Cluster Server Agent for TIBCO EMS Server Installation and Configuration Guide

AIX, Linux, Solaris

8.0.2

VERITAS

Veritas InfoScale™ Availability Agents

Last updated: 2023-06-30

Legal Notice

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 7   **Troubleshooting the agent for TIBCO EMS Server**

Appendix A   **Sample Configurations** 57

Introducing the agent for TIBCO EMS Server

This chapter includes the following topics:

- About the Cluster Server agent for TIBCO EMS Server

- Supported software

- Features of the agent

- How the agent makes TIBCO EMS Server highly available

- How the agent supports intelligent resource monitoring

- Tibco EMS Server agent functions

About the Cluster Server agent for TIBCO EMS Server

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

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

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

Supported software

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

Features of the agent

The following are the features of the Cluster Server agent for TIBCO EMS Server:

- Support for validation of attributes that are based on the 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.

- Support for intelligent resource monitoring and poll-based monitoring
  The agent supports the Cluster Server Intelligent Monitoring Framework (IMF) feature. IMF allows the agent to register the resources to be monitored with the IMF notification module so as to receive immediate notification of resource state changes without having to periodically poll the resources.
  See "About Intelligent Monitoring Framework" on page 38.

- 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 Cluster Server 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.

How the agent supports intelligent resource monitoring

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

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

For example, the agent for TIBCO EMS Server registers the PIDs of the TIBCO EMS Server processes with the AMF kernel driver.

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

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

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 agent performs the following tasks as part of the online function:

- 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.
  In RHEL7, SLES12, or later supported Linux environments, the agent attempts to start the Tibco EMS server instance through the service file
  vcs-*resourceName*-simple.service using the following command:

  EMSHomeDir/tibemsd -config ConfigFile StartOptions

  The command is always executed in the context of the system slice.

- 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. The agent for TIBCO EMS Server also supports Intelligent Monitoring Framework (IMF) in the first-level check. IMF enables intelligent resource monitoring. The agent for TIBCO EMS Server is IMF-aware and uses the asynchronous monitoring framework (AMF) kernel driver for resource state change notifications. See "How the agent supports intelligent resource monitoring" on page 9. You can use the MonitorFreq key of the IMF attribute to specify the frequency at which the agent invokes the monitor function. See "MonitorFreq" on page 41.

- 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.

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.

Note: For information about the additional functions of the agent for TIBCO EMS Server when IMF is enabled: See "Agent functions for the IMF functionality" on page 39.

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

■ 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 Cluster Server agent must be able to uniquely identify each of the TibcoEMS instance on that system. Each TibcoEMS server has a unique configuration file. The 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 Cluster Server agent kills the processes of a non-responsive or failed TibcoEMS instance in the absence of unique ConfigFile for each TibcoEMS Server instance, the agent may kill processes for more than one TibcoEMS server instance during a clean operation.

InstName	InstType
DVEBMGS00	SAP Application Server - ABAP (Primary)
D01	SAP Application Server - ABAP (Additional)
ASCS02	SAP Central Services - ABAP
J03	SAP Application Server - Java
SCS04	SAP Central Services - Java
ERS05	SAP Enqueue Replication Server
SMDA97	Solution Manager Diagnostics Agent

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 synchronize 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 the main Tibco configuration file specified as part of the 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 Cluster Server agent for TIBCO EMS Server
- About the ACC library
- Installing the ACC library
- Installing the agent in a VCS environment
- Uninstalling the agent in a VCS environment
- Removing the ACC library
- Upgrading the agent for TibcoEMS

Before you install the Cluster Server agent for TIBCO EMS Server

You must install the Cluster Server agent for TIBCO EMS Server on all the systems that will host TIBCO EMS Server service groups.

Before you install the agent for TIBCO EMS Server, ensure that the following prerequisites are met.

- 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 17.

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 18.

- 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
'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/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.

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

5 Install the package. Enter **Yes**, if asked to confirm overwriting of files in the
existing package.

AIX	`# installp -ac -d VRTSacclib.bff VRTSacclib`
Linux	`# rpm -i \` `VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm`
Solaris	`# pkgadd -d VRTSacclib.pkg`

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

Installing the ACC library IPS package on Oracle Solaris 11 systems

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

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

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

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

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

3   Add a file-based repository in the system.

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

4   Install the package.

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

5   Remove the publisher from the system.

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

6   Enable the publishers that were disabled earlier.

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

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

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

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

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

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

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

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

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

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

4 Copy the `VRTSacclib.p5p` package from the `pkgs` directory to the non-global zone (for example, at the `/tmp/install` directory).

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

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

6 Add a file-based repository in the non-global zone.

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

7 Install the package.

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

8 Remove the publisher on the non-global zone.

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

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

```
# svcadm clear svc:/application/pkg/system-repository:default
```

10 Enable the publishers that were disabled earlier.

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

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

Installing the agent in a VCS environment

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 from the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

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

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

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

AIX `cd1/aix/vcs/application/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 tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).

4 Log in as a superuser.

5 Install the package.

AIX `# installp -ac -d`
`VRTSvcstib.rte.bff VRTSvcstib.rte`

Linux `# rpm -ihv \`
`VRTSvcstib-AgentVersion-GA_GENERIC.noarch.rpm`

Solaris `# pkgadd -d . VRTSvcstib`

After installing the agent package, you must import the agent type configuration file.

See "Importing the agent types files in a VCS environment" on page 27.

Installing the agent IPS package on Oracle Solaris 11 systems

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

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

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

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

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

3 Add a file-based repository in the system.

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

4 Install the package.

```
# pkg install --accept VRTSvcstib
```

5 Remove the publisher from the system.

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

6 Enable the publishers that were disabled earlier.

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

Installing agent packages on Solaris brand non-global zones

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

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

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

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

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

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

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

4 Copy the `VRTSvcstib.p5p` package from the `pkgs` directory to the non-global
zone (for example, at the `/tmp/install` directory).

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

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

6 Add a file-based repository in the non-global zone.

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

7 Install the package.

```
# pkg install --accept VRTSvcstib
```

8 Remove the publisher on the non-global zone.

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

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

```
# svcadm clear svc:/application/pkg/system-repository:default
```

10 Enable the publishers that were disabled earlier.

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

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

Installing the agent in a Solaris 10 brand zone

To install the TIBCO EMS Server agent in a Solaris 10 brand zone:

- Ensure that the ACC library package, VRTSacclib, is installed in the non-global zone.
  To install VRTSacclib in the non-global zone, run the following command from the global zone:
  ```
  # pkgadd -R /zones/zone1/root -d VRTSacclib.pkg
  ```

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

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

```
## Executing postinstall script.

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

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

or ## Executing postinstall script.

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

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

Uninstalling 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 running the following command from any node in the cluster:

```
# haconf -makerw
```

3   Remove all TIBCO EMS Server resources from the cluster. Run 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 running 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 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 TIBCO EMS Server from each node in the cluster.

Run the following command to uninstall the agent:

AIX	`# installp -u VRTSvcstib.rte`
Linux	`# rpm -e VRTSvcstib`
Solaris	`# pkgrm VRTSvcstib`
	Note: To uninstall the agent IPS package on a Solaris 11 system, run the following command:
	`# pkg uninstall 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`

Linux `# rpm -e VRTSacclib`

Solaris `# pkgrm VRTSacclib`

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

`# pkg uninstall VRTSacclib`

Upgrading the agent for TibcoEMS

Perform the following steps to upgrade the agent with minimal disruption.

To upgrade the agent for TibcoEMS

1 Verify the version of the agent for TibcoEMS.

`# `**`rpm -qi VRTSvcstib | grep Version`**

The output resembles:

`Version : 8.0.0.0`

2 Save the configuration.

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

3 Identify the TibcoEMS resource and service group.

`# `**`hatype -resources TibcoEMS`**

The output resembles:

`tib_res`

`# `**`hares -display tib_res | grep Group`**

The output resembles:

`tib_res Group global tib_grp`

4 Freeze the TibcoEMS service group.

```
# hagrp -freeze tib_grp
```

5 Check if the TibcoEMS agent is running.

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

The output resembles:

```
TibcoEMS Running Yes
```

6 If the agent is running, stop the agent.

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

7 Verify the status of the agent.

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

The output resembles:

```
TibcoEMS Running No
```

8 Upgrade the agent.

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

9 Start the agent.

```
# haagent -start TibcoEMS -sys hostname
```

The output resembles:

```
VCS NOTICE V-16-1-10001 Please look for messages in the log file
```

10 Verify the status of the agent.

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

The output resembles:

```
TibcoEMS Running Yes
```

11 Unfreeze the TibcoEMS service group.

```
# hagrp -unfreeze tib_grp
```

Configuring the agent for TIBCO EMS Server

This chapter includes the following topics:

- About configuring the Cluster Server agent for TIBCO EMS Server
- Importing the agent types files in a VCS environment
- Tibco EMS Server agent attributes

About configuring the Cluster Server agent for TIBCO EMS Server

After installing the Cluster Server agent for TIBCO EMS Server, 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 TIBCO EMS Server resources.

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

See "About sample configurations for the agents for TIBCO EMS Server" on page 57.

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. 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	■ AIX ■ Linux ■ Solaris	`/etc/VRTSvcs/conf/sample_TibcoEMS/` `TibcoEMSTypes.cf`
VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/TibcoEMS/` `TibcoEMSTypes50.cf`
VCS 5.1 or later	Solaris SPARC	`/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 *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_TibcoEMS/
	Linux	TibcoEMSTypes.cf
	Solaris	
VCS 5.0	Solaris SPARC	/etc/VRTSagents/ha/conf/TibcoEMS/
		TibcoEMSTypes50.cf
VCS 5.1 or later	Solaris SPARC	/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 *Cluster Server Administrator's Guide*.

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

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

1 Start the Cluster Manager (Java Console) and connect to the cluster on which the agent is installed.

2 Click **File > Import Types**.

3 In the **Import Types** dialog box, select the following file:

VCS 4.x	■ AIX	/etc/VRTSvcs/conf/sample_TibcoEMS/
	■ Linux	TibcoEMSTypes.cf
	■ Solaris	
VCS 5.x or later	■ AIX	/etc/VRTSagents/ha/conf/TibcoEMS/
	■ Linux	TibcoEMSTypes.cf

VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/TibcoEMS/`
		`TibcoEMSTypes50.cf`
VCS 5.1 or later	Solaris SPARC	`/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 CLI, refer to the *Cluster Server Administrator's Guide*.

Importing the agent types file using the CLI

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

1 If VCS is running, run the `/etc/VRTSagents/ha/conf/TibcoEMS/TibcoEMSTypes.cmd` file from the command line.

2 If VCS is not running, perform the following steps:

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

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

VCS 4.x	■ AIX	`/etc/VRTSvcs/conf/sample_TibcoEMS/`
	■ Linux	`TibcoEMSTypes.cf`
	■ Solaris	
VCS 5.x or later	■ AIX	`/etc/VRTSagents/ha/conf/TibcoEMS/`
	■ Linux	`TibcoEMSTypes.cf`
VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/TibcoEMS/`
		`TibcoEMSTypes50.cf`
VCS 5.1 or later	Solaris SPARC	`/etc/VRTSagents/ha/conf/TibcoEMS/`
		`TibcoEMSTypes51.cf`

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

3. Start HAD.

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 any one of the following: ■ The full path of the directory where the tibemsd binary file is located. In this case, the agent uses the tibems binary by default. ■ The full path of the Tibco EMS binary — tibemsd or tibemsd64. Type and Dimension: string-scalar Default: No default value Example 1: `/home/tibco/ems/bin` Example 2: `/home/tibco64/tibco/ems/8.0/bin/tibemsd64`
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-scalar Default: No default value Example: `/home/tibco/ems/bin/tibemsd.conf`

Table 4-1   Required attributes *(continued)*

Required attributes	Description
User	UNIX user name that the VHA agent will use to 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-scalar
	Default: No default value
	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-scalar
	Default: INFO
	Example: TRACE
	Note: The use of the ResLogLevel attribute is deprecated from VCS version 6.2 onwards. You must use the LogDbg attribute instead of the ResLogLevel attribute to enable debug logs for the ACCLib-based agents, when the ACCLib version is 6.2.0.0 or later. The agent captures the first failure data of the unexpected events and automatically logs debug messages in their respective agent log files.

Table 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 the tibemsadmin utility. Type and Dimension: string-scalar Default: No default value 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-scalar Default: No default value 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 the tibemsadmin utility. The value of this attribute need to be encrypted using VCS provided $VCSHOME/bin/vcsencrypt utility Type and Dimension: string-scalar Default: No default value Example: HTIvKTITNnINjNKnL

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

Table 4-2   Optional attributes

Optional attributes	Description
LogDbg	For ACCLib-based agents, you must use the LogDbg resource type attribute to enable the debug logs when the ACCLib version is 6.2.0.0 or later and the VCS version is 6.2 or later.
	Set the LogDbg attribute to DBG_5 to enable debug logs for ACCLIB based agent. By default, setting the LogDbg attribute to DBG_5 enables debug logs for all TibcoEMS resources in the cluster. If debug logs must be enabled for a specific TibcoEMS resource, override the LogDbg attribute.
	Type and dimension: string-keylist
	Default: {} (none)
	For more information on how to use the LogDbg attribute, refer to the *Cluster Server Administrator's Guide*.
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-scalar
	Default: 0
	Example: 1

Table 4-2 Optional attributes *(continued)*

Optional attributes	Description
LevelTwoMonitorFreq	Specifies the frequency at which the agent for this resource type must perform second-level or detailed monitoring. You can also override the value of this attribute at the resource level. The value indicates the number of monitor cycles after which the agent will monitor the Tibco EMS Server in detail.
	For example, the value 5 indicates that the agent will monitor the Tibco EMS Server in detail after every five online monitor intervals.
	Note: This attribute is applicable to VCS version 5.1 SP1 or later with TIBCO EMS Server agent version 5.1.2.0 or later. If the VCS version is earlier than VCS 5.1 SP1 and the TIBCO EMS Server agent version is earlier than 5.1.2.0, the SecondLevelMonitor attribute should be used.
	If you upgraded the VCS version to VCS 5.1 SP1 or later and the TIBCO EMS Server agent version to 5.1.2.0 (or later), and if you had enabled detail monitoring in the previous version, then do the following:
	■ Set the value of the LevelTwoMonitorFreq attribute to the same value as that of the SecondLevelMonitor attribute.
	Type and dimension: integer-scalar
	Default: 0

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. Veritas 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. Type and Dimension: string-scalar Default: No default value 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 the **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-scalar Default: No default value 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. Veritas recommends to store the file on a shared disk.
	The shell environments supported are: ksh, sh, and csh.
	Type and Dimension: string-scalar
	Default: No default value
	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 the tibemsadmin utility. By default, the agent specifies the -server, -user, and -password parameters, hence these parameter should not be specified as part of TibemsadminOptions again.
	Type and Dimension: association-scalar
	Default: No default value
	Example: `-ssl_trusted /home/tibco/ssl/certificate1 -ssl_identity /home/tibco/ssl/certificate2 -ssl_trace`

Note: For information about the additional attributes of the agent for TIBCO EMS Server when IMF is enabled: See "Attributes that enable IMF" on page 40.

Chapter

5

Enabling the agent for TIBCO EMS Server to support IMF

This chapter includes the following topics:

- About Intelligent Monitoring Framework

- Agent functions for the IMF functionality

- Attributes that enable IMF

- Before you enable the agent to support IMF

- Enabling the agent to support IMF

- Disabling intelligent resource monitoring

- Sample IMF configurations

About Intelligent Monitoring Framework

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

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

You can enable the TIBCO EMS Server agent for IMF, provided the following software versions are installed:

- Cluster Server (VCS) 5.1 SP1 or later

- Cluster Server agent for TIBCO EMS Server version 5.1.0.0 or later

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

Benefits of IMF

IMF offers the following benefits:

- Performance
  Enhances performance by reducing the monitoring of each resource at a default of 60 seconds for online resources, and 300 seconds for offline resources. IMF enables the agent to monitor a large number of resources with a minimal effect on performance.

- Faster detection
  Asynchronous notifications would detect a change in the resource state as soon as it happens. Immediate notification enables the agent to take action at the time of the event.

Agent functions for the IMF functionality

If the TIBCO EMS Server agent is enabled for IMF support, the agent supports the following functions, in addition to the functions mentioned in the TIBCO EMS Server agent functions topic.

See "Tibco EMS Server agent functions" on page 10.

imf_init

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

imf_getnotification

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

imf_register

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

Attributes that enable IMF

If the agent for TIBCO EMS Server is enabled for IMF support, the agent uses type-level attributes in addition to the agent-specific attributes.

See "Tibco EMS Server agent attributes" on page 31.

IMF

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

This attribute includes the following keys:

Mode

Define this attribute to enable or disable intelligent resource monitoring.

The valid values are as follows:

- 0—Does not perform intelligent resource monitoring

- 1—Performs intelligent resource monitoring for offline resources and performs poll-based monitoring for online resources

- 2—Performs intelligent resource monitoring for online resources and performs poll-based monitoring for offline resources

- 3—Performs intelligent resource monitoring for both online and for offline resources.

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

Type and dimension: integer-association

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

MonitorFreq

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

Default: 1

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

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

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

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

RegisterRetryLimit

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

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

Default: 3.

IMFRegList

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

Type and dimension: string-vector

Default: No default value

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

Before you enable the agent to support IMF

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

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

Enabling the agent to support IMF

In order to enable the TIBCO EMS Server agent to support IMF, you must make the following configuration changes to the attributes of the agent:

- AgentFile: Set the AgentFile attribute to **Script51Agent** or **Script60Agent** as appropriate for your agent version

- IMF Mode: Set the IMF Mode attribute to **2**

- IMFRegList: Update the IMFRegList attribute

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

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

```
# cd /opt/VRTSagents/ha/bin/<resourceType>
```

```
# ln -s /opt/VRTSamf/imf/imf_getnotification imf_getnotification
```

```
# ln -s /opt/VRTSagents/ha/bin/<resourceType>/monitor imf_register
```

If VCS is in a running state

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

1   Make the VCS configuration writable.

```
# haconf -makerw
```

2   Run the following command to update the AgentFile attribute.

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

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

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

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

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

4   Run the following command to update the IMF attribute.

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

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

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

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

5   Run the following command to update the IMFRegList attribute:

```
# hatype -modify TibcoEMS IMFRegList EMSHomeDir ConfigFile User
```

6   Save the VCS configuration.

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

7   If the agent is running, restart the agent.

For information on the commands you can use to restart the agent, see Restarting the agent.

Restarting the agent

To restart the agent:

1 Run the following command to stop the agent forcefully:

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

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

---

2 Run the following command to start the agent:

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

If VCS is not in a running state

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

1 Update the AgentFile attribute.

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

2 Update the IMF attribute.

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

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

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

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

3 Update the IMFRegList attribute.

```
static str IMFRegList[] = { EMSHomeDir, ConfigFile, User }
```

Disabling intelligent resource monitoring

To disable intelligent resource monitoring

1   Make the VCS configuration writable.

```
# haconf -makerw
```

2   To disable intelligent resource monitoring for all the resources of a certain type, run the following command:

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

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

```
# hares -override resource_name IMF
```

```
# hares -modify resource_name IMF -update Mode 0
```

4   Save the VCS configuration.

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

Sample IMF configurations

An example of a type definition file for a TIBCO EMS Server agent that is IMF-enabled is as follows.

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

AgentFile	/opt/VRTSvcs/bin/Script51Agent
IMF{}	{ Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
IMFRegList[]	{ EMSHomeDir, ConfigFile, User }
LevelTwoMonitorFreq	25

```
type TibcoEMS (
    static int IMF{} = { Mode=2, MonitorFreq=3, RegisterRetryLimit=3 }
    static str IMFRegList[] = { EMSHomeDir, ConfigFile, User }
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/TibcoEMS"
    static str AgentFile = "/opt/VRTSvcs/bin/Script51Agent"
    static int LevelTwoMonitorFreq = 2
    static keylist LogDbg = { DBG_1, DBG_2, DBG_3, DBG_4, DBG_5 }
    static str ArgList[] = { ResLogLevel, State, IState, User, EnvFile,
    EMSHomeDir, ConfigFile, TibEmsServerUrl, SecondLevelMonitor, TibUser,
```

```
    TibPassword, MonitorProgram, StartOptions, TibemsadminOptions }
    static boolean AEPTimeout = 1
    str ResLogLevel = INFO
    str User
    str EnvFile
    str EMSHomeDir
    str ConfigFile
    str TibEmsServerUrl
    int SecondLevelMonitor
    str TibUser
    str TibPassword
    str MonitorProgram
    str StartOptions
    str TibemsadminOptions
)
```

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

```
TibcoEMS tibco_res (
    Critical = 0
    ResLogLevel = TRACE
    User = tibco
    EMSHomeDir = "/home/tibco/tibco/ems/6.3/bin"
    ConfigFile = "/home/tibco/TIBCO_HOME/tibco/
    cfgmgmt/ems/data/tibemsd.conf"
    TibEmsServerUrl = "tcp://tuxedovm1:7222"
    TibUser = admin
    TibPassword = gumSjuJ
)
```

Configuring the service groups for TIBCO EMS Server using the CLI

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

- Creating service groups for TIBCO EMS Server under Solaris non-global zones

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 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 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.

- Verify that the Cluster Server agent for TIBCO EMS Server is installed on all nodes in the cluster.

- Verify that the type definition for the Cluster Server 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 at http://www.veritas.com/docs/000034279.

See "Sample service group configuration for Tibco EMS server in Failover setup" on page 60.

See "Sample service group configuration for Tibco EMS server in CFS based fault tolerant setup" on page 66.

Creating service groups for TIBCO EMS Server under Solaris non-global zones

To configure zones on each cluster node

1   Set up the non-global zone configuration.

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

For example:

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

2   Verify the non-global zone configuration.

```
hazoneverify servicegroup_name
```

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

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

5   Make sure that the application is identical on all nodes. If you update the application configuration on one node, apply the same updates to all nodes.

6   Configure the service groups for TIBCO EMS Server.

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

- Troubleshooting the configuration for IMF

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

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

Meeting prerequisites

Before installing the agent for TIBCO EMS Server, ensure that the following prerequisites are met.

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

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 described 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

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. Veritas recommends that you localize the ResLogLevel attribute for a particular resource.

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

To localize ResLogLevel attribute for a resource

1 Identify the resource for which you want to enable detailed logging.

2 Localize the ResLogLevel attribute for the identified resource:

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

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

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

4 Note the time before you begin to operate the identified resource.

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

6 Note the time when the problem is reproduced.

7 Set the ResLogLevel attribute back to INFO for the identified resource:

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

8   Save the configuration changes.

```
# haconf -dump
```

9   Review the contents of the log file.

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

You can also contact Veritas support for more help.

To enable debug logs for all resources of type TibcoEMS

Enable the debug log.

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

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.

Troubleshooting the configuration for IMF

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

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

  - AgentFile

  - IMF

  - IMFRegList
    If IMFRegList is not configured correctly, the TIBCO EMS Server resources that have been registered for IMF get unregistered every time the monitor function is run.

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

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

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

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

  - Traditional—Poll-based resource monitoring

  - IMF—Intelligent resource monitoring

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

  The amfstat command shows the PIDs monitored by the Tibco EMS Server agent.

  ```
  [root@tuxedovm1 ~]#amfstat
  AMF Status Report

  Registered Reapers (3):
  =======================
  RID     PID       MONITOR   TRIGG   REAPER
  14      8475      1         0       TibcoEMS

  Process ONLINE Monitors (1):
  ============================
  RID     R_RID   PID       GROUP
  20      14      3187      tibco_res
  ```

  The agent identifies the process for a Tibco EMS Server instance by applying pattern matching on the output of the ps -ef command. The pattern is as follows:

  ```
  tibco     3187  3169  0 16:27 pts/2
  00:00:00 /home/tibco/tibco/ems/6.3/bin/tibemsd -config
  /home/tibco/TIBCO_HOME/tibco/cfgmgmt/ems/data/tibemsd.conf
  ```

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

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

■ Run the following command to view the content of the AMF in-memory trace buffer.

```
# amfconfig -p dbglog
```

Known issues

This release of the agent for TIBCO EMS Server has the following known issues:

Problem

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

Description

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

This results in an error message from the engine log.

Workaround

It is safe to ignore this error message.

Sample Configurations

This appendix includes the following topics:

About sample configurations for the agents 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 agents for TIBCO EMS Server. For more information

about these resource types, refer to the *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 and later

```
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 resource configuration

This section lists the sample resource configuration when the ACCLib installed is 6.2.0.0 and the agent installed is 6.2. This configuration is also applicable for later versions of the ACC library and the agent.

```
tibco_grp (
            SystemList = { system01 = 0, system02 = 1 }
```

```
            )

TibcoEMS tibco_res (
                User = tibco
                EMSHomeDir = "/home/tibco/ems/8.1/bin"
                ConfigFile = "/home/tibco/TIB/tibco/cfgmgmt/ems/data/tibemsd.co
                TibEmsServerUrl = "tcp://system01:7222"
                TibPassword = GUMsJUjKKmHMiMJ
                LogDbg = { DBG_5 }
            )
```

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
```
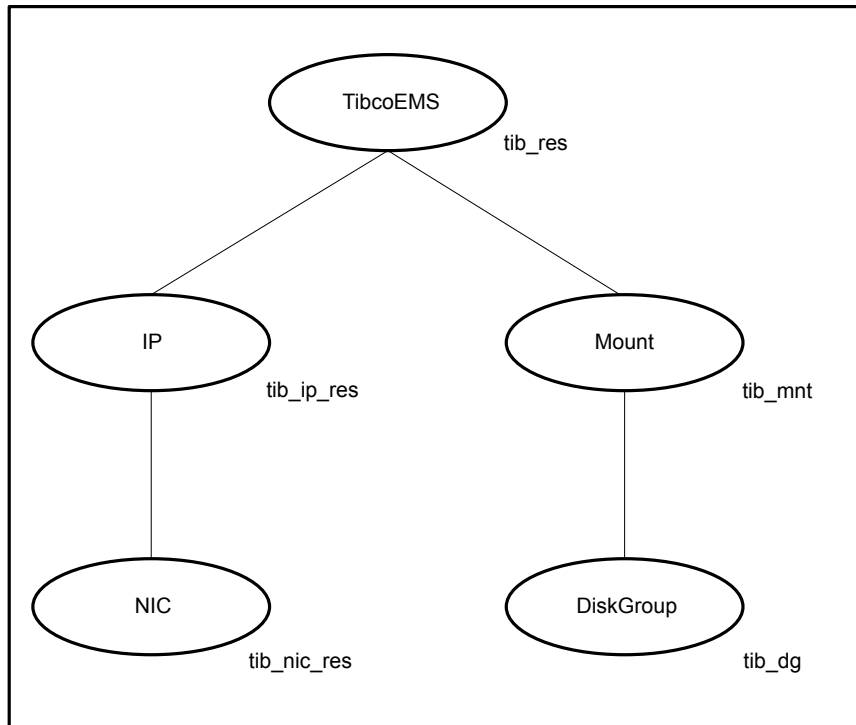
```
// resource dependency tree
//
// group tibcoZoneSG
// {
// TibcoEMS tib_zone_res
//      {
//      IP tib_ip
//         {
//         Zone zoneres
//            {
//            Mount tib_data_mnt
//               {
//               DiskGroup tib_data_dg
//               }
//            NIC tib_nic
//            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.
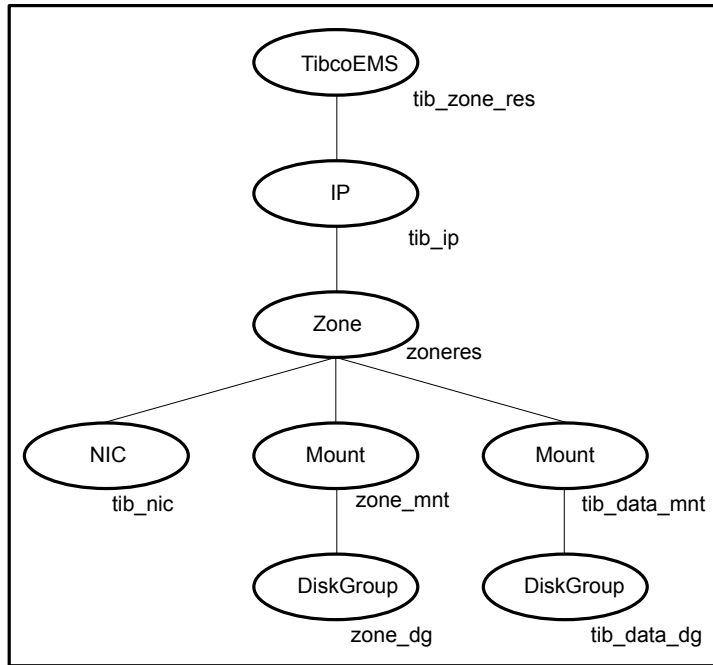
Figure A-1   Sample resource dependency in failover configuration

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.

Figure A-3 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
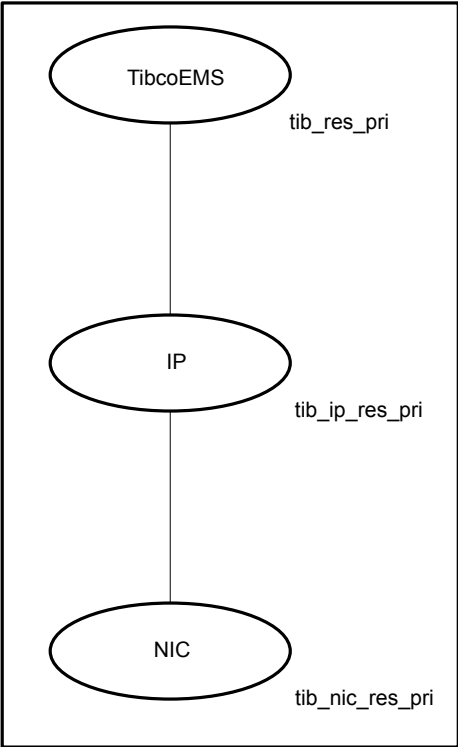

Figure A-4 depicts the sample resource dependency of secondary Tibco service group TibcoSG_Sec in a CFS based fault tolerant configuration.

Figure A-4   Sample resource dependency of secondary Tibco service group in a CFS based fault tolerant configuration
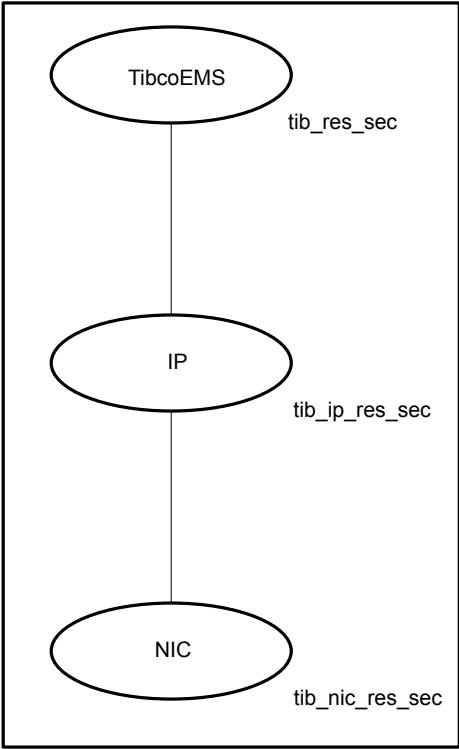

Figure A-5The following diagram depicts the sample resource dependency of Tibco infrastructure service group TibcoSG_infra in a CFS based fault tolerant configuration.

Figure A-5   Sample resource dependency of Tibco infrastructure service group in a CFS based fault tolerant configuration
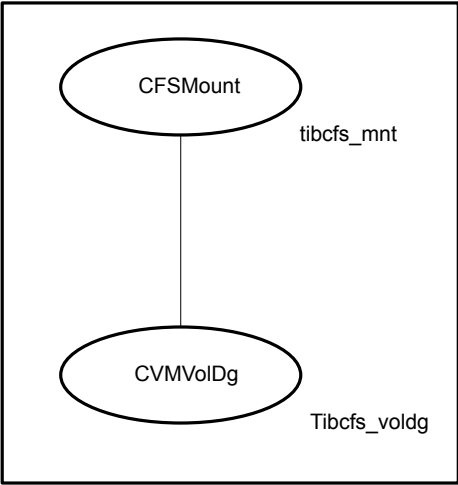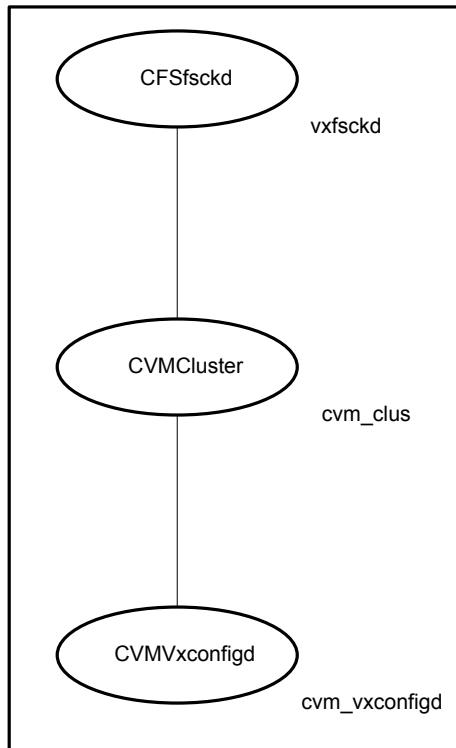

Figure A-6The following diagram depicts the sample resource dependency of cvm service group in a CFS based fault tolerant configuration.

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