Veritas™ High Availability Agent for Tibco EMS Server Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

5.1

Symantec™

Veritas High Availability Agent for Tibco EMS Server Installation and Configuration Guide

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

Agent Version: 5.1.0.0

Document version: 5.1.0.0.1

Legal Notice

Technical Support

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

Symantec's support offerings include the following:

■ A range of support options that give you the flexibility to select the right amount of service for any size organization

■ Telephone and/or Web-based support that provides rapid response and up-to-the-minute information

■ Upgrade assurance that delivers software upgrades

■ Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis

■ Premium service offerings that include Account Management Services

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

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

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

Contacting Technical Support

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

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

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

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

■ Product release level

- Hardware information

- Available memory, disk space, and NIC information

- Operating system

- Version and patch level

- Network topology

- Router, gateway, and IP address information

- Problem description:

  - Error messages and log files

  - Troubleshooting that was performed before contacting Symantec

  - Recent software configuration changes and network changes

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

- Questions regarding product licensing or serialization

- Product registration updates, such as address or name changes

- General product information (features, language availability, local dealers)

- Latest information about product updates and upgrades

- Information about upgrade assurance and support contracts

- Information about the Symantec Buying Programs

- Advice about Symantec's technical support options

- Nontechnical presales questions

- Issues that are related to CD-ROMs or manuals

Support agreement resources

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

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

Documentation

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

http://sort.symantec.com/documents

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

docs@symantec.com

About Symantec Connect

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

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

Contents

Introducing the Veritas High Availability Agent for Tibco EMS Server

This chapter includes the following topics:

- About the Veritas agent for Tibco EMS Server

- Supported software

- Features of the Veritas agent for Tibco EMS Server

- How the agent makes Tibco EMS Server highly available

- Tibco EMS Server agent functions

About the Veritas agent for Tibco EMS Server

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

The Veritas agent for Tibco EMS Server provides high availability for EMS server in a clustered environment. The Veritas High Availability agent brings specific instances of the EMS Server online, monitors the instance, and brings it offline. The Veritas High Availability agent monitors the processes of the EMS Server instance and shuts down the EMS Server in case of a failure.

See the following Technical Support TechNote for the latest updates or software issues for this agent:

http://www.symantec.com/business/support/index?page=content&id=TECH46455

Supported software

The Veritas agent for TibcoEMS server supports the following software versions:

Veritas Cluster Server	■ AIX—VCS 4.0, 5.0, 5.1 ■ HP-UX—VCS 4.1, 5.0 ■ Linux—VCS 4.0, 4.1, 5.0, 5.1 ■ Solaris—VCS 4.0, 4.1, 5.0, 5.1 and all intermediate Maintenance Packs of these releases.
ACC Library	5.2.2.0 and later Review the ACC Library version for i18n support. See "Prerequisites for enabling i18n support" on page 22.
Operating Systems	■ AIX 5.1, 5.2, 5.3, 6.1 on pSeries ■ HP-UX 11i v2, 11i v3 on Itanium and PA-RISC ■ Red Hat Enterprise Linux 3.0, 4.0, 5.0 on Intel ■ SUSE Linux Enterprise Server 9, 10 ■ Solaris 8, 9, 10 on SPARC and x86
TIBCO	TIBCO Enterprise Message Service 4.4.0, 4.4.1, 4.4.2, 4.4.3, 5.0.0, 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 6.0.0 and all intermediate minor versions of this release.

Features of the Veritas agent for Tibco EMS Server

The following are the features of the Veritas agent for Tibco EMS Server:

- Support for validation of attributes that are based on agent functions.
  The agent can validate attributes in each agent function before the actual data processing starts.

- Support for First Failure Data Capture (FFDC)
  In case of a fault, the agent generates a huge volume of the debug logs that enable troubleshooting of the fault.

- Support for Fast First Level Monitor (FFLM)
  The agent maintains PID files based on search patterns to expedite the monitoring process.

- Support for external user-supplied monitor utilities
  The agent enables user-specified monitor utilities to be plugged in, in addition to the built-in monitoring logic. This enables administrators to completely customize the monitoring of the application.

■ Delayed agent function
The agent manages the first monitor after online for slow initializing applications.

How the agent makes Tibco EMS Server highly available

The Veritas agent for Tibco EMS Server continuously monitors the EMS Server processes to verify that they function properly.

The agent mainly provides the following levels of application monitoring:

■ Primary or Basic monitoring
This mode has Process check monitoring option. With the default Process check option, the agent verifies that the Tibco EMS Server process is present in the process table. Process check cannot detect whether process is in hung or stopped state.

■ Optional Secondary or Detail monitoring
In this mode, the agent runs a tibemsadmin utility to verify the status of Tibco EMS Server. The agent reports the application as online if the tibemsadmin utility agent is able to connect to the Tibco EMS server, otherwise the monitoring routine reports the application as offline.

■ Custom Monitoring
In this mode, the agent can perform a customized check using a user-supplied monitoring utility. The agent detects application failure if the monitoring routine reports an improper function of the Tibco EMS server processes. When this application failure occurs, the Tibco EMS Server service group could be failed over to another node in the cluster.
Thus, the agent ensures high availability for Tibco EMS Server.

Tibco EMS Server agent functions

The agent consists of resource type declarations and agent executables. The agent executables implement online, offline, monitor, and clean operations.

Online

The online function performs the following tasks:

■ Verifies whether the required attributes are set correctly.

- Performs the preliminary check to ensure that the Tibco EMS server is not online on the specified node in the cluster.

- Attempts to start the Tibco EMS server instance with the following command:

  EMSHomeDir/tibemsd -config *ConfigFile StartOptions* &

  The command is always executed in the context of the User.

- Uses the `tibemsd` command available at the EMSHomeDir attribute value to start the Tibco EMS server.

  The startup parameters to tibemsd command are passed using the StartOptions attribute value.

- Sources a file that the EnvFile attribute specifies. This environment file ensures that the required shell environment variables are properly set before executing the start script.

- Ensures that the Tibco EMS server instance is up and running successfully. The operation uses the wait period that the OnlineTimeout attribute specifies, to enable the Tibco EMS server instance to initialize fully before allowing the monitor function to probe the resource.

Offline

The offline operation performs the following tasks:

- Verifies that the Tibco EMS server instance is not already offline.

- Attempts to stop the Tibco EMS server instance with the following command.

  EMSHomeDir/tibemsadmin -server *TibEmsServerUrl* -user *TibUser*

  TibemsadminOptions << EOF

  TibPassword

  shutdown

  yes

  exit

  EOF

  After connecting to the EMS server using the `tibemsadmin` command, the agent issues shutdown command.

- Sources a file that the EnvFile attribute specifies. This environment file ensures that the required shell environment variables are properly set before executing the stop script.

Note: The TibEmsServerUrl attribute specifies the server URL, when connecting to Tibco EMS server. The server uses the TibUser attribute and the TibPassword attribute values for authentication while connecting to the EMS server.

- Uses the TibemsadminOptions attribute to pass the optional parameters to the tibemsadmin command.

- Ensures that the TibcoEMS server is offline. The operation uses a wait period that the OfflineTimeout attribute specifies, to allow the Tibco EMS server instance to complete the offline sequence before allowing further probing of the resource.

Monitor

The monitor function monitors the states of the EMS Server instance running on all nodes within the cluster.

The monitor operation performs following tasks:

- Conducts a first level check to determine that the EMS Server processes that the user specified in the TibUser attribute owns, are running on the system in the cluster. If the first level check does not find these processes running on the node, the check exits immediately, and reports the instance as OFFLINE.

- Uses the ConfigFile attribute value to uniquely identify the Tibco EMS server instance, when multiple Tibco EMS server instances are running on the same node.

- If the SecondLevelMonitor attribute is set to greater than 0, the monitor function performs a second-level check to determine the status of the EMS Server instance. The second level check tries to connect to Tibco EMS server instance using tibemsadmin command to ensure that the processes are truly available for Tibco EMS server instance.

- Depending upon the MonitorProgram attribute, the monitor function can perform a customized check using a user-supplied monitoring utility.
  See "Executing a customized monitoring program" on page 34.

Clean

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

The clean operation performs following tasks:

- Attempts to gracefully shut down the EMS Server instance.

- If a graceful shutdown fails, the clean function looks for all the processes running for the EMS Server instance, and ends the processes using the kill command.

Installing and configuring Tibco EMS Server for high availability

This chapter includes the following topics:

- About Tibco EMS

- Uniquely identifying Tibco EMS Server server instances

- About installing Tibco EMS Server for high availability

- Configuring the Tibco EMS Server for high availability

About Tibco EMS

TIBCO Enterprise Message Service software is a messaging software. It provides Java Message Service (JMS) compliant communications across platforms and application technologies. It provides distributed and reliable architecture, with support for load-balancing, routing, and fault tolerant configurations that together remove single points of failure. It supports request and reply and publish and subscribe interactions, synchronous and asynchronous messaging, multicast deployments and different levels of reliable messaging. These capabilities enable developers and administrators to support different types of service protocols on the same platform.

It allows different resources from different vendors to participate in a single transaction. It provides multi-protocol support. It supports many open standards. It offers native support for development technologies and platforms. It provides full SSL support for client-to-server and server-to-server connectivity and plug-in security capability for custom-built authentication (JAAS) and authorization. Its

built-in monitoring and management capabilities provide detailed administrative functions and statistics and support automation through an administrative API or command-line shell.

Uniquely identifying Tibco EMS Server server instances

For multiple TibcoEMS server instance running concurrently on a single node, the Veritas agent must be able to uniquely identify each of the TibcoEMS instance on that system. Each TibcoEMS server has a unique configuration file. The Veritas agent uses the `ConfigFile` attribute value to identify the TibcoEMS server instance uniquely.

Differentiating TibcoEMS server instances is important to identify each TibcoEMS server uniquely. When the Veritas agent kills the processes of a non-responsive or failed TibcoEMS instance in the absence of unique `ConfigFile` for each TibcoEMS Server instance, the Veritas agent may kill processes for more than one TibcoEMS server instance during a clean operation.

About installing Tibco EMS Server for high availability

Install Tibco EMS server on shared storage if you want the same installation to be shared among all the nodes in the cluster.

When installing Tibco EMS Server, ensure that the user name, UNIX uid, group name, and UNIX gid for the Tibco EMS installation user is the same on all the nodes.

For more details, refer to the product documentation.

Configuring the Tibco EMS Server for high availability

This section provides the information about the tasks you must perform to configure Tibco EMS Server for high availability.

Synchronizing accounts and services

Ensure that you synchoronize accounts and services in the following ways:

- Synchronize the TibcoEMS installation user accounts user name, UNIX uid, group name, and UNIX gid across all nodes in the cluster.

- The /etc/services entries should be consistent on all cluster nodes.

Removing physical host dependencies

Perform the following tasks to remove the physical host dependencies:

■ Update the `listen` tag in main Tibco configuration file specified as part of ConfigFile attribute. The form is for `listen tag is tcp://hostname:port`.

■ The listen tag value should be specified as part of the `TibEmsServerUrl` attribute.

Installing, upgrading, and removing the agent for Tibco EMS Server

This chapter includes the following topics:

- Before you install the Veritas agent for Tibco EMS Server
- 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

Before you install the Veritas agent for Tibco EMS Server

For VCS, do the following:

- Install and configure Veritas Cluster Server.
  For more information on installing and configuring Veritas Cluster Server, refer to the *Veritas Cluster Server Installation Guide*.

- Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation in the agentpack tarball.
  See "Installing the ACC library" on page 22.

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 22.

- 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	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/5.0
VCS 4.1	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/4.1
VCS 4.0	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/4.0

where *arch_dist* takes the following values:
'sol_sparc' for Solaris SPARC
'sol_x64' for Solaris x64
'generic' for HP-UX and Linux

Note: *arch_dist* is not applicable to AIX.

About the ACC library

The operations of a VCS 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 agent's tar file has already been extracted.

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 superuser.

2   Download ACC Library.

    You can download either the complete Agent Pack tarball or the individual
    ACCLib tarball from the following sites:

    ■   For the complete Agent Pack tarball: Symantec Operations Readiness
        Tools (SORT) site (https://sort.symantec.com/agents) or the FileConnect
        site (https://fileconnect.symantec.com).

    ■   For the individual agent tarball: https://sort.symantec.com/agents.

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

    AIX   *cd1*/aix/vcs/application/acc_library/*version*_library/pkgs

    HP-UX   *cd1*/hpux/generic/vcs/application/acc_library/*version*_library/pkgs

    Linux   *cd1*/linux/generic/vcs/application/acc_library/*version*_library/rpms

    Solaris   *cd1*/solaris/*dist_arch*/vcs/application/acc_library/*version*_library/pkgs

                where *dist_arch* is sol_sparc or sol_x64.

4   If you downloaded the individual ACCLib tarball, navigate to the pkgs directory
    (for AIX, HP-UX, 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

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

    Solaris   # pkgadd -d VRTSacclib.pkg

Installing the agent in a VCS environment

Install the agent for Tibco EMS Server on each node in the cluster.

To install the agent in a VCS environment

1   Download the agent.

    You can download either the complete Agent Pack tarball or an individual
    agent tarball from the following sites:

    ■ For the complete Agent Pack tarball: the Symantec Operations Readiness
      Tools (SORT) site (https://sort.symantec.com/agents) or the FileConnect
      site (https://fileconnect.symantec.com)

    ■ For the individual agent tarball: https://sort.symantec.com/agents

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

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

AIX	`cd1/aix/vcs/application/tibco_agent/` `vcs_version/version_agent/pkgs`
HP-UX	`cd1/hpux/generic/vcs/application/tibco_agent/` `vcs_version/version_agent/pkgs`
Linux	`cd1/linux/generic/vcs/application/tibco_agent/` `vcs_version/version_agent/rpms`
Solaris	`cd1/solaris/dist_arch/vcs/application/tibco_agent/` `vcs_version/version_agent/pkgs`

    If you downloaded the individual agent tarball, navigate to the pkgs directory
    (for AIX, HP-UX, and Solaris), or rpms directory (for Linux).

4   Log in as superuser.

5   Install the package.

AIX	`# installp -ac -d VRTSvcstib.rte.bff VRTSvcstib.rte`
HP-UX	`# swinstall -s 'pwd' VRTSvcstib`
Linux	`# rpm -ihv \` `VRTSvcstib-AgentVersion-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d . VRTSvcstib`

Removing the agent in a VCS environment

You must uninstall the agent for Tibco EMS Server from a cluster while the cluster is active.

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 Tibco EMS Server resources from the cluster. Use the following command to verify that all resources have been removed:

    ```
    # hares -list Type=TibcoEMS
    ```

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

    ```
    # hatype -delete TibcoEMS
    ```

    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 Tibco EMS Server from each node in the cluster.

    Execute the following command to uninstall the agent:

AIX	`# installp -u VRTSvcstib.rte`
HP-UX	`# swremove VRTSvcstib`
Linux	`# rpm -e VRTSvcstib`
Solaris	`# pkgrm VRTSvcstib`

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`
Linux	`# rpm -e VRTSacclib`
Solaris	`# pkgrm VRTSacclib`

Configuring the agent for Tibco EMS Server

This chapter includes the following topics:

- About configuring the Veritas agent for Tibco EMS Server
- Importing the agent types files in a VCS environment
- Tibco EMS Server agent attributes
- Executing a customized monitoring program

About configuring the Veritas agent for Tibco EMS Server

After installing the Veritas agent for Tibco EMS Server, you must import the agent type configuration file. After importing this file, you can create and configure Tibco EMS Server resources. Before you configure a resource, review the attributes table that describes the resource type and its attributes.

See "About sample configurations for the agent for Tibco EMS Server" on page 43.

Importing the agent types files in a VCS environment

To use the agent for Tibco EMS Server, you must import the agent types file into the cluster.

To import the agent types file using the Veritas Cluster Server graphical user interface

1   Start the Veritas Cluster Manager 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_TibcoEMS/
	■ HP-UX	TibcoEMSTypes.cf
	■ Linux	
	■ Solaris	
VCS 5.0	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/TibcoEMS/ TibcoEMSTypes50.cf
VCS 5.1	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/TibcoEMS/ TibcoEMSTypes51.cf

4   Click **Import**.

5   Save the VCS configuration.

The Tibco EMS agent type is now imported to the VCS engine.

You can now create Tibco EMS Server resources. For additional information about using the VCS GUI, refer to the *Veritas Cluster Server User's Guide*.

To import the agent types file using the Veritas Cluster Server command line interface (CLI), perform the following steps.

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 from the following location:

VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample_TibcoEMS/
	■ HP-UX	TibcoEMSTypes.cf
	■ Linux	
	■ Solaris	

VCS 5.0	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/TibcoEMS/ TibcoEMSTypes50.cf
VCS 5.1	■ Solaris SPARC and x64	/etc/VRTSagents/ha/conf/TibcoEMS/ TibcoEMSTypes51.cf

4   Create a dummy main.cf file:

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

5   Create the EMS server resource type as follows:

```
# hacf -verify .

# haconf -makerw

# sh main.cmd

# haconf -dump
```

The Tibco EMS agent type is now imported to the VCS engine.

You can now create Tibco EMS Server resources. For additional information about using the VCS CLI, refer to the *Veritas Cluster Server User's Guide*.

Tibco EMS Server agent attributes

Refer to the required attributes and optional attributes while configuring the agent for Tibco EMS Server.

Table 4-1 lists the required attributes for the Tibco agent.

Table 4-1   Required attributes

Required attributes	Description
EMSHomeDir	Specifies the full path of the directory where the tibemsd binary file and the tibemsadmin utility is located.
	Type and Dimension: string-scaler
	Default: ""
	Example: /home/tibco/ems/bin

Table 4-1 Required attributes *(continued)*

Required attributes	Description
ConfigFile	Specifes the full path and the file name of the main configuration file tibemsd.conf for the Tibco Enterprise Message Server.
	Type and Dimension: string-scaler
	Default: ""
	Example: /home/tibco/ems/bin/tibemsd.conf
User	UNIX user name that the VHA agent will use for execute the programs for managing a Tibco Enterprise Message Server.
	The user name must 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. VHA Agent entry points use the getpwnam (3c) 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.
	The supported shell environments are: ksh, sh, and csh.
	Type and Dimension: string-scaler
	Default: ""
	Example: tibco
ResLogLevel	Specifies the logging detail performed by the agent for the resource.
	The valid values are as follows:
	■ ERROR: Only logs error messages. ■ WARN: Logs above plus warning messages. ■ INFO: Logs above plus informational messages. ■ TRACE: Logs above plus trace messages. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations.
	Type and Dimension: string-scaler
	Default: INFO
	Example: TRACE

Table 4-1   Required attributes *(continued)*

Required attributes	Description
TibEmsServerUrl	Description: Tibco EMS server URL, default is local server. During Offline and second-level monitoring, this URL is used to specify the -server parameter for tibemsadmin utility.
	Type and Dimension: string-scaler
	Default: ""
	Example: tcp://virtualhost:7222
TibUser	Tibco user name to connect to server. During Offline and second-level monitoring, this TibUser is used to specify the -user parameter for tibemsadmin utility.
	Type and Dimension: string-scaler
	Default: ""
	Example: admin
TibPassword	Password of the Tibco operator who is connecting to EMS server. During Offline and second-level monitoring, this Password is used to specify the -password parameter for tibemsadmin utility. The value of this attribute need to be encrypted using VCS provided $VCSHOME/bin/vcsencrypt utility
	Type and Dimension: string-scaler
	Default: ""
	Example: HTIvKTlTNnINjNKnL

Table 4-2 lists the optional attributes for the Tibco agent.

Table 4-2   Optional attributes

Optional attributes	Description
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the configured TibcoEMS instance. The numeric value specifies how often the monitoring routines are 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. This interpretation may be extended to other values. **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 second level check is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. You can run the second level check once a month, if required. Also, verify the second level monitoring utility before enabling second level monitor. Type and Dimension: integer-scaler Default: 0 Example: 1

Table 4-2   Optional attributes *(continued)*

Optional attributes	Description
MonitorProgram	Specifies the full path and file name of an external, user-supplied monitor program. If specified, the monitor entry point executes this file to perform an additional server state check. There are no restrictions for the actions the external monitor program performs to determine the state of a TibcoEMS instance server. The only constraint is that the external monitor program must return one of the following integer values: ■ If the exit code is 0 the TibcoEMS server is online. ■ If the exit code is 110 the TibcoEMS server is online. ■ If the exit code is 100 the TibcoEMS server is offline. ■ If the exit code is 1 the TibcoEMS server is offline. ■ All other: TibcoEMS server is unknown. Symantec recommends storing the external monitor utility on the shared disk directory to ensure the file is always available on the online system. Arguments are supported. See "Executing a customized monitoring program" on page 34. Type and Dimension: string-scaler Default: "" Example: /home/tibco/ems/myMonitor.sh
StartOptions	List of tibemsd startup parameters. During startup of Tibco Enterprise Message Server, these startup parameters will be passed to tibemsd command. The command options to tibemsd are similar to the parameters you specify in tibemsd.conf, and the command options override any value specified in the parameters. By default the agent specifies -config parameter, hence this parameter should not be specified as part of StartOptions again. Type and Dimension: association-scaler Default: "" Example: -ft_active tcp://tibhost:7245 -ssl_trace

Table 4-2   Optional attributes *(continued)*

Optional attributes	Description
EnvFile	Specifies the full path to the file that must be sourced with the UNIX shell. Source this file to set the environment before executing Tibco Enterprise Message Service command. Symantec recommends to store the file on a shared disk.
	The shell environments supported are: ksh, sh, and csh.
	Type and Dimension: string-scaler
	Default: ""
	Example: /home/Tibco/ems/envfile
TibemsadminOptions	List of tibemsadmin command startup parameters. During offline and second-level monitoring, these startup parameters will be passed to tibemsadmin utility. By default the agent specifies -server, -user and -password parameters, hence these parameter should not be specified as part of TibemsadminOptions again.
	Type and Dimension: association-scaler
	Default: ""
	Example: -ssl_trusted /home/tibco/ssl/certificate1 -ssl_identity /home/tibco/ssl/certificate2 -ssl_trace

Executing a customized monitoring program

The monitor function executes a custom monitor program to perform an additional Tibco EMS Server state check. The monitor function executes the utility specified in the MonitorProgram attribute, if:

The monitor function interprets the utility exit code as follows:

110 or 0	Tibco EMS server instance is online
100 or 1	Tibco EMS server instance is offline
99	Tibco EMS server instance is unknown
Any other value	Tibco EMS server instance is unknown

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

Configuring the service groups for Tibco EMS Server

This chapter includes the following topics:

- About configuring service groups for Tibco EMS Server
- Before configuring the service groups for Tibco EMS Server
- Configuring service groups for Tibco EMS Server

About configuring service groups for Tibco EMS Server

Configuring the Tibco EMS Server service group involves creating the Tibco EMS 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 Tibco EMS Server

Before you configure the Tibco EMS Server service group, you must:

- Verify that VCS is installed and configured on all nodes in the cluster where you will configure the service group.

Refer to the *Veritas Cluster Server Installation Guide* for more information.

- Verify that Tibco EMS server is installed and configured on all the systems in the cluster on the same path that will host a Tibco EMS service group.
  See "About installing Tibco EMS Server for high availability" on page 18.

- Verify that the Veritas agent for Tibco EMS Server is installed on all nodes in the cluster.
  See "Installing the agent in a VCS environment" on page 23.

- Verify that the type definition for the Veritas agent for Tibco EMS Server is imported into the VCS engine.
  See "Importing the agent types files in a VCS environment" on page 27.

Configuring service groups for Tibco EMS Server

A cluster handles Tibco EMS server failover scenarios. It provides application failover by encapsulating the application resources required into a service group. A service group is a logical setup containing all the resources that support Tibco EMS Server instance in a clustered environment. A service group can contain a set of dependent resources such as disk groups, 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 to apply patches. The service group includes logical information about the dependencies between the application components.

In a clustered environment, you can configure a Tibco EMS server in the following configuration:

- Failover configuration for Tibco EMS Server
- Cluster File System (CFS) based fault tolerant configuration for Tibco EMS Server

Failover configuration for Tibco EMS Server

In a failover configuration, when the Tibco EMS server faults on one node, the storage infrastructure is moved to the failover node and the EMS server is started on the failover node. Tibco EMS server is fully operational once the Tibco EMS resource is online on failover node. The storage resources, the network resources, and the TibcoEMS resources are part of the same service group which fails over across VCS nodes.

Cluster File System (CFS) based fault tolerant configuration for Tibco EMS Server

In a Cluster File System based configuration, you can arrange TIBCO Enterprise Message Service servers for fault-tolerant operation by configuring a pair of servers, one primary and one backup. The primary server accepts client connections and interacts with clients to deliver messages. If the primary server fails, the backup server resumes operation in its place. However, it does not support more than two servers.

Separate service groups are used to configure the Tibco EMS servers. These application service groups depend on underlying storage, which is CFS based and it is shared across VCS nodes. Storage resources are configured as part of Tibco infrastructure group. When the primary Tibco EMS server faults, the backup server readily resumes operation in its place. Since the storage infrastructure is already available to backup server, the failover time is reduced in this configuration.

The various ft_* parameters in the tibemsd.conf file dictate how fault tolerance is set up. To enable fault tolerance, you must set the server, store, and ft_activate parameters. The server parameter is set to an arbitrary name (TIBCO Enterprise Message Service server name). It should be identical in the configuration file for both the primary server and the standby server. The shared data store is designated by the store argument and it should point to a shared file system or a clustered file system. The ft_active parameter should point to the other server in a primary or standby pair, such that on the primary it should point to the standby and on the standby it should point to the primary. During the startup, the servers communicate with each other and decide the current primary server and the current standby server. By default, the first node that starts is the primary server.

The main configuration files excerpts are as follows:

- Primary tibemsd_tibems01.conf

```
server = EMS01
store = /home/tibco/ems01/datastore
listen = tcp://tibhostpri:7222
ft_active = tcp://tibhostsec:7225
```

- Secondary tibemsd_tibems02.conf

```
server = EMS01
store = /home/tibco/ems01/datastore
listen = tcp://tibhostsec:7225
ft_active = tcp://tibhostpri:7222
```

For details about other features, refer to the *TIBCO Enterprise Message Service User's Guide.*

For details about CFS based fault tolerant configuration for Tibco EMS server, refer to the *Application Note: Symantec™ High Availability Solution for TIBCO Enterprise Message Service™.*

See "Sample service group configuration for Tibco EMS server in Failover setup" on page 45.

See "Sample service group configuration for Tibco EMS server in CFS based fault tolerant setup" on page 51.

Troubleshooting the agent for Tibco EMS Server

This chapter includes the following topics:

■ Using the correct software and operating system versions

■ Meeting prerequisites

■ Configuring Tibco EMS Server resources

■ Starting the Tibco EMS Server instance outside a cluster

■ Reviewing error log files

Using the correct software and operating system versions

Ensure that no issues arise due to incorrect software and operating system versions. For the correct versions of operating system and software to be installed on the resource systems:

See "Supported software" on page 12.

Meeting prerequisites

Before installing the agent for Tibco EMS Server, double check that you meet the prerequisites.

For example, you must install the ACC library on VCS before installing the agent for Tibco EMS Server.

See "Before you install the Veritas agent for Tibco EMS Server" on page 21.

Configuring Tibco EMS Server resources

Before using Tibco EMS Server resources, ensure that you configure the resources properly. For a list of attributes used to configure all Tibco EMS Server resources, refer to the agent attributes.

Starting the Tibco EMS Server 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 Tibco EMS Server instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the Tibco EMS Server 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 EMS server instance outside the cluster framework, is illustrated as follows.

To restart the Tibco EMS server outside the framework

1   Log in to the Tibco EMS node as an User.

     # su User

2   Source the environment file.

     # . EnvFile

3   Start the Tibco EMS server.

     # cd EMSHomeDir

     # EMSHomeDir/tibemsd -config *config-file StartOptions*

If the Tibco EMS server works properly outside the cluster framework, you can then attempt to implement the Tibco EMS server within the cluster framework.

Reviewing error log files

If you face problems while using Tibco EMS Server or the agent for Tibco EMS Server, 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 Tibco EMS Server, you can access the engine log file for more information about a particular resource. The engine log file is located at `/var/VRTSvcs/log/engine_A.log`.

You can also access the Tibco EMS Server agent log file for more detailed information. The agent log file is located at /var/VRTSvcs/log/TibcoEMS_A.log

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each Tibco EMS Server resource. You can set this attribute to TRACE, which enables very detailed and verbose logging.

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

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

   You can also contact Symantec support for more help.

Using Tibco EMS Server log files

By default the log file is disabled in Tibco EMS server. To enable logging, update the Tibco EMS server configuration file.

For more details about Tibco log file and tracing parameters, refer to the Tibco documentation.

Sample Configurations

This appendix includes the following topics:

About sample configurations for the agent for Tibco EMS Server

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agent for Tibco EMS Server. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition

This section lists the sample agent type definition files for EMS Server agent on different versions of VCS.

For VCS 4.x

```
type TibcoEMS (
        static str ContainerType = Zone
static str ArgList[] = { ResLogLevel, State, IState, User, EnvFile,
EMSHomeDir, ConfigFile, TibEmsServerUrl, SecondLevelMonitor, TibUser,
TibPassword, MonitorProgram, StartOptions, TibemsadminOptions }
 str ResLogLevel = INFO
 str User
 str EnvFile
 str EMSHomeDir
 str ConfigFile
 str TibEmsServerUrl
 int SecondLevelMonitor = 0
 str TibUser
 str TibPassword
 str MonitorProgram
 str StartOptions
 str TibemsadminOptions
 str ContainerName
)
```

For VCS 5.0

```
type TibcoEMS (
        static boolean AEPTimeout = 1
        static str ContainerType = Zone
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/TibcoEMS"
static str ArgList[] = { ResLogLevel, State, IState, User, EnvFile,
EMSHomeDir, ConfigFile, TibEmsServerUrl, SecondLevelMonitor, TibUser,
TibPassword, MonitorProgram, StartOptions, TibemsadminOptions }
 str ResLogLevel = INFO
 str User
 str EnvFile
 str EMSHomeDir
 str ConfigFile
 str TibEmsServerUrl
 int SecondLevelMonitor = 0
 str TibUser
```

```
 str TibPassword
 str MonitorProgram
 str StartOptions
 str TibemsadminOptions
        str ContainerName
)
```

For VCS 5.1

```
type TibcoEMS (
        static int ContainerOpts {} = { RunInContainer = 1, PassCInfo = 0 }
        static boolean AEPTimeout = 1
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/TibcoEMS"
static str ArgList[] = { ResLogLevel, State, IState, User, EnvFile,
EMSHomeDir, ConfigFile, TibEmsServerUrl, SecondLevelMonitor, TibUser,
TibPassword, MonitorProgram, StartOptions, TibemsadminOptions }
 str ResLogLevel = INFO
 str User
 str EnvFile
 str EMSHomeDir
 str ConfigFile
 str TibEmsServerUrl
 int SecondLevelMonitor = 0
 str TibUser
 str TibPassword
 str MonitorProgram
 str StartOptions
 str TibemsadminOptions
)
```

Sample service group configuration for Tibco EMS server in Failover setup

This section provides a sample configuration for Tibco EMS server agent in failover setup. The sample configuration depicts a graphical view of the resource types, resources, and resource dependencies within the service group.

```
include "types.cf"
include "TibcoEMSTypes.cf"

cluster system01_02 (
 UserNames = { admin = IhiAhcHeiDiiGqiChf }
```

```
 Administrators = { admin }
 HacliUserLevel = COMMANDROOT
 )

system system01 (
 )

system system02 (
 )

group TibcoSG (
 SystemList = { system01 = 0, system02 = 1 }
 )

 DiskGroup tib_dg (
  DiskGroup = tibco_diskgroup
  )

 IP tib_ip_res (
  Device = bge0
  Address = "10.209.73.81"
  NetMask = "255.255.252.0"
  )

 Mount tib_mnt (
  MountPoint = "/home/tibco"
  BlockDevice = "/dev/vx/dsk/tibco_diskgroup/tibco_vol"
  FSType = vxfs
  FsckOpt = "-y"
  )

 NIC tib_nic_res (
  Device = bge0
  )

 TibcoEMS tib_res (
  User = tibco
  EMSHomeDir = "/home/tibco/ems/bin"
  ConfigFile = "/home/tibco/ems/bin/tibemsd.conf"
  TibEmsServerUrl = "tcp://tibhost:7222"
  SecondLevelMonitor = 1
  TibUser = admin
  TibPassword = hvnTkvK
```

```
 )

tib_ip_res requires tib_nic_res
tib_mnt requires tib_dg
tib_res requires tib_ip_res
tib_res requires tib_mnt


// resource dependency tree
//
// group TibcoSG
// {
// TibcoEMS tib_res
//     {
//     Mount tib_mnt
//         {
//         DiskGroup tib_dg
//         }
//     IP tib_ip_res
//         {
//         NIC tib_nic_res
//         }
//     }
// }
```

Sample service group configuration for Tibco EMS server in Failover setup with Solaris Zones

This section provides a sample configuration for Tibco EMS server agent in failover setup with Solaris zone support.

```
include "types.cf"
include "TibcoEMSTypes.cf"

cluster system01_02 (
 UserNames = { admin = eLMeLGlIMhMMkUMgLJ }
 Administrators = { admin }
 HacliUserLevel = COMMANDROOT
 )

system system01 (
 )
```

```
system system02 (
 )

group tibcoZoneSG (
 SystemList = { system01 = 0 }
 ContainerInfo @system01 = { Name = tibzone, Type = Zone, Enabled = 1 }
 AutoStartList = { system01 }
 Administrators = { z_zoneres_system01 }
 )

 DiskGroup tib_data_dg (
  DiskGroup = tibco_diskgroup
  )

 DiskGroup zone_dg (
  DiskGroup = zone_diskgroup
  )

 IP tib_ip (
  Device = bge0
  Address = "10.209.73.81"
  NetMask = "255.255.252.0"
  )

 Mount tib_data_mnt (
  MountPoint = "/tibco"
  BlockDevice = "/dev/vx/dsk/tibco_diskgroup/tibco_vol"
  FSType = vxfs
  FsckOpt = "-y"
  )

 Mount zone_mnt (
  MountPoint = "/root/zones/tibzone"
  BlockDevice = "/dev/vx/dsk/zone_diskgroup/zone_vol"
  FSType = vxfs
  FsckOpt = "-y"
  )

 NIC tib_nic (
  Device = bge0
  )
```

```
TibcoEMS tib_zone_res (
 User = root
 EMSHomeDir = "/tibco/ems/bin"
 ConfigFile = "/tibco/ems/bin/tibemsd.conf"
 TibEmsServerUrl = "tcp://tibzone:7222"
 TibUser = admin
 TibPassword = CQIoFQf
 )

Zone zoneres (
 )

tib_data_mnt requires tib_data_dg
tib_ip requires zoneres
tib_zone_res requires tib_ip
zone_mnt requires zone_dg
zoneres requires tib_data_mnt
zoneres requires tib_nic
zoneres requires zone_mnt


// resource dependency tree
//
// group tibcoZoneSG
// {
// TibcoEMS tib_zone_res
//     {
//     IP tib_ip
//         {
//         Zone zoneres
//             {
//             Mount tib_data_mnt
//                 {
//                 DiskGroup tib_data_dg
//                 }
//             NIC tib_nic
//             Mount zone_mnt
//                 {
//                 DiskGroup zone_dg
//                 }
//             }
//         }
```

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

Sample resource dependency in a Failover configuration for Tibco EMS server

This section includes resource dependencies of Tibco service group in a failover configuration.

Figure A-1   Sample resource dependency in failover configuration
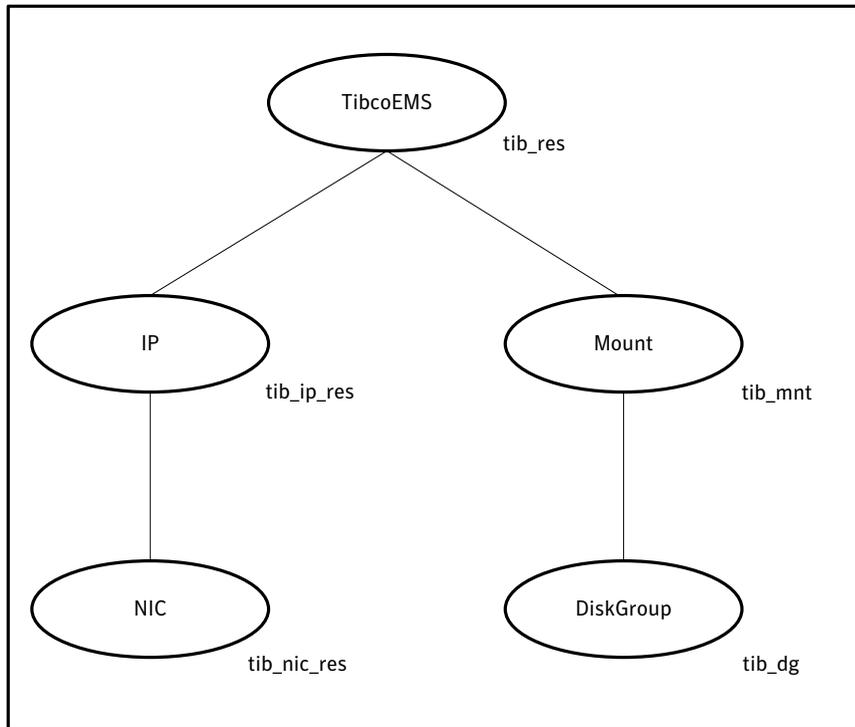
Sample resource dependency in a Failover configuration for Tibco EMS server with Solaris Zones

This section includes sample service groups with Solaris zone support.

Figure A-2 shows the resource dependencies of Tibco service group in a failover configuration with Solaris Zones.

Figure A-2   Sample service group with a Tibco EMS Server instance
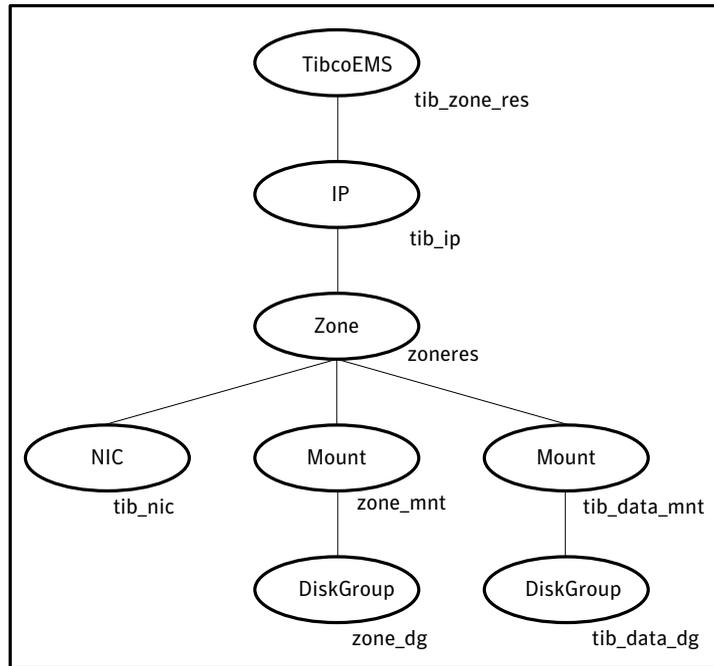
Sample service group configuration for Tibco EMS server in CFS based fault tolerant setup

This section provides a sample configuration for Tibco EMS server agent in Cluster File System based fault tolerant setup. The sample configuration depicts a graphical view of the resource types, resources, and resource dependencies within the service group.

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "TibcoEMSTypes.cf"

cluster system01_02 (
 UserNames = { admin = IhiAhcHeiDiiGqiChf }
 Administrators = { admin }
 HacliUserLevel = COMMANDROOT
 )
```

```
system system01 (
 )

system system02 (
 )

group TibcoSG_Pri (
 SystemList = { system01 = 0, system02 = 1 }
 )

 IP tib_ip_res_pri (
  Device = bge0
  Address = "10.209.73.81"
  NetMask = "255.255.252.0"
  )

 NIC tib_nic_res_pri (
  Device = bge0
  )

 TibcoEMS tib_res_pri (
  ResLogLevel = TRACE
  User = root
  EMSHomeDir = "/home/tibco/ems/bin"
  ConfigFile = "/home/tibco/ems/bin/tibemsd5.conf"
  TibEmsServerUrl = "tcp://tibhost:7222"
  SecondLevelMonitor = 1
  TibUser = admin
  TibPassword = hvnTkvK
  StartOptions = "-ft_active tcp://tibhostsec:7225"
  )

 requires group Tibco_infra online local firm
 tib_ip_res_pri requires tib_nic_res_pri
 tib_res_pri requires tib_ip_res_pri


 // resource dependency tree
 //
 // group TibcoSG_Pri
 // {
 // TibcoEMS tib_res_pri
```

```
//     {
//     IP tib_ip_res_pri
//        {
//        NIC tib_nic_res_pri
//        }
//     }
// }


group TibcoSG_Sec (
 SystemList = { system01 = 0, system02 = 1 }
 )

 IP tib_ip_res_sec (
  Device = bge0
  Address = "10.209.73.82"
  NetMask = "255.255.252.0"
  )

 NIC tib_nic_res_sec (
  Device = bge0
  )

 TibcoEMS tib_res_sec (
  User = root
  EMSHomeDir = "/home/tibco/ems/bin"
  ConfigFile = "/home/tibco/ems/bin/tibemsd1.conf"
  TibEmsServerUrl = "tcp://tibhostsec:7225"
  TibUser = admin
  TibPassword = hvnTkvK
  StartOptions = "-ft_active tcp://tibhost:7222"
  )

 requires group Tibco_infra online local firm
 tib_ip_res_sec requires tib_nic_res_sec
 tib_res_sec requires tib_ip_res_sec


 // resource dependency tree
 //
 // group TibcoSG_Sec
 // {
 // TibcoEMS tib_res_sec
```

```
//      {
//      IP tib_ip_res_sec
//          {
//          NIC tib_nic_res_sec
//          }
//      }
//  }


group Tibco_infra (
 SystemList = { system01 = 0, system02 = 1 }
 Parallel = 1
 )

 CFSMount tibcfs_mnt (
  MountPoint = "/home/tibco"
  BlockDevice = "/dev/vx/dsk/tibco_install_dg/tibco_install_vol"
  )

 CVMVolDg tibcfs_voldg (
  CVMDiskGroup = tibco_install_dg
  CVMVolume = { tibco_install_vol }
  CVMActivation = sw
  )

 requires group cvm online local firm
 tibcfs_mnt requires tibcfs_voldg


 // resource dependency tree
 //
 // group Tibco_infra
 // {
 // CFSMount tibcfs_mnt
 //      {
 //      CVMVolDg tibcfs_voldg
 //      }
 // }


group cvm (
 SystemList = { system01 = 0, system02 = 1 }
 AutoFailOver = 0
```

```
Parallel = 1
AutoStartList = { system01, system02 }
)

CFSfsckd vxfsckd (
 )

CVMCluster cvm_clus (
 CVMClustName = system01_075
 CVMNodeId = { system01 = 0, system02 = 1 }
 CVMTransport = gab
 CVMTimeout = 200
 )

CVMVxconfigd cvm_vxconfigd (
 CVMVxconfigdArgs = { syslog }
 )

cvm_clus requires cvm_vxconfigd
vxfsckd requires cvm_clus


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

Sample resource dependency in a CFS based fault tolerant configuration for Tibco EMS server

This section includes resource dependencies of Tibco service group in a CFS based fault tolerant configuration.

The following diagram depicts the sample resource dependency of primary Tibco service group TibcoSG_Pri in a CFS based fault tolerant configuration.

Figure A-3   Sample resource dependency of primary Tibco service group in a CFS based fault tolerant configuration
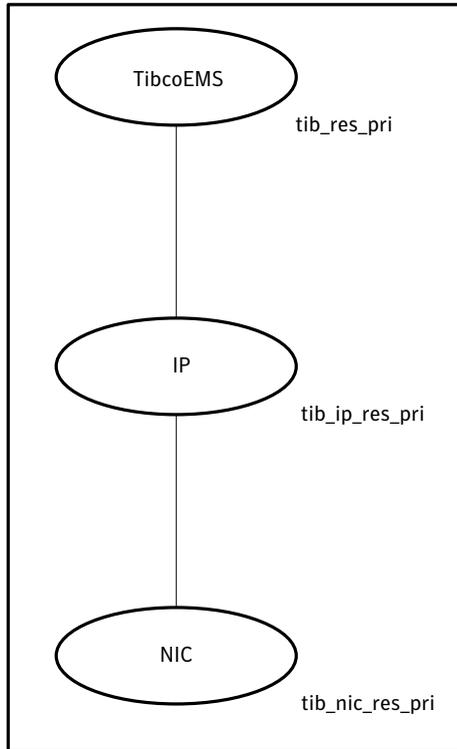
The following diagram depicts the sample resource dependency of secondary Tibco service group TibcoSG_Sec in a CFS based fault tolerant configuration.

Figure A-4   Sample resource dependency of secondary Tibco service group in a CFS based fault tolerant configuration
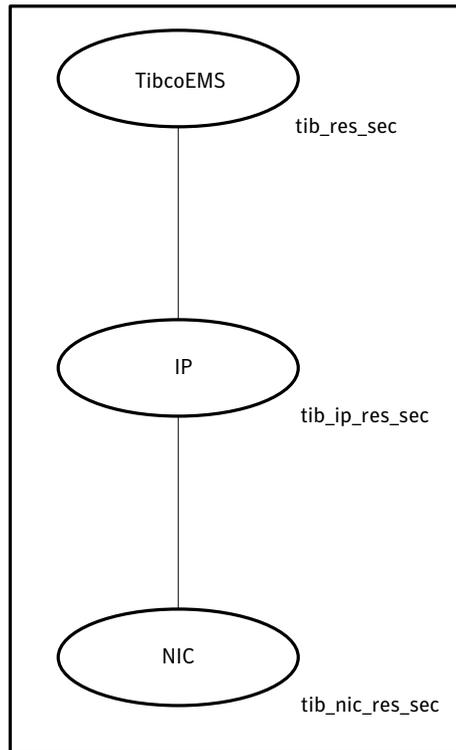
The following diagram depicts the sample resource dependency of Tibco infrastructure service group TibcoSG_infra in a CFS based fault tolerant configuration.

Figure A-5   Sample resource dependency of Tibco infrastructure service group in a CFS based fault tolerant configuration
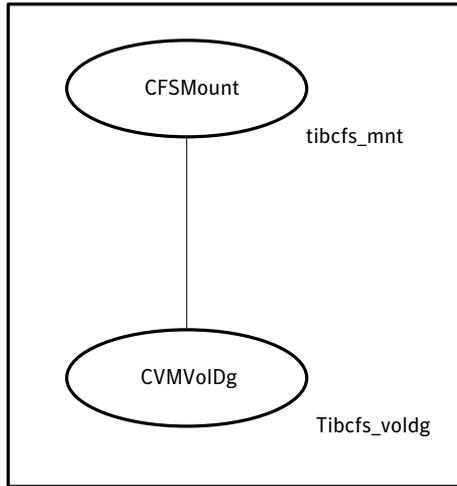
The following diagram depicts the sample resource dependency of cvm service group in a CFS based fault tolerant configuration.

Figure A-6   Sample resource dependency of cvm service group in a CFS based fault tolerant configuration
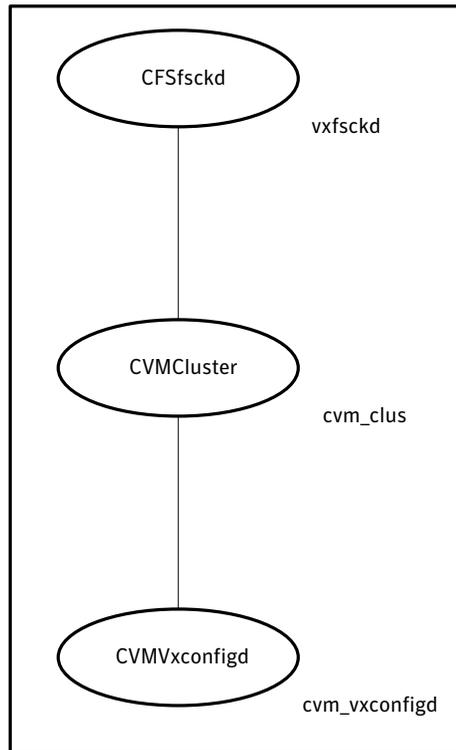
Service group dependency in a CFS based fault tolerant configuration for Tibco EMS Server

This section includes service groups that show the group dependency in a CFS based fault tolerant configuration for Tibco EMS Server.

Figure A-7   Sample service group dependency

Index