource type definitions to the Policy Master	
	Server on UNIX	20
	Adding the agent resource type definitions to the Policy Master	
	Server on Windows	22
	Uninstalling the agent in a VCS environment	22
	Removing the agent in VCS One environment	23
	Removing the agent packages using the installer	24

	Removing the agent package using CLI	25
	Removing the agent type definition from the Policy Master system on UNIX	26
	Removing the agent type definition from the Policy Master system on Windows	27
	Removing the ACC library	27
	Upgrading the agent in a VCS environment	27
	Upgrading the agent in a VCS One environment	29
Chapter 3	Configuring the agent for WebSphere MQ FTE	31
	About configuring the Cluster Server agent for WebSphere MQ FTE	31
	About Configuring WebSphere MQ FTE for high availability	32
	Importing the agent types files in a VCS environment	32
	WebSphere MQ FTE agent attributes	34
	Executing a customized monitoring program	37
	Uniquely identifying the WebSphere MQ FTE agent	38
	Monitoring the WebSphere MQ FTE agent	38
Chapter 4	Configuring the service groups for WebSphere MQ FTE using the CLI	40
	About configuring service groups for WebSphere MQ FTE	40
	Before configuring the service groups for WebSphere MQ FTE	40
	Configuring service groups for WebSphere MQ Queue Managers	41
Chapter 5	Troubleshooting the agent for WebSphere MQ FTE	43
	Using the correct software and operating system versions	43
	Meeting prerequisites	43
	Configuring WebSphere MQ FTE agent resources	44
	Starting the WebSphere MQ FTE agent instance outside a cluster	44
	Reviewing error log files	45
	Reviewing cluster log files	45
	Using trace level logging	45

Appendix A	Sample Configurations	47
	About sample configurations for the agents for WebSphere MQ FTE	47
	Sample agent type definition for WebSphere MQ FTE	47
	VCS One	49
	Sample configuration in a VCS environment	49
	Sample configuration in a VCS One environment	51
	Sample service group configurations	51
Index		54

Introducing the agent for WebSphere MQ FTE

This chapter includes the following topics:

- [About the Cluster Server agent for WebSphere MQ FTE](#)
- [Supported software](#)
- [How the agent makes WebSphere MQ FTE highly available](#)
- [About WebSphere MQ FTE](#)
- [WebSphere MQ FTE agent functions](#)

About the Cluster Server agent for WebSphere MQ FTE

The Cluster Server (VCS) agent for WebSphere MQ FTE provides high availability for all WebSphere MQ File Transfer agents in a cluster. The agent can bring a specific WebSphere MQ File Transfer agent online and monitor the state of the FTE agent. The agent can also detect failures and stop the FTE agent in case of a failure.

Supported software

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

How the agent makes WebSphere MQ FTE highly available

The agent for WebSphereMQFTE agent continuously monitors the FTE agent processes to verify that they function properly.

The agent provides the following levels of application monitoring:

- Primary or Basic monitoring
This mode has Process check and Health check monitoring options. With the default Process check option, the agent verifies that the agent processes are present in the process table. Process check cannot detect whether processes are in the hung or stopped states.
- Secondary or Detail monitoring
In this mode, the agent runs a utility to verify the status of the agent. The agent detects application failure if the monitoring routine reports an improper function of the agent processes. When this application failure occurs, the agent service group fails over to another node in the cluster.
Thus, the agent ensures high availability for agents.

About WebSphere MQ FTE

IBM® WebSphere® MQ File Transfer Edition, called FTE, integrates with an existing WebSphere MQ network and adds managed file transfer solution. From an architectural perspective, FTE is constructed like any other WebSphere MQ application. It connects to queue managers to put messages into or get messages from queues. It is secured through authentication and access control, like any other MQ application. However, unlike most messaging applications, FTE crosses a security domain boundary from the messaging realm into the operating system's file and directory service.

Component architecture

There are different versions of FTE, depending on whether it will connect to the queue manager in bindings or client mode. In addition to the functional components, each version also includes installation media containing remote tools and documentation packages. During configuration, the administrator is asked to supply the details of three queue managers known as the agent queue manager, command queue manager, and coordination queue manager. FTE is a single long-running daemon process called an agent, with tools to let users submit commands to the agent and inquire on its status. Beyond that, everything else is just configuration details.

In a typical installation of FTE there is a sender FTE agent and a receiver FTE agent, each of which has its own command queue. File transfers are managed by users or administrators by placing messages on the sending FTE agent's command queue, either directly or with the supplied tooling. The FTE agent processes respond to command messages, move files through WebSphere MQ, and then publish activity logs onto the SYSTEM.FTE topic. Users, administrators, and applications may then subscribe to the status updates on the SYSTEM.FTE topic.

Each FTE component requires access to a queue manager. In the simplest case, the users and FTE agents can all connect to the same queue manager. At the other extreme, it is possible to distribute the FTE agents and users across the network and let WebSphere MQ take care of routing the commands and data traffic between them. When discussing topologies, it is necessary to distinguish the different nodes based on their role. As noted above, they are referred to as agent queue manager, command queue manager, and coordination queue manager.

Agent queue manager

Each FTE agent is associated with a single V6.0 or later queue manager. The FTE agent and its queue manager may be hosted on the same server or the FTE agent may use a WebSphere MQ client channel to communicate with a remote queue manager. When the FTE agent is deployed, the administrator will create a set of queues for that agent to use. As the queue names include the FTE agent's name, it is possible for one queue manager to host multiple agents.

Command queue manager

The command-line and Eclipse plug-in tooling lets users and instrumentation issue commands to the various FTE agents in the network. In a small network with two FTE agents connected to a single queue manager, the tooling can be configured to connect directly to that same queue manager. But in a distributed network with many FTE agent queue managers, it is not practical or necessary to have the tooling connect directly to each FTE agent queue manager to issue commands.

Instead, the tooling is configured to connect to a single queue manager that has connectivity to all of the FTE agents, relying on WebSphere MQ to route the command messages accordingly. Whichever queue manager the tooling connects to is referred to as the command queue manager. There is nothing special about this queue manager other than that it can resolve a route to each agent queue manager. There is no FTE code required to run here and the only queues required are the temporary dynamic queues created by the tooling at run time. Any V6.0 or later queue manager will work. There is also no requirement that all users connect to the same command queue manager and in fact the expected deployment would group sets of users by role onto different command queue managers.

Coordination queue manager

The coordination queue manager is the one thing all of the components in the FTE network have in common. Any significant event occurring at the FTE agent results in a message published by the agent on the SYSTEM.FTE topic hosted on the coordination queue manager. Any user or application that needs to know what is happening in the FTE network must subscribe to these publications. Both the command-line and Eclipse plug-in tooling require a connection to the coordination queue manager to consume publications. FTE agents have no such requirement, because the publications can be sent from the agent queue manager across the WebSphere MQ network to the coordination queue manager. There is no actual FTE code required to run on the coordination queue manager. All of the work is accomplished using native WebSphere MQ functionality. FTE requires that this queue manager must be a WebSphere MQ V7.0 or later queue manager for the publish/subscribe features.

Deployment topologies

The simplest example of an FTE network is a single queue manager that serves as agent queue manager, coordination queue manager, and command queue manager all bundled into one. While this will work, it would not be very useful if it just moved a file from one directory on a server to another directory on the same server. To get the FTE network out of the lab and useable in the real world, the first thing to do is to distribute the FTE agent processes onto the servers where files will be consumed or delivered. The queue manager in this case serves simultaneously in all three roles of Coordination, Command, and Agent queue manager.

A high volume FTE network is alternative approach to deploy FTE as mentioned in the above paragraph. In this topology, command and status messages still flow through a single queue manager serving as both coordination and command queue manager, but data flows directly between the two agent queue managers over WebSphere MQ channels such as a Sender/Receiver pair.

For more information about WebSphere MQ FTE see the IBM WebSphere MQ FTE information centre.

WebSphere MQ FTE agent functions

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

Online

The online function performs the following tasks:

- Verifies that WebSphere MQ FTE is not already online.
- Uses an IBM provided start script to start the WebSphere MQ FTE agent using the name of the FTE agent and coordination 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 FTE agent is up and running successfully. The online operation uses the wait period that the OnlineTimeout attribute specifies, to enable the FTE agent to initialize fully before allowing the monitor function to probe the resource

Offline

The offline function performs the following tasks:

- Verifies that the WebSphere MQ FTE agent is not already offline.
- Uses an IBM provided stop script to stop the WebSphere MQ FTE agent using the name of the Agent Name and Coordination Queue Manager.
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 FTE agent is given enough time to go offline successfully. The offline operation uses a wait period that the OfflineTimeout attribute specifies, to allow the WebSphere MQ FTE agent to complete the offline sequence before allowing further probing of the resource.

Monitor

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

- The first level check searches for all system processes that must be running for a WebSphere MQ FTE agent. If the first level check does not find these processes running on the node, the check exits immediately, and reports the FTE agent as offline.
- If the SecondLevelMonitor attribute is set to greater than 0, the monitor function performs a second level check to determine the status of the WebSphere MQ

FTE agent. The second level check performs a dummy transfer to see if the FTE agent is up and running. This check ensures that the processes are truly available for MQ FTE agent processing.

- 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 37.

Clean

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

The function performs the following tasks:

- Attempts to gracefully shut down the WebSphere MQ FTE agent.
- If a graceful shutdown fails, the clean function looks for all the processes running for the WebSphere MQ FTE agent , and cleans the processes by killing them.

Installing, upgrading, and removing the agent for WebSphere MQ FTE

This chapter includes the following topics:

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

Before you install the Cluster Server agent for WebSphere MQ FTE

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

Before you install the agent for WebSphere MQ FTE, 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.
- 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 14.

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 15.
- 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 `cd1/aix/vcs/application/acc_library/version_library/pkg`s

HP-UX `cd1/hpux/generic/vcs/application/acc_library/version_library/pkg`s

Solaris `cd1/solaris/dist_arch/vcs/application/acc_library/version_library/pkg`s

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
HP-UX        # swinstall -s `pwd` VRTSacclib
Solaris      # pkgadd -d VRTSacclib.pkg
```

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 agent in a VCS environment

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

Note: The agent package VRTSmq6 includes the Cluster Server agents for WebSphere MQ and WebSphere MQ FTE. The following procedure lets you install both these agents.

If you are installing the agent for WebSphere MQ FTE and if you already have an existing older version of the agent for WebSphere MQ, you must first remove the agent for WebSphere MQ from your system. For information on removing and upgrading the agent for WebSphere MQ, refer to the *Cluster Server Agent for WebSphere MQ Installation and Configuration Guide*.

To install the agent 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.

Installing the agent packages	See “Installing the agent packages using the installer” on page 18.
Adding the agent resource type definitions	See “Adding the agent resource type definitions to the Policy Master Server on UNIX” on page 20.
	See “Adding the agent resource type definitions to the Policy Master Server on Windows” on page 22.

Note: The `installagpack` program supports only the `-addtypes`, `-rmtypes`, `-responsefile`, and `-rsh` options. Veritas recommends that you do not use any of the other options from the `installagpack` command help output.

Installing the agent packages using the installer

You can install the agent packages on one or more client systems of a specific platform type.

Note: To install the VCS One client for managing VMware ESX Servers, download the tar ball for Red Hat Enterprise Linux 4 (RHEL 4) x86 (32-bit) or RHEL 5 x86_64

Perform the following steps to install the agent packages using the installer

- 1 On the Policy Master system, download the complete Agent Pack tarball or the individual agent tarball from the Veritas Services and Operations Readiness Tools (SORT) site: <https://sort.veritas.com/agents>.
- 2 Uncompress the file to a temporary location, say /tmp.
- 3 If you downloaded the complete Agent Pack tarball, navigate to the following directory containing the installer for the VCS One agents, for the platform running in your environment:

AIX `cd1/aix/vcsone/vcsone_version`

HP-UX `cd1/hpux/hpuxos_version/vcsone/vcsone_version`

Where `os_version` is the HP-UX version.

Linux `cd1/linux/dist_arch/vcsone/vcsone_version`

Where `dist` is the Linux distribution and `arch` is the architecture.

Solaris `cd1/solaris/dist_arch/vcsone/vcsone_version`

Where, `dist_arch` is 'sol_sparc'.

- 4 Enter the following command to start the agent pack installation:

```
# ./installagpack [-rsh]
```

You can use the `-rsh` option if `rsh` and `rcp` are used for communication between systems instead of the default `ssh` and `scp`. This option requires that systems be preconfigured such that the `rsh` commands between systems execute without prompting for passwords or confirmations.

- 5 Enter the name of the client systems where you want to install the agents.
- 6 Choose whether to install all the agents or any specific agent. Follow the installer prompt to specify your option.
- 7 Review the output as the installation program installs the agent packages.

You can view installation logs in the `/var/VRTS/install/logs` directory.

Installing the agent package using the CLI

You can install the desired agent package using the CLI, on one or more client systems of a specific platform type.

Perform the following steps to install the agent packages using CLI

- 1 On the Policy Master system, download the complete Agent Pack tarball or the individual agent tarball from the Veritas Services and Operations Readiness Tools (SORT) site: <https://sort.veritas.com/agents>.
- 2 Uncompress the file to a temporary location, say `/tmp`.

- 3 If you downloaded the complete Agent Pack tarball, navigate to the following directory containing the installer for the VCS One agents, for the platform running in your environment:

AIX # `cd1/aix/vcsone/vcsone_version/pkggs`

HP-UX # `cd1/hpux/hpuxos_version/vcsone/vcsone_version/depot`

Linux # `cd1/linux/dist_arch/vcsone/vcsone_version/rpms`

Where, *dist* is the Linux distribution and *arch* is the architecture

Solaris # `cd1/solaris/dist_arch/vcsone/vcsone_version/pkggs`

Where *dist_arch* is 'sol_sparc'.

- 4 Type the following command on each client system to install the agent. Answer the prompt accordingly:

AIX # `installp -ac -d . VRTSmq6.rte`

HP-UX # `swinstall -s `pwd` VRTSmq6`

Linux # `rpm -ivh VRTSmq6_rpm_filename`

Solaris For Solaris 10:

`pkgadd -d . VRTSmq6`

For Solaris 11:

Adding the agent resource type definitions to the Policy Master Server on UNIX

You must add the agent resource type definitions to the Policy Master database configuration. You can perform this task from any client system in the VCS One cluster.

Note: You must add the agent resource type definitions only one time per platform type.

To add the agent resource types to the policy master database configuration

- 1 Set up RSH or SSH communications between the client system and the policy master system.

For information on configuring SSH for remote communication, refer to the *Veritas Cluster Server One Installation Guide*.

- 2 Make sure that the PM daemon is running.

```
# /opt/VRTSvcstone/bin/haclus -display
```

The output should show ClusterState is RUNNING.

- 3 Access the temporary location where you downloaded the tar ball and depending on the platform type, navigate to the directory containing the agent installer:

AIX `cd1/aix/vcsone/vcsone_version`

HP-UX `cd1/hpux/hpuxos_version/vcsone/vcsone_version`

Where *os_version* is the HP-UX version.

Linux `cd1/linux/dist_arch/vcsone/vcsone_version`

Where *dist* is the Linux distribution and *arch* is the architecture.

Solaris `cd1/solaris/dist_arch/vcsone/vcsone_version`

Where *dist_arch* is the sol_sparc.

- 4 Enter the command to start the agent pack installer for adding resource types to the Policy Master configuration database. Use the `-addtypes` option:

```
# ./installagpack -addtypes
```

- 5 When the installer prompts, enter the virtual IP address of the Policy Master.
- 6 Review the output as the installer verifies communication with the Policy Master system.
- 7 Choose whether to add the type definitions for all the agents or for specific agents. Follow the installer prompts to add the type definitions.
- 8 Review the output as the installer adds the agent types to the PM database configuration and copies the appropriate types.xml files to the PM system.

You can view installation logs in the `/var/VRTS/install/logs` directory.

Adding the agent resource type definitions to the Policy Master Server on Windows

After you have installed the agent package, you must add the agent resource type definitions to the Policy Master database configuration. You must perform this task from the Policy Master Server.

Note: You must add the agent resource type definitions only one time per platform type.

To add the agent resource types to the Policy Master Server on Windows

Perform the following steps from the Policy Master Server command prompt.

- 1 Create a temporary directory on the Policy Master Server, to add the type definitions.

```
C:\> mkdir addtypes_tmp
```

- 2 Change your working directory to the temporary directory created in step 1.

```
C:\> chdir addtypes_tmp
```

- 3 Copy the agent's type xml file in the temporary directory.

- 4 Convert this type xml file into the type cmd file.

```
C:\addtypes_tmp> haconf -xmlto cmd type_xml_filename.xml
```

- 5 Rename the *type_xml_filename.xml.cmd* file to *type_xml_filename.bat*

- 6 Run the batch file.

```
C:\addtypes_tmp> type_xml_filename.bat >log.txt 2>&1
```

- 7 Review the log.txt file for any errors.

- 8 Verify that the type has been successfully added to the Policy Master Server.

```
C:\addtypes_tmp> hatype -list -platform platform_name
```

Uninstalling the agent in a VCS environment

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

Warning: The agent package VRTSmq6 includes the agents for WebSphere MQ and WebSphere MQ FTE. So, the following procedure to remove the agent for WebSphere MQ FTE removes the agent for WebSphere MQ 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 running the following command from any node in the cluster:

```
# haconf -makerw
```

- 3 Remove all WebSphere MQ FTE agent resources from the cluster. Run the following command to verify that all resources have been removed:

```
# hares -list Type=WebSphereMQFTE
```

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

```
# hatype -delete WebSphereMQFTE
```

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 running 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 FTE from each node in the cluster.

Run the following command to uninstall the agent:

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

```
HP-UX       # swremove VRTSmq6
```

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

```
Solaris    # pkgrm VRTSmq6
```

Removing the agent in VCS One environment

Removing the agent package involves removing the agent files from each client system where it was installed.

You can remove the packages using the agent pack installer or the command line.

See [“Removing the agent packages using the installer”](#) on page 24.

See [“Removing the agent package using CLI”](#) on page 25.

After removing the agent packages you can remove the agent type definition from the Policy Master system.

See [“Removing the agent type definition from the Policy Master system on UNIX”](#) on page 26.

See [“Removing the agent type definition from the Policy Master system on Windows”](#) on page 27.

Removing the agent packages using the installer

You can remove all the agent packages or the desired agent package using the `uninstallagpack` program.

Note: The `uninstallagpack` program supports only the `-responsefile` and `-rsh` options. Veritas recommends that you do not use any of the other options from the `uninstallagpack` command help output.

To remove the agent packages from the client systems

- 1 Freeze the service groups that hosts the application, on the system from which you want to remove the agent package.

```
# hagr -freeze <groupname>
```

- 2 Stop the agent on all client systems before you remove the agent package from the system.

```
# haagent -stop -notransition <AgentName> -sys <system_name>
```

- 3 Ensure that the agent operations are stopped on all the cluster systems.

```
# haagent -display <AgentName>
```

- 4 Access the temporary location where you downloaded the Agent Pack and navigate to the directory containing the package for the platform running in your environment:

AIX `cd1/aix/vcsone/vcsone_version`

HP-UX `cd1/hpux/hpuxos_version/vcsone/vcsone_version`

Where *os_version* is the HP-UX version.

Linux `cd1/linux/dist_arch/vcsone/vcsone_version`

Where *dist* is the Linux distribution and *arch* is the architecture.

Solaris `cd1/solaris/dist_arch/vcsone/vcsone_version`

Where *dist_arch* is the *sol_sparc*.

- 5 Start the `uninstallagpack` program.

```
# ./uninstallagpack [-rsh]
```

- 6 Enter the name of the client systems on which you want to uninstall the agent pack. The names must be separated by spaces.
- 7 Choose whether to remove all the agent packages or a specific agent package. Follow the installer prompt to remove the agent package.
- 8 Review the output as the program verifies the agent pack that you installed and removes the agent packages.

You can view logs in the `/var/VRTS/install/logs` directory.

Removing the agent package using CLI

You can remove a desired agent package using the CLI.

Note: You must remove this agent package from each client system in the cluster.

To remove the agent for WebSphere MQ FTE from a client system

Run the following command on each client system to remove the agent. Answer prompts accordingly:

```
AIX          # installp -u VRTSmq6
```

```
HP-UX       # swremove VRTSmq6
```

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

```
Solaris     For Solaris 10:
```

```
            # pkgrm VRTSmq6
```

```
            For Solaris 11:
```

```
            # pkg uninstall VRTSmq6
```

Removing the agent type definition from the Policy Master system on UNIX

After you remove the agent packages, you can remove the agent type definitions for agents you removed, from the Policy Master system.

To remove the agent type definition from the Policy Master system on UNIX

- 1 Navigate to the following directory on the client system.

```
# cd /opt/VRTS/install
```

- 2 Run the following command to remove the agent type definition from the Policy Master system:

```
# ./installagpack -rmtypes
```

- 3 When the installer prompts, enter the virtual IP address of the Policy Master.
- 4 Choose whether to remove the type definitions for all the agents or for specific agents. Follow the installer prompts to remove the type definitions.

You can view logs in the `/var/VRTS/install/logs` directory.

Removing the agent type definition from the Policy Master system on Windows

After you remove the agent packages, you can remove the agent type definitions for agents you removed, from the Policy Master system.

To remove the agent type definition from the Policy Master system on Windows

Run the following command from the Policy Master Server command prompt.

```
C:\> hatype -delete agentname_i.e._typename -platform platformname
```

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
HP-UX       # swremove VRTSacclib
Solaris     # pkgrm VRTSacclib
```

Upgrading the agent in a VCS environment

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

Note: The agent package VRTSmq6 includes the Veritas agents for WebSphere MQ and WebSphere MQ FTE. Hence, both the agents will be upgraded as the result of upgrading the package. So, perform the following steps for the agent for WebSphere MQ as well.

To upgrade the agent in a VCS environment

- 1 Persistently freeze the service groups that host the application.

```
# hagrpfreeze -freeze groupName -persistent
```

- 2 Stop the cluster services forcibly.

```
# hastop -all -force
```

3 Ensure that the agent operations are stopped on all the nodes.

```
# ps -ef | grep WebSphereMQFTE
```

4 Uninstall the agent package from all the nodes. Use the platform's native software management program to remove the agent for WebSphere MQ FTE from each node in the cluster.

Run the following command to uninstall the agent:

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

```
HP-UX # swremove VRTSmq6
```

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

Solaris For Solaris 10:

```
# pkgrm VRTSmq6
```

For Solaris 11:

```
# pkg uninstall VRTSmq6
```

5 Install the new agent on all the nodes.

See [“Installing the agent in a VCS environment”](#) on page 16.

6 Copy the new `WebSphereMQFTETypes.cf` file from the agent's conf directory, to the VCS conf directory `/etc/VRTSvcS/conf/config`.

VCS 4.x	■ AIX	/etc/VRTSvcS/conf/sample_WebSphereMQFTE/
	■ HP-UX	WebSphereMQFTETypes.cf
	■ Linux	
	■ Solaris	

VCS 5.x or later	■ AIX	/etc/VRTSagents/ha/conf/WebSphereMQFTE/
	■ HP-UX	WebSphereMQFTETypes.cf
	■ Linux	

VCS 5.0	■ Solaris SPARC	/etc/VRTSagents/ha/conf/WebSphereMQFTE/ WebSphereMQFTETypes50.cf
---------	--------------------	---

VCS 5.1 or later	■ Solaris SPARC	/etc/VRTSagents/ha/conf/WebSphereMQFTE/ WebSphereMQFTETypes51.cf
------------------	--------------------	---

- 7 Check for the changes in the resource values required, if any, due to the new agent types file.

Note: To note the list of changed attributes, compare the new type definition file with the old type definition file.

- 8 Start the cluster services.

```
# hastart
```

- 9 Start the agent on all nodes, if not started.

```
# haagent -start WebSphereMQFTE -sys SystemName
```

- 10 Unfreeze the service groups once all the resources come to an online steady state.

```
# hagrps -unfreeze GroupName -persistent
```

Upgrading the agent in a VCS One environment

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

Note: The agent package VRTSmq6 includes the agents for WebSphere MQ and WebSphere MQ FTE. Hence, both the agents will be upgraded as the result of upgrading the package. So, perform the following steps for the agent for WebSphere MQ as well.

To upgrade the agent with minimal disruption, in a VCS One environment

- 1 Freeze service groups that hosts the application.

```
# hagrps -freeze -propagate GroupName
```

- 2 Stop the clients forcibly. Execute the following command from the Policy Master.

```
# hastop -client -sys SystemName -force
```

- 3 Ensure that the agent operations are stopped on all the nodes.

```
# ps -ef | grep WebSphereMQFTE
```

- 4 Uninstall the agent package from all the nodes. Type the following command on each client system to remove the agent. Answer prompts accordingly:

```
AIX          # installp -u VRTSmq6
```

```
HP-UX       # swremove VRTSmq6
```

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

```
Solaris    For Solaris 10:
```

```
           # pkgrm VRTSmq6
```

```
           For Solaris 11:
```

```
           # pkg uninstall VRTSmq6
```

- 5 Install the new agent on all the nodes in the cluster.
See [“Installing the agent in VCS One environment”](#) on page 17.
- 6 Add the agent types, using the `installagpack` program.
See [“Adding the agent resource type definitions to the Policy Master Server on UNIX”](#) on page 20.
- 7 Check for the changes in the resource values required, if any, due to the new agent types file.
- 8 Start the clients.

```
# hastart -client
```
- 9 Start the agent on all nodes, if not started.

```
# haagent -start WebSphereMQFTE -sys SystemName
```
- 10 Unfreeze the service groups.

```
# hagrps -unfreeze -propagate GroupName
```

Configuring the agent for WebSphere MQ FTE

This chapter includes the following topics:

- [About configuring the Cluster Server agent for WebSphere MQ FTE](#)
- [About Configuring WebSphere MQ FTE for high availability](#)
- [Importing the agent types files in a VCS environment](#)
- [WebSphere MQ FTE agent attributes](#)
- [Executing a customized monitoring program](#)
- [Uniquely identifying the WebSphere MQ FTE agent](#)
- [Monitoring the WebSphere MQ FTE agent](#)

About configuring the Cluster Server agent for WebSphere MQ FTE

After installing the Cluster Server agent for WebSphere MQ FTE, 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 FTE agent resources.

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

See [“About sample configurations for the agents for WebSphere MQ FTE”](#) on page 47.

About Configuring WebSphere MQ FTE for high availability

You can configure the WebSphere MQ FTE agent to be highly available by installing it on all the nodes in the cluster and moving its data directory (`/var/IBM/WMQFTE`) on the shared storage.

The WebSphere MQ FTE agent depends on three WebSphere MQ queue managers for its functioning. In some cases, a single queue manager can be used for all the three purpose while in others the three different queue manager can be used and all of them can be on different hosts. So it is recommended to configure the dependency between the service group containing the WebSphere MQ FTE agent resource and the service group containing the WebSphere MQ resource for corresponding queue manager.

The Dependency type between the WebSphere MQ FTE agent service group and service group containing WebSphere MQ queue manager will be “Online Global Soft”. All the queue manager should be running for WebSphere MQ FTE agent to work properly. If the FTE agent is running and any of the queue manager goes down then it does not affect the FTE agent until any transaction is requested by the application which involve that particular queue manager. In that case the FTE agent will notify that it is not able to connect to that particular queue manager. When the particular queue manager again comes up running and application request for the same transaction, the FTE agent will connect to the queue manager and perform the appropriate action to fulfill the request. Since for each transaction a new connection will be established by the FTE agent to the respective queue manager, there is no need to restart the FTE agent in case of any failure. The queue managers can be running on any machine but the hostname should be known to the FTE agent since it will connect to the hostname though the listener on which the queue manager listen to.

Importing the agent types files in a VCS environment

To use the agent for WebSphere MQ FTE, you must import the agent types file into the cluster. You can import the agent types file using the VCS graphical user interface or using the command line interface.

To import the agent types file using the VCS Java GUI

- 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	<ul style="list-style-type: none"> ■ AIX ■ HP-UX ■ Linux ■ Solaris 	<pre>/etc/VRTSvcs/conf/sample_WebSphereMQFTE/ WebSphereMQFTETypes.cf</pre>
VCS 5.x or later	<ul style="list-style-type: none"> ■ AIX ■ HP-UX ■ Linux 	<pre>/etc/VRTSagents/ha/conf/WebSphereMQFTE/ WebSphereMQFTETypes.cf</pre>
VCS 5.0	Solaris SPARC	<pre>/etc/VRTSagents/ha/conf/WebSphereMQFTE/ WebSphereMQFTETypes50.cf</pre>
VCS 5.1 or later	Solaris SPARC	<pre>/etc/VRTSagents/ha/conf/WebSphereMQFTE/ WebSphereMQFTETypes51.cf</pre>

4 Click **Import**.

5 Save the VCS configuration.

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

You can now create WebSphere MQ FTE agent resources. For additional information about using the VCS GUI, refer to the *Cluster Server Administrator's Guide*.

To import the agent types file using the CLI

1 Log on to any one of the systems in the cluster as the superuser.

2 Create a temporary directory.

```
# mkdir ./temp  
  
# cd ./temp
```

3 Copy the sample file `Types.cf`.

VCS 4.x	AIX	/etc/VRTSvcs/conf/sample_WebSphereMQFTE/
	HP-UX	WebSphereMQFTETypes.cf
	Solaris	
VCS 5.x or later	AIX	/etc/VRTSagents/ha/conf/WebSphereMQFTE/
	HP-UX	WebSphereMQFTETypes.cf
VCS 5.0	Solaris SPARC	/etc/VRTSagents/ha/conf/WebSphereMQFTE/
		WebSphereMQFTETypes50.cf
VCS 5.1 or later	Solaris SPARC	/etc/VRTSagents/ha/conf/WebSphereMQFTE/
		WebSphereMQFTETypes51.cf

4 Create a dummy `main.cf` file.

```
# echo 'include "WebSphereMQFTETypes.cf"' > main.cf
```

5 Create the WebSphere MQ FTE resource type as follows:

```
# hacf -verify .  
# haconf -makerw  
# sh main.cmd  
# haconf -dump
```

The WebSphere MQ FTE agent type is now imported to the VCS engine. You can now create WebSphere MQ FTE agent resources. For additional information about using the VCS CLI, refer to the *Cluster Server Administrator's Guide*.

WebSphere MQ FTE agent attributes

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

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

Table 3-1 Required attributes

Required attributes	Description
FTEAgentName	<p>Name of the WebSphere MQ FTE agent.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: AGENT1</p>
DataDir	<p>Absolute path to the data directory of WebSphere MQ FTE provided at the time of installing WebSphere MQ FTE.</p> <p>Type and dimension: string-scalar</p> <p>Default: /var/IBM/WMQFTE</p> <p>Example: /var/IBM/WMQFTE</p>
HomeDir	<p>Absolute Path to the home installation directory of the WebSphere MQ FTE provided at the time of installation of WebSphere MQ FTE.</p> <p>Type and dimension: string-scalar</p> <p>Default: /opt/IBM/WMQFTE</p> <p>Example: /opt/IBM/WMQFTE</p>
CoordinationQM	<p>Coordination queue manager name for the WebSphere MQ FTE agent.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: QM1</p>
AgentQM	<p>Agent queue manager name for the WebSphere MQ FTE agent.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: QM2</p>

Table 3-1 Required attributes (*continued*)

Required attributes	Description
FTEUser	<p>User name that the agent for WebSphere MQ FTE uses to execute the programs for managing a WebSphere MQ FTE agent.</p> <p>The user name must also be synchronized across the systems in the cluster. In other words, the user name must resolve to the same UID and have the same default shell on each system in the cluster.</p> <p>The agent functions use the <code>getpwnam(3c)</code> function call to obtain UNIX user attributes. As a result, the user can be defined locally or can be defined in a common repository (that is, NIS, NIS+, or LDAP). In the latter case, the agent will fail if the access to this repository fails.</p> <p>The supported shell environments are: ksh, sh, and csh.</p> <p>Type and dimension: string-scalar</p> <p>Default: mqm</p> <p>Example: mqusr1</p>
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>

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

Table 3-2 Optional attributes

Optional attribute	Description
EnvFile	<p>Full path to the file that the WebSphere MQ FTE sources to set the environment variables. Veritas recommends storing this file on a shared disk.</p> <p>The supported shell environments are: ksh, sh, and csh.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /opt/IBM/WMQFTE/envfile</p>
SecondLevelMonitor	<p>Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the WebSphere MQ FTE agent. 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>Type and dimension: integer-scalar</p> <p>Default: 0</p> <p>Example: 5</p>
MonitorProgram	<p>Absolute path name of an external, user-supplied monitor executable.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /var/ibm/wmqfte/myMonitor.sh</p>

Executing a customized monitoring program

The monitor function can execute a customized monitoring utility to perform an additional WebSphere MQ FTE agent 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 FTE agent agent instance is online.

- The `LevelTwoMonitorFreq` attribute is either set to 0 or 1, and the second-level check indicates that the WebSphere MQ FTE agent instance is online.
- The `LevelTwoMonitorFreq` attribute is set to greater than 1, but the second-level check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

110 or 0	WebSphere MQ FTE agent instance is online
100 or 1	WebSphere MQ FTE agent instance is offline
99	WebSphere MQ FTE agent instance is unknown
Any other value	WebSphere MQ FTE agent 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 the online node.

Uniquely identifying the WebSphere MQ FTE agent

For multiple WebSphere MQ FTE agents running concurrently on a single node, the agent for FTE must be able to uniquely identify each FTE agent on that system. Each FTE agent has a unique agent name under same coordination queue manager. The agent for FTE will use the `CoordinationQM` and `FTEAgentName` attribute value to identify the FTE agent uniquely. Differentiating the WebSphere MQ FTE agent is important to identify each FTE agent uniquely. In the absence of unique names for each FTE agent, the agent for FTE may kill the processes for more than one FTE agent during a clean operation, while attempting to kill the processes of a non-responsive or failed instance.

Monitoring the WebSphere MQ FTE agent

- In first level monitoring, the agent for WebSphere MQ FTE checks for the existence of the Java process that represents the WebSphere MQ FTE agent instance. The agent for WebSphere MQ FTE checks for the process existence by scanning the system process table and searching for strings in the process command line that uniquely identify the Java process associated with the WebSphere MQ FTE agent. These search strings include the values specified in resource attributes `DataDir`, `HomeDir`, `CoordinationQM`, and `FTEAgentName`.
- The agent for WebSphere MQ FTE also performs a `Pid` file-based monitoring in the first level monitoring. When each FTE agent gets started, it creates a `Pid` file by the name `agent.lck` under `FTEAgentName`. The agent for WebSphere

MQ FTE checks for the existence of this Pid file in the FTE agent directory. If it is present, the agent for WebSphere MQ FTE extracts the process id from that file and then perform the process level check in that process command line.

If the file does not exist in the FTE agent directory, the agent for WebSphere MQ FTE will continue with the process check from the system process table.

- In second level monitoring, the agent for WebSphere MQ FTE performs a dummy transfer of a file. The agent for WebSphere MQ FTE transfers a small dummy file by executing the `fteCreateTransfer` command in the `/tmp` directory. Thereafter, the agent for WebSphere MQ FTE repeatedly checks for the existence of destination file by delaying the monitor function. If the file is transferred successfully, it implies that the FTE agent is working fine.

Configuring the service groups for WebSphere MQ FTE using the CLI

This chapter includes the following topics:

- [About configuring service groups for WebSphere MQ FTE](#)
- [Before configuring the service groups for WebSphere MQ FTE](#)
- [Configuring service groups for WebSphere MQ Queue Managers](#)

About configuring service groups for WebSphere MQ FTE

Configuring the WebSphereMQFTE agent service group involves creating the service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group. You can configure the service groups using the Cluster Manager (Java console).

Before configuring the service groups for WebSphere MQ FTE

Before you configure the WebSphere MQ FTE agent 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 *Installation Guide*.

- Verify that the Cluster Server agent for WebSphere MQ FTE is installed on all nodes in the cluster.
- Verify that WebSphere MQ FTE is installed on all the systems in the cluster on the same path that will host a WebSphereMQFTE service group.
- Verify that the WebSphere MQ FTE agents are created using the tools available with the WebSphere MQ FTE installation, such as fteCreateAgent. Make sure that coordination, command and all queue managers are running, respective channels and listeners are properly configured and running.
- Verify that the WebSphere MQ FTE agent to be clustered is working fine.
- Verify that the agent for WebSphere MQ FTE is installed on all nodes in the cluster.
- Verify that the type definition for the agent WebSphere MQ FTE is imported into the VCS engine.

Configuring service groups for WebSphere MQ Queue Managers

A cluster must have spare capacity to handle WebSphere MQ FTE agent failover scenarios. A cluster can provide application failover by encapsulating the resources required for an application into a service group. A service group is a virtualized application that can move among cluster nodes. A service group can contain a set of dependent resources, such as disk groups, disk volumes, file systems, IP addresses, NIC cards, and dependent application processes.

A cluster can start, stop, monitor, and switch service groups within the cluster, depending upon server or resource faults. An administrator can proactively move a service group between cluster nodes to perform preventative maintenance or apply patches. The service group includes logic about the dependencies between application components.

Perform the following steps to add a service group for WebSphere MQ FTE agent:

To add a service group for WebSphere MQ FTE agent

- 1 Create a separate service groups for each WebSphere MQ FTE agent. In each service group create the respective WebSphere MQ FTE resource.
- 2 Provide the specific attribute values for all the resources. Refer to the attribute table.
- 3 Create a separate file system for WebSphere MQ FTE on shared disk.

- 4 Add the file system to respective agent service group using Mount, DiskGroup and Volume resources.
- 5 Copy the contents of data directory of the WebSphere MQ FTE to this shared file system. Ensure that you copy the sub-directories also.
- 6 Delete the contents of data directory and then create a link between data directory and the directory on the shared disk where you copied the contents of the data directory.
- 7 Create the resource dependency between the resources by linking them.
- 8 Create the service group for WebSphere MQ Queue Manager on which the WebSphereMQFTE agent will depend for its functioning. For more information refer to the *Cluster Server Agent for WebSphere MQ Installation and Configuration Guide*.
- 9 Add the service group dependency between the service group created in step 1 and service group created in step 8 by performing following steps
 - Go to the Service Group View.
 - Link the service group containing WebSphere MQ FTE and WebSphere queue manager as mentioned in the preceding section.
 - Choose Relationship as “Online Global” and Dependency Type as “Soft.”

Troubleshooting the agent for WebSphere MQ FTE

This chapter includes the following topics:

- [Using the correct software and operating system versions](#)
- [Meeting prerequisites](#)
- [Configuring WebSphere MQ FTE agent resources](#)
- [Starting the WebSphere MQ FTE agent instance outside a cluster](#)
- [Reviewing error log files](#)

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 FTE 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 FTE, ensure that the following prerequisites are met.

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

See “[Before you install the Cluster Server agent for WebSphere MQ FTE](#)” on page 13.

Configuring WebSphere MQ FTE agent resources

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

Starting the WebSphere MQ FTE agent 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 FTE agent instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the WebSphere MQ FTE agent 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 FTE instance outside the cluster framework, is described as follows.

To restart the WebSphereMQ FTE agent instance outside a cluster

- 1 Log in to the WebSphere MQ FTE agent as an MQUser.

```
# su - MQUser
```

- 2 Start the WebSphere MQ FTE agent.

```
# fteStartAgent -p (coordination_QM) agent_name
```

- 3 Do a simple File Transfer using WebSphere MQ FTE.

```
# fteCreateTransfer -sa AG1 -sm MQ -da AGENT1 -dm QM2 -df
/tmp/mq.copy -pr 0 -p QM1 -cs MD5
/var/IBM/WMQFTE/config/MQ1/agents/AG1/agent.properties
```

If the WebSphere MQ FTE Agent works properly outside the cluster framework, you can then attempt to implement the FTE agent within the cluster framework.

Reviewing error log files

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

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

Reviewing cluster log files

In case of problems while using the agent for WebSphere MQ FTE, 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`.

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

The VCS One client log file is at `/var/VRTSvcsone/log/vcsoneclientd_A.log`.

The VCS WebSphere MQ FTE resource log file is `/var/VRTSvcs/log/WebSphereMQFTE_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 FTE agent 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.

Note: For ACC library versions 5.1.1.0 and later, the `TRACE` level logs for any ACCLib based agent are generated locally at the location

`/var/VRTSvcs/log/WebSphereMQFTE_A.log`.

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 Test the identified resource. The function reproduces the problem that you are attempting to diagnose.

- 5 Set the ResLogLevel attribute back to INFO for the identified resource:

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

- 6 Save the configuration changes.

```
# haconf -dump
```

- 7 Review the contents of the log file.

You can also contact Veritas support for more help.

To enable debug logs for all resources of type WebSphereMQFTE

Enable the debug log.

```
# hatype -modify WebSphereMQFTE 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 WebSphereMQFTE LogDbg
# hares -modify WebSphereMQFTE LogDbg DBG_5
```

Sample Configurations

This appendix includes the following topics:

- [About sample configurations for the agents for WebSphere MQ FTE](#)
- [Sample agent type definition for WebSphere MQ FTE](#)
- [Sample configuration in a VCS environment](#)
- [Sample configuration in a VCS One environment](#)
- [Sample service group configurations](#)

About sample configurations for the agents for WebSphere MQ FTE

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 FTE. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition for WebSphere MQ FTE

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 `WebSphereMQFTETypes.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 4.x

```
type WebSphereMQFTE
(
    static str ArgList[] = { ResLogLevel, State, IState, FTEAgentName,
                           CoordinationQM, AgentQM, HomeDir, DataDir,
                           FTEUser, EnvFile, MonitorProgram,
                           SecondLevelMonitor }

    str ResLogLevel      = INFO
    str FTEAgentName
    str CoordinationQM
    str AgentQM
    str HomeDir          = "/opt/IBM/WMQFTE"
    str DataDir          = "/var/IBM/WMQFTE"
    str FTEUser
    str EnvFile
    str MonitorProgram
    int SecondLevelMonitor = 0
)
```

For VCS 5.x

```
type WebSphere MQFTE
(
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
    static str AgentDirectory =
"/opt/VRTSagents/ha/bin/WebSphereMQFTE"
    static str ArgList[] = { ResLogLevel, State, IState, FTEAgentName,
                           CoordinationQM, AgentQM, HomeDir, DataDir,
                           FTEUser, EnvFile, MonitorProgram,
                           SecondLevelMonitor }

    str ResLogLevel = INFO
    str FTEAgentName
    str CoordinationQM
    str AgentQM
    str HomeDir = "/opt/IBM/WMQFTE"
    str DataDir = "/var/IBM/WMQFTE"
    str FTEUser
    str EnvFile
    str MonitorProgram
    int SecondLevelMonitor = 0
)
```

)

VCS One

After installing the agent, go to the `/etc/VRTSagents/ha/conf/WebSphereMQ6/` directory to view the `WebSphereMQ6Types.platform.xml` agent definition file.

Sample configuration in a VCS environment

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

```
group WMQFTE_Sol_GRP (
    SystemList = { system_A = 0, System_B = 1 }
)
DiskGroup DG_WMQFTE (
    DiskGroup = WMQFTE_DG
)
Mount Mount_WMQFTE (
    MountPoint = "/opt/IBM/WMQFTE"
    BlockDevice = "/dev/vx/dsk/WMQFTE/MQFTE_VOL"
    FSType = vxfs
    FsckOpt = "-y"
)
Volume Volume_WMQFTE (
    Volume =MQFTE_VOL
    DiskGroup = WMQFTE_DG
)
WebSphereMQFTE WMQFTE_Agent01_Res (
    Critical = 1
    FTEAgentName = AGENT1
    CoordinationQM = QM1
    AgentQM = QM1
    FTEUser = mqm
    HomeDir = "/opt/IBM/WMQFTE"
    DataDir = "/var/IBM/WMQFTE"
)
Mount_WMQFTE requires Volume_WMQFTE
Volum_WMQFTE requires DG_WMQFTE
WMQFTE_Agent01_Res requires Mount_WMQFTE
requires group WMQ_QueueManager_GRP online global soft
```

```
// resource dependency tree
// group WMQFTE_Sol_GRP
// {
//     WebSphereMQFTE WMQFTE_Agent01_Res
//     {
//         Mount Mount_WMQFTE
//     }
//     Volume Volume_WMQFTE
//     {
//         DiskGroup DG_WMQFTE
//     }
// }
// }

group WMQ_QueueManager_GRP (
SystemList = { system_A = 0, System_B = 1 }
)

Application res_listener (
Critical = 0
User = mqm
StartProgram = "/opt/mqm/bin/runmqcls -r -m QM1 -t TCP -p 1415 &"
StopProgram = "/opt/mqm/bin/endmqcls -m QM1"
MonitorProcesses = { "/opt/mqm/bin/runmqcls -r -m QM1 -t TCP -p 1415" }
)

WebSphereMQ6 mq_res (
    Critical = 0
    QueueManager = QM1
    MQVer = "7.0"
)

DiskGroup DG_WMQ (
DiskGroup = WMQ_DG
)

Mount Mount_WMQ (
MountPoint = "/opt/IBM/WMQ"
BlockDevice = "/dev/vx/dsk/WMQ/MQ_VOL"
FSType = vxfs
EsckOpt = "-y"
)

Volume Volume_WMQ (
Volume =MQ_VOL
DiskGroup = WMQ_DG
```

)

```
Mount_WMQ requires Volume_WMQ  
Volum_WMQ requires DG_WMQ  
mq_Res requires Mount_WMQ  
mq_res requires res_listener
```

Sample configuration in a VCS One environment

To view a sample VCS One configuration file (main.xml) with an MQ Listener and a WebSphere MQ Queue Manager, go to the /etc/VRTSagents/ha/conf/WebSphereMQ6/ directory.

Sample service group configurations

[Figure A-1](#) and [Figure A-2](#) represent a sample service group that shows WebSphere MQ FTE agent resources.

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

Figure A-1 Sample Service group configuration

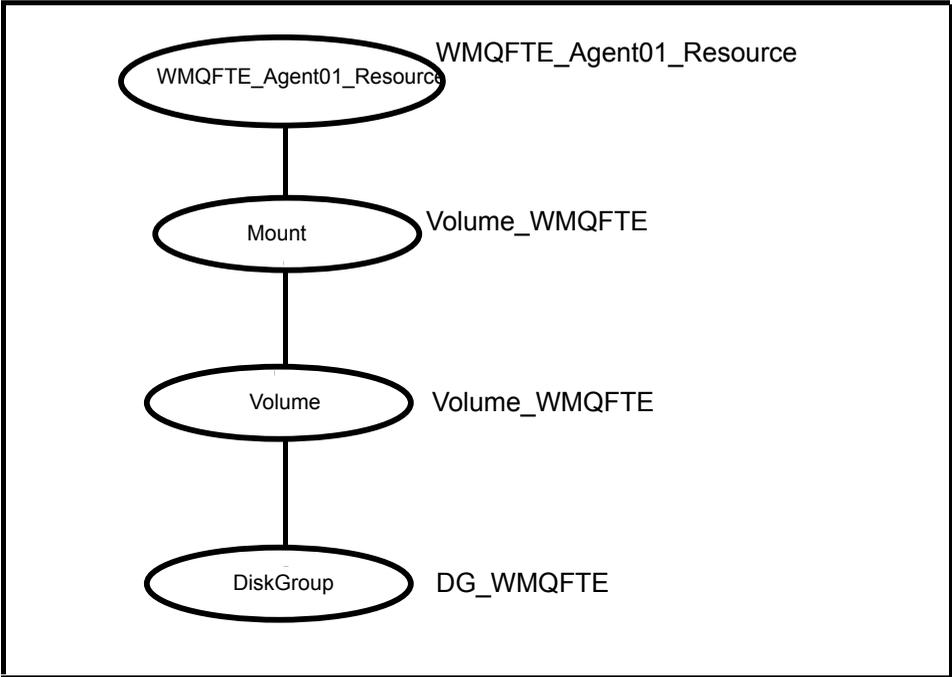


Figure A-2 Sample service group configuration in properties view

The screenshot shows the 'Properties View' for a service group named 'WMQFTE_Agent01_Res'. It is divided into three sections: 'Key Attributes', 'Type Specific Attributes', and 'Overridden Attributes'. The 'Key Attributes' section contains one attribute: 'Critical' with a value of 'true'. The 'Type Specific Attributes' section contains ten attributes, each with a value and edit/help icons. The 'Overridden Attributes' section is empty, indicating no attributes are overridden.

Properties View: WMQFTE_Agent01_Res Show all attributes

Key Attributes

Attribute	Value	Edit	Help
• Critical	: true		

Type Specific Attributes

Attribute	Value	Edit	Help
• ResLogLevel	: INFO		
• FTEAgentName	: AGENT1		
• CoordinationQM	: QM1		
• AgentQM	: QM1		
• HomeDir	: /opt/IBM/WMQFTE		
• DataDir	: /var/IBM/WMQFTE		
• FTEUser	: mqm		
• EnvFile	:		
• MonitorProgram	:		
• SecondLevelMonitor	: 0		

Overridden Attributes

No Attributes are Overridden.

Index

A

- about ACC library 14
- ACC library
 - installing 15
 - removing 27
- add
 - resource type
 - Policy Master on UNIX 20
 - Policy Master Server on Windows 22
- agent
 - i18n support 14
 - importing agent types files 32
 - installing, VCS environment 16
 - overview 7
 - uninstalling, VCS environment 22
 - upgrading 27
- agent attributes
 - EnvFile 37
 - FTEAgentName 35
 - SecondLevelMonitor 37
- agent configuration file
 - importing 32
- agent functions
 - clean 12
 - monitor 11
 - offline 11
 - online 11
- agent installation
 - general requirements 13
 - steps to install 16

B

- before
 - configuring the service groups 40

C

- clustering
 - WebSphere MQ FTE 41
- configuring monitor function 37

E

- executing custom monitor program 37

I

- install
 - agent package
 - using CLI 19
 - using installer 18

L

- logs
 - reviewing cluster log files 45
 - reviewing error log files 45
 - using trace level logging 45

R

- remove
 - agent package
 - using CLI 25
 - using installer 24
 - resource type
 - Policy Master on UNIX 26

S

- sample configurations
 - sample file 49
 - service group 51
 - VCS environment 49
 - VCS One environment 51
- starting the WebSphere MQ FTE agent instance
 - outside a cluster 44

T

- troubleshooting
 - meeting prerequisites 43
 - reviewing error log files 45
 - reviewing cluster log files 45
 - using trace level logging 45
 - using correct software 43

U

- uninstalling agent, VCS environment 22
- upgrading agent 27
 - VCS One environment 29

W

- WebSphere MQ FTE
 - about 8
- WebSphere MQ FTE agent
 - configuring resources 44
 - starting instance outside cluster 44