Veritas™ High Availability Agent for BEA Tuxedo Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

5.0

December 2008

Veritas High Availability Agent for BEA Tuxedo 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: 5.0.2.0

Document version: 5.0.2

Legal Notice

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

http://www.symantec.com

Technical Support

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

Symantec's maintenance 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 Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers automatic software upgrade protection
- Global support that is available 24 hours a day, 7 days a week
- Advanced features, including Account Management Services

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

www.symantec.com/techsupp/

Contacting Technical Support

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

www.symantec.com/business/support/assistance_care.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/techsupp/

Customer service

Customer service information is available at the following URL:

www.symantec.com/techsupp/

Customer Service is available to assist with 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 maintenance 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

Documentation feedback

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

Maintenance agreement resources

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

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

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Symantec Early Warning Solutions	These solutions provide early warning of cyber attacks, comprehensive threat analysis, and countermeasures to prevent attacks before they occur.
Managed Security Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.
Consulting Services	Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.
Educational Services	Educational Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about Enterprise services, please visit our Web site at the following URL:

www.symantec.com

Select your country or language from the site index.

Contents

Technical Support 4

Chapter 1 Introducing the Veritas High Availability Agent for BEA Tuxedo 11

About the Veritas agent for BEA Tuxedo 11
Features of the Veritas agent for BEA Tuxedo 12
What's new in this agent 12
Supported software 12
About BEA Tuxedo 13
About the BEA Tuxedo server 13
BEA Tuxedo agent operations 14
Online 14
Offline 15
Monitor 16
Clean 18

Chapter 2 Installing, upgrading, and removing the agent for BEA Tuxedo 21

Before you install the Veritas agent for BEA Tuxedo 21
Prerequisites for installing the agent to support Solaris zones 22
About ACC Library 22
Installing the ACC library 22
Installing the agent in a VCS environment 23
Removing the agent in a VCS environment 24
Removing the ACC library 25
Upgrading the agent for BEA Tuxedo 25

Chapter 3 Preparing to configure the agent for BEA Tuxedo 27

About configuring the Veritas agent for BEA Tuxedo 27
Importing the agent types files in a VCS environment 27
Agent attributes 28
Executing a customized monitoring program 37

Configuring BEA Tuxedo resources for Solaris zones support 37

Chapter 4 Clustering BEA Tuxedo 39

Overview of BEA Tuxedo 39
Installing BEA Tuxedo software and applications on shared disks 43
Directory structure recommendation 44
Basic steps to cluster a BEA Tuxedo machine 45
Configuring BEA Tuxedo using the ULGPFX attribute 50
Additional requirements for multiple machine domains (MP model) 52
High availability scenarios supported 53
About service group dependencies with the MP model 55
Configuring Remote Shell or OpenSSH 55
Tuning MonitorTimeout value 56
Configuring ToleranceLimit attribute 57
Clustering BEA Tuxedo listeners 57
BEA Tuxedo entities in a clustered environment 60
Virtualizing BEA Tuxedo 61

Chapter 5 Clustering PeopleSoft Enterprise Servers 63

Overview of PeopleSoft 63
Creating the BEA Tuxedo environment file 64
Method 1: Capturing the environment using the ps command 64
Method 2: Capture the environment using the psadmin utility 68
Testing the Tuxedo environment file 76
Sample resource definition 79
Obtaining values for attributes MachineName and TuxDomain 80
Additional PeopleSoft configuration notes 81
Registering long PeopleSoft service names 82

Chapter 6 Troubleshooting the agent for BEA Tuxedo 85

Using correct software and operating system versions 85
Meeting prerequisites 85
Configuring BEA Tuxedo resources 86
Verifying virtualization 86
Starting the BEA Tuxedo instance outside a cluster 86
Reviewing error log files 87
Using trace level logging 88

Appendix A Sample Configurations 91

About sample configurations for the agent for BEA Tuxedo 91
Sample configuration file (SHM model) 91
Sample configuration file for PSFT (SHM mode) 93
Sample configuration file (MP model) 95
Sample agent type definition (VCS 4.x) 102
Sample agent type definition (VCS 5.0) 103
Sample agent type definition (VCS 5.0 under Solaris 10 zone) 104
Sample configuration for Solaris zone support 104

Index 107

Chapter 1

Introducing the Veritas High Availability Agent for BEA Tuxedo

This chapter includes the following topics:

- About the Veritas agent for BEA Tuxedo
- Features of the Veritas agent for BEA Tuxedo
- What's new in this agent
- Supported software
- About BEA Tuxedo
- About the BEA Tuxedo server
- BEA Tuxedo agent operations

About the Veritas agent for BEA Tuxedo

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 BEA Tuxedo manages and provides high availability for all the BEA Tuxedo machines in a clustered environment. The agent can bring a specific BEA Tuxedo machine instance online and can monitor the state of the BEA Tuxedo machine. The agent can also detect failures and can shut down the instance in case of a failure.

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

http://seer.entsupport.symantec.com/docs/282004.htm

Features of the Veritas agent for BEA Tuxedo

The following are the features of the Veritas agent for BEA Tuxedo:

- 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.

What's new in this agent

The enhancements in this release of Veritas agent for BEA Tuxedo are as follows:

- Added support for Solaris 10 zones.

Supported software

The Veritas agent for BEA Tuxedo supports the following software versions:

Veritas Cluster Server	4.0, 4.1, 5.0
ACC Library	5.1.3.0 and later

Operating systems	■ AIX 5.3 on pSeries ■ HP-UX 11i v2 on PA-RISC ■ Red Hat Enterprise Linux 4.0, 5.0 ■ Solaris 8, 9, 10 on SPARC
BEA Tuxedo	8.0, 8.1, 9.1, 10.0
PeopleSoft	PeopleTools 8.44, 8.45, 8.46, 8.47, 8.48, 8.49

About BEA Tuxedo

The BEA Tuxedo system is a middleware product that distributes applications across multiple platforms, databases, and operating systems using message-based communications and, if desired, distributed transaction processing.

The BEA Tuxedo system provides the following benefits:

- An industry standard for the creation and central administration of distributed online transaction applications in a heterogeneous client/server environment.
- Ease of use for application developers, who do not need to know details about server locations, routing, or platforms that are used. These aspects of a program are transparent in a BEA Tuxedo application.
- The fundamental underpinnings for creating, managing, and maintaining reliable, high performance, easily managed distributed systems.

About the BEA Tuxedo server

A BEA Tuxedo server is a process that oversees a set of services, dispatching them automatically for clients that request them. A service, in turn, is a function within a server program that performs a particular task needed by a business. A bank, for example, might have one service that accepts deposits and another that reports account balances. A server at this bank might receive requests from clients for both services. The server is responsible for dispatching each request to the appropriate service.

Service functions implement business logic through calls to database interfaces such as SQL and, possibly, calls to the ATMI to access additional services, queues, and other resources. The servers on which these services reside then reply to the clients or send the client requests to a new service.

BEA Tuxedo provides the framework, or middleware, for building scalable multi-tier client/server applications in heterogeneous (dissimilar), distributed environments that extend from the Web to the Enterprise. Using BEA Tuxedo,

users can develop, manage, and deploy distributed applications independently of the underlying hardware, operating system, network, and database environment.

Middleware services provide a more functional set of application programming interfaces (API) than the operating system and network services. The main purpose of middleware services is to help solve application connectivity and interoperability problems.

BEA Tuxedo agent operations

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

Online

The online operation is responsible for starting a single BEA Tuxedo machine instance on a node in the cluster.

It performs the following tasks in the given sequence:

- Identifies and removes system processes or IPC resources that may remain from a previous incomplete shutdown of the target BEA Tuxedo machine instance.
- Boots the stand-alone BEA Tuxedo machine, if the MachineType is Single, by first switching to the user specified by the attribute User, setting up the shell environment by sourcing the file specified by the attribute EnvFile, and executing tmboot for the online machine instance.
- Attempts to locate the active master machine and performs the following steps if the MachineType is not Single (that is, application model is Multiple Machine Domain or MP):
  - If an active master machine is located, the online operation cleans the domain bulletin board of stale entries that may remain for the machine to be booted. This is necessary if the machine to be started is flagged as partitioned in the active master's bulletin board. The operation cleans the bulletin board by connecting to the active master machine with tmadmin and issuing the pclean sub-command for the machine to be booted.
  - If MachineType is Master or Backup and the active master machine is not detected, the online operation starts the BEA Tuxedo machine as the active master by executing tmboot locally with the options specified in TmbootOptMaster.

- If an active master machine is detected, the online operation starts the BEA Tuxedo machine by remotely executing tmboot on the active master machine and specifying the non-master machine to be booted. Remote execution of tmadmin is implemented with either rsh or ssh.

Offline

The offline operation is responsible for stopping a single BEA Tuxedo machine instance on a node in the cluster. The steps to take a BEA Tuxedo machine instance offline depend on its MachineType.

If MachineType is Single, the offline operation performs the following tasks in the given sequence:

- Switches to the user specified in the attribute User, sources the file specified in the attribute EnvFile, and executes tmshutdown to shut down the machine.
- Identifies and removes BEA Tuxedo server processes and IPC resources that may remain after tmshutdown exits.

If MachineType is not Single (that is, the application model is MP) and the BEA Tuxedo machine to be taken offline is not the active master machine, the offline operation performs the following tasks in the given sequence:

- Remotely executes tmshutdown on the active master machine with the options specified by TmshutdownOptNonMaster using the remote execution program specified by the attribute RemoteShell. Remote execution is performed in the context of the active master machine's user account (User) and by using the active master machine's environment file (EnvFile).
- Identifies and removes BEA Tuxedo server processes and IPC resources that may remain after the remote tmshutdown completes.
- Attempts to clean the domain bulletin board of stale entries that may remain from the machine that was shut down. The agent accomplishes this by connecting to the active master machine with tmadmin and issuing the pclean sub-command for the target machine.

If MachineType is not Single and the BEA Tuxedo machine to be taken offline is detected as the active master machine, the offline entry point performs the following tasks in the given sequence:

- Attempts to locate the Tuxedo machine acting as the backup, and if successful, migrates the DBBL to make it the active master.
- Shuts down this instance as a backup machine, following the same approach as that for a machine that is not the active master (explained above).

- Attempts to stop the detected instance as the active master with the options specified by TmshutdownOptMaster, if the backup machine in an online state cannot be located.
- Identifies and removes BEA Tuxedo server processes and IPC resources that may remain after tmshutdown exits.

The offline operation exits either after it detects a new active master in the domain, or after the timeout period specified in the OfflineTimeout attribute expires.

Monitor

The monitor operation is responsible for determining the state of a single BEA Tuxedo machine instance on a particular node in the cluster.

The state options are as follows:

Online	Machine is running and responding to service requests
Offline	Machine is not running or responding to service requests
Unknown	Machine is in neither online nor offline state

The monitor operation determines the server's state by performing the following tasks:

- Searches for the target BEA Tuxedo server BBL process and for the optional processes specified by attribute TuxServers. The search is constrained to those processes owned by the account specified in the attribute User. If the processes are found on the target system, the resource is online and the entry point continues to the next step. If they don't exist, the entry point exits and reports the resource as offline.
- If second-level monitoring is enabled (SecondLevelMonitor >= 1), the monitor entry point performs a deeper state check by connecting to the BEA Tuxedo machine's bulletin board using tmadmin. Once connected, the entry point issues the sub-commands printserver (psr) and printservice (psc). The entry point then parses the output from the sub-commands, searching for the BBL process and values specified through attributes TuxServers and TuxServices. If the servers and services are found in the output, the resource is considered online and the entry point continues to the next step. If they are not found, the entry point exits and reports the resource as offline.
- If a valid executable is specified by the attribute MonitorProgram, it is executed in this step. If the return code from the executable is either 0 or 110, the resource is considered online. If the return code is either 1 or 100, the resource

is considered offline and the entry point exits. Note that MonitorProgram is executed regardless of whether second-level monitor is enabled or disabled.

- If the resource MachineType is Single (application model is SHM), monitoring is complete and the entry point exits. If MachineType is not Single, the remaining steps are performed.
- Confirms that the active master (DBBL) for the domain is hosted by either the master or backup machine. The entry point constrains its search for the active master to the VCS resources where MachineType is Master or Backup. To avoid a premature resource fault, if the entry point cannot detect the active master, it will continue to probe for the active master multiple times until the end of the monitor period (that is, the time frame specified by the attribute MonitorTimeout). Thus, the longer the MonitorTimeout period, the longer the monitor entry point waits to detect the active master.
- If an active master is detected and the BEA Tuxedo machine being monitored is part of the domain (that is, it is not partitioned), the machine is considered online, the monitoring process is complete, and the entry point exits.
- If an active master is not detected and the BEA Tuxedo machine being monitored is defined as MachineType Master or Backup, the monitor entry point infers that the machine hosting the active master (DBBL) has faulted and initiates a master migration process on the machine being monitored. The entry point accomplishes this by executing tmadmin and connecting to the machine being monitored. Once connected, it issues the sub-command "master -yes". If the migration is successful, the entry point exits and monitoring is complete. If the migration fails, the entry point exits and reports the resource as offline.
- If an active master is not detected and the BEA Tuxedo machine being monitored is defined as MachineType NonMaster, the monitor entry point continues to search for the active master until the monitor entry point reaches the end of its execution window, which is governed by the value in the attribute MonitorTimeout. Continuing to search for the active master until the end of the monitor period provides ample time for a master migration process to complete. If an active master is still not found by the end of the monitor period, the entry point infers that the entire domain has faulted and declares the resource as offline.

Depending upon the MonitorProgram attribute, the monitor operation can perform a customized check using a user-supplied monitoring utility.

For details about executing a custom monitor program:

See "Executing a customized monitoring program" on page 37.

Clean

The clean operation is responsible for removing BEA Tuxedo processes and IPC resources that may remain after an ungraceful or incomplete shutdown of a BEA Tuxedo machine instance.

The operation cleans the system by performing the following tasks:

- Identifies and removes remaining processes pertaining to the BEA Tuxedo machine being clustered.
- Identifies and removes remaining IPC resources pertaining to the BEA Tuxedo machine being clustered.
- If the resource MachineType is not Single, the clean operation attempts to clean the domain bulletin board of remaining entries. The operation accomplishes this by executing "tmadmin" and connecting to the active master machine. Once connected, the operation issues the pclean sub-command for the target machine. If an active master machine is not found, or if an active master machine is found but the pclean sub-command fails, the clean operation exits with a success return code (0).

Identifying IPC resources pertaining to BEA Tuxedo

Symantec highly recommends installing each BEA Tuxedo instance to run as a unique UNIX login in the cluster.

This ensures maximum high availability to the Tuxedo domain.

The agent uses the following approach when identifying IPC resources pertaining to a particular BEA Tuxedo:

- If the UNIX login declared through the User attribute is unique within the cluster then the agent removes all IPC resources that this login owns. This ensures that stale IPC resources do not exist, which could prevent the online function of the resource on this cluster node.
- If the UNIX login declared through the User attribute is not unique within the cluster, then the following holds true:
  - The value of the User attribute indicates that the UNIX login has not been dedicated to this VCS resource alone.
  - The agent executes the tmipcrm BEA Tuxedo utility to list the IPC resources that belong to the current BEA Tuxedo machine instance.
  - The IPC resources listed via the tmipcrm utility are then removed. However, tmipcrm has been documented to fail. (For more details, refer to the *BEA Tuxedo Command Reference Guide* available on the BEA documentation Web site). If the tmipcrm utility cannot be successfully

executed, the stale IPC resources cannot be identified by the agent for removal.

In such an event, the onus of identifying and clearing such IPC resources pertaining to a particular BEA Tuxedo machine lies with the system administrator. Failure to do so can prevent the BEA Tuxedo machine from restarting on this cluster node.

Chapter 2

Installing, upgrading, and removing the agent for BEA Tuxedo

This chapter includes the following topics:

- Before you install the Veritas agent for BEA Tuxedo
- Installing the ACC library
- Installing the agent in a VCS environment
- Removing the agent in a VCS environment
- Removing the ACC library
- Upgrading the agent for BEA Tuxedo

Before you install the Veritas agent for BEA Tuxedo

You must install the Veritas agent for BEA Tuxedo on all the systems that will host a BEA Tuxedo service group.

Ensure that you meet the following prerequisites to install the agent for BEA Tuxedo.

- Install and configure Veritas Cluster Server.
- Remove any previous version of this agent.
- Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation on the agentpack disc.

See "About ACC Library" on page 22.

Prerequisites for installing the agent to support Solaris zones

Ensure that you meet the following prerequisites to install the agent for BEA Tuxedo:

- Install Tuxedo to support Solaris zones. For details, refer to the Tuxedo user documentation.
- Install and configure the VCS 5.0 environment to support Solaris zones. Refer to the VCS user documentation for details.
- Install the required version of ACC Library.
- Remove any previous version of this agent.

About ACC Library

The operations for the Veritas agent for BEA Tuxedo depend on a set of Perl modules known as the ACC library. The library must be installed on each system in the cluster that will run the agent for BEA Tuxedo. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

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. Navigate to the pkgs directory (the pkgs directory on the CD).

AIX	*cd_mount*/aix/application/acc_library/vcs/*version*_library/pkgs
HP-UX	*cd_mount*/hpux/generic/application/acc_library/vcs/*version*_library/pkgs
Linux	*cd_mount*/linux/generic/application/acc_library/vcs/*version*_library/rpms
Solaris	*cd_mount*/solaris/*dist_arch*/application/acc_library/vcs/*version*_library/pkgs where *dist_arch* is sparc or sol_x64.

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

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

HP-UX
```
# swinstall -s 'pwd' VRTSacclib
```

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

Solaris
```
# pkgadd -d . VRTSacclib
```

4 For HP-UX, install the HP-UX patch PHCO_29042 if it is not already installed.

Installing the agent in a VCS environment

Install the agent for BEA Tuxedo on each node in the cluster.

To install the agent

1 Log in as superuser.

2 Navigate to the directory containing the package for the platform running in your environment.

AIX
```
cd_mount/aix/application/tuxedo_agent/
vcs_version/version_agent/pkgs
```

HP-UX
```
cd_mount/hpux/generic/application/tuxedo_agent/
vcs_version/version_agent/pkgs
```

Linux
```
cd_mount/linux/generic/
application/tuxedo_agent/vcs_version/
version_agent/rpms
```

Solaris
```
cd_mount/solaris/dist_arch/application/
tuxedo_agent/vcs_version/version_agent/pkgs
```

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

3 Install the package.

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

HP-UX
```
# swinstall -s 'pwd' VRTStuxedo
```

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

Solaris
```
# pkgadd -d . VRTStuxedo
```

Removing the agent in a VCS environment

You must uninstall the agent for BEA Tuxedo 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 BEA Tuxedo resources from the cluster. Use the following command to verify that all resources have been removed:

```
# hares -list Type=Tuxedo
```

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

```
# hatype -delete Tuxedo
```

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 BEA Tuxedo from each node in the cluster.

Execute the following command to uninstall the agent:

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

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.rte
HP-UX	# swremove VRTSacclib
Linux	# rpm -e VRTSacclib
Solaris	# pkgrm VRTSacclib

Upgrading the agent for BEA Tuxedo

To upgrade the agent, first remove the older version of the agent.

See "Removing the agent in a VCS environment" on page 24.

Then, follow the instructions to install the new agent software.

See "Installing the agent in a VCS environment" on page 23.

Chapter 3

Preparing to configure the agent for BEA Tuxedo

This chapter includes the following topics:

- About configuring the Veritas agent for BEA Tuxedo
- Importing the agent types files in a VCS environment
- Agent attributes
- Executing a customized monitoring program
- Configuring BEA Tuxedo resources for Solaris zones support

About configuring the Veritas agent for BEA Tuxedo

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

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

See "About sample configurations for the agent for BEA Tuxedo" on page 91.

Importing the agent types files in a VCS environment

To use the agent for BEA Tuxedo, you must import the agent types file into the cluster.

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

Do not use the VCS GUI interface to import the files in to the cluster.

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	/etc/VRTSvcs/conf/sample_Tuxedo/TuxedoTypes.cf
VCS 5.0	/etc/VRTSagents/ha/conf/Tuxedo/TuxedoTypes.cf
VCS 5.0 under Solaris zones	/etc/VRTSagents/ha/conf/Tuxedo/TuxedoTypes_zones.cf

   The following example assumes VCS 5.0 is installed:

```
# cp /etc/VRTSagents/ha/conf/Tuxedo/TuxedoTypes.cf .
```

4. Create a dummy main.cf file:

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

5. Create the Tuxedo resource type as follows:

```
# hacf -verify .

# haconf -makerw

# sh main.cmd

# haconf -dump
```

   The BEA Tuxedo agent type is now imported to the VCS engine.

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

Agent attributes

Table 3-1 lists the required attributes for the agent for BEA Tuxedo.

Table 3-1 Required attributes

Attribute name	Description
AppDir	Absolute path name of the application directory from which the application and administrative servers are booted on this machine. The agent requires that all deployed servers for this BEA Tuxedo instance share the same application directory. **Note:** If this instance of BEA Tuxedo makes use of the ULGPFX attribute, set the value of AppDir as specified in the guide. See "Configuring BEA Tuxedo using the ULGPFX attribute" on page 50. Type and dimension: string-scalar Default: "" Example: /bea/tux91mach1/apps/simpapp
EnvFile	Absolute path of the file to source for setting the environment prior to executing BEA Tuxedo administrative programs such as tmboot, tmshutdown, and tmadmin. The following shell environments are supported: Bourne, Korn, and C shell. Symantec recommends using the Bourne shell script included in the BEA Tuxedo installation ($TUXDIR/tux.env) and adding additional variables as required by your environment. Symantec also recommends storing the environment file on a shared disk along with the BEA Tuxedo software and application files (that is, files in $TUXDIR and $APPDIR). Also consider setting the PMID enviroment variable to the MachineName attribute value in this file, which is the same as defined in the MACHINES section of the UBBCONFIG file. Type and dimension: string-scalar Default: "" Example: /tuxedo/srvr1/tuxedo9.1/tux.env
MachineName	Logical machine name (LMID) assigned to this BEA Tuxedo machine as specified in the BEA Tuxedo configuration file. Although not required for clustering, to avoid confusion in the BEA Tuxedo configuration file, consider setting the machine's physical machine name (PMID) and the LMID to the same value. Type and dimension: string-scalar Default: "" Example: Machine1

Table 3-1 Required attributes *(continued)*

Attribute name	Description
MachineType	Identifies the bulletin board architecture and machine type of the BEA Tuxedo machine. The agent entry points use this value and other resource state information to determine how to boot, shut down, and monitor the BEA Tuxedo machine. The valid values and their definitions are as follows: ■ Value-Single; Model-SHM; MachineType-Master ■ Value-Master; Model- MP; MachineType-Master ■ Value-Backup; Model-MP; MachineType-Backup ■ Value-NonMaster; Model-MP; MachineType-NonMaster Type and dimension: string-scalar Default: Single Example: Master
RemoteShell	Absolute path name of the remote shell program used by agent entry points to remotely issue administrative BEA Tuxedo commands such as "tmboot" and "tmshutdown" on the system hosting the active master (DBBL). This attribute must be specified if MachineType is not Single (that is, when clustering a machine that is part of a multiple machine domain); otherwise, the attribute should be null (not specified). Type and dimension: string-scalar Default: "" Example 1: /bin/remsh Example 2: /usr/local/bin/ssh

Table 3-1 Required attributes *(continued)*

Attribute name	Description
TmbootOptMaster	Command-line options passed to tmboot when starting a master or backup machine to fulfill the role of active master. The online entry point uses these options to boot a master or a backup machine only if it cannot detect the existence of an active master machine for the domain (TuxDomain). If the online entry point detects an active master machine for the domain, the machine is booted using the options specified in TmbootOptNonMaster. This attribute applies only to resources where MachineType is Master or Backup. The program tmboot starts a BEA Tuxedo machine in whole or in part depending on the options specified. Refer to the BEA Tuxedo documentation for a complete list of options and their descriptions. If the resource is a BEA Tuxedo machine defined as the master in TUXCONFIG and the machine will not host any application servers (that is, it is only running the DBBL and other administrative processes), then follow Example 1. The option -M starts only the administrative processes on the master machine. If the resource is a BEA Tuxedo machine defined as the backup in TUXCONFIG and the machine will host application servers, then follow Example 2. The options "-M -l MachineName" start both administrative and application server processes on the master machine. The tmboot option -b should never be used because the online entry point will automatically include the -b option if it is forced to start the active master on the backup machine. If -b is specified as an option, the entry point will exclude it from the tmboot command and use the remaining options. The tmboot option -y should not be used because this would cause all BEA Tuxedo machines configured through the TUXCONFIG file to be booted outside VCS control. Type and dimension: string-scalar Default: "" Example 1: -M Example 2: -M -l machine1

Table 3-1 Required attributes *(continued)*

Attribute name	Description
TmshutdownOptMaster	Command-line options passed to tmshutdown to stop a BEA Tuxedo machine that is currently the active master of a multiple machine domain. The offline entry point uses these options if the resource MachineType is Master or Backup and it detects that the machine to be taken offline is the active master. The program tmshutdown stops a BEA Tuxedo application in whole or in part, depending on the options specified. Refer to the BEA Tuxedo documentation for a complete list of options and their descriptions. If the BEA Tuxedo machine is defined as the master in TUXCONFIG and the machine will not be hosting any application servers (that is, it is only running the DBBL and other administrative processes), then follow Example 1. The option -M stops only the administrative processes on the master server. If the BEA Tuxedo machine is defined as the backup in TUXCONFIG and the machine will be hosting application servers, then follow example 2. The command-line options "-M -l MachineName" stop both administrative and application processes on the master machine. The tmboot option -y should not be used because this would cause all BEA Tuxedo machines configured through the TUXCONFIG file to be shut down outside VCS control. Type and dimension: string-scalar Default: "" Example 1: -M Example 2: -M -l machine1
TuxDir	Absolute path name to the BEA Tuxedo software base installation directory. The agent entry points use TuxDir to locate programs such as tmboot, tmshutdown, and tmadmin. In a multiple machine domain, when the application model is MP, each machine in the domain must have a unique TuxDir value. This ensures that multiple machines can run on the same system simultaneously without conflict. Type and dimension: string-scalar Default: "" Example: /tuxedo/srvr1/tuxedo9.1

Table 3-1 Required attributes *(continued)*

Attribute name	Description
TuxDomain	Name of the BEA Tuxedo domain to which the machine belongs, which is the same name specified by the parameter DOMAINID in the RESOURCES section of the BEA Tuxedo configuration file. When the application model is MP, the agent entry points use the TuxDomain value to associate BEA Tuxedo resources in the same domain. The value is also used to ensure that only one machine in the domain is defined as MachineType Master and only one machine is defined as MachineType Backup. Type and dimension: string-scalar Default: "" Example: finance
User	UNIX login name assigned to the BEA Tuxedo machine. Symantec strongly recommends that the login name be unique within the cluster. Sharing a UNIX login name across BEA Tuxedo machines could compromise the high availability of the BEA Tuxedo setup. In case the UNIX login name is not dedicated to the BEA Tuxedo machine, it has to be registered with the agent by adding a + at the start of the actual login name. For more information about the behavior of the agent when a shared login is registered: See "Identifying IPC resources pertaining to BEA Tuxedo" on page 18. The login name must also be synchronized across the nodes in the cluster. In other words, the login name must resolve to the same UID and have the same default shell on each node in the cluster. 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 fails if the access to this repository fails. With this user, the agent entry points execute BEA Tuxedo administrative programs such as tmboot, tmshutdown, tmipcrm, and tmadmin. The user's login shell must be Bourne, Korn, or C shell. Type and dimension: string-scalar Default: "" Example 1 (dedicated UNIX login): tux91adm Example 2 (shared UNIX login): +tux91adm

Table 3-2 lists the optional attributes for the agent for BEA Tuxedo.

Table 3-2 Optional attributes

Attribute name	Description
MonitorProgram	Absolute path and file name of an external, user-supplied monitor executable. If specified, the monitor entry point executes this file to perform an additional BEA Tuxedo machine state check. There are no restrictions for what actions the external monitor program performs to determine the state of a BEA Tuxedo machine instance. The only constraint is that the external monitor program must return one of the following integer values: ■ 110 or 0: BEA Tuxedo machine is online ■ 100 or 1: BEA Tuxedo machine is offline ■ Any other value: BEA Tuxedo machine status is unknown Symantec recommends storing the external monitor utility in the shared disk directory specified in TuxDir to ensure that the file is always available on the online system. Arguments are supported. Type and dimension: string-scalar Default: "" Example 1: /tuxedo/srvr1/tuxedo9.1/bin/myMonitor.pl Example 2: /tuxedo/srvr1/tuxedo9.1/bin/myMonitor.sh arg1 arg2
ResLogLevel	Logging details performed by the agent for the resource. The valid values are as follows: ■ ERROR: Logs only error messages. ■ WARN: Logs error messages and warning messages. ■ INFO: Logs error messages and informational messages. ■ TRACE: Logs error messages and 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

Table 3-2 Optional attributes *(continued)*

Attribute name	Description
SecondLevelMonitor	Enables second-level monitoring for the resource and indicates how often it is run. Second-level monitoring is a deeper, more thorough state check of the configured BEA Tuxedo instance. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval. A similar interpretation may be assumed for succeeding values. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if MonitorInterval is set to 60 seconds and 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. Type and dimension: integer-scalar Default: 0 Example: 5
TmbootOptNonMaster	Command-line options passed to tmboot to start any type of BEA Tuxedo machine in a multiple machine domain as a non-master machine (that is, an active master is already running and this machine will join a live domain). These options apply to resources where MachineType is Master, Backup, or NonMaster. Follow Example 1 to support this scenario. The attribute is also used to specify command-line options passed to tmboot to start a stand-alone BEA Tuxedo machine (that is, where the resource MachineType is Single). Follow example 2 to support this scenario. Review the definition of attribute TmbootOptMaster to specify consistent values between these two attributes. Type and dimension: string-scalar Default: -y Example 1 (MP mode): -B machine1 -l machine1 Example 2 (SHM mode): -y

Table 3-2 Optional attributes *(continued)*

Attribute name	Description
TmshutdownOptNonMaster	If MachineType is Master, Backup, or NonMaster, the options specified here are passed to tmshutdown when stopping the machine as a non-master machine (that is, when the agent detects that the machine to be stopped is not the active master). Follow Example 1for this scenario. If MachineType is Single, the options specified here are passed to tmshutdown to stop a stand-alone, independent BEA Tuxedo machine. Follow example 2 for this scenario. Read the definition of attribute TmshutdownOptMaster to specify consistent values between these two attributes. Type and dimension: string-scalar Default: -y Example 1 (MP mode): -B machine1 -l machine1 Example 2 (SHM mode): -y
TuxServers	Comma-delimited list of BEA Tuxedo servers to monitor. Candidate server names can be obtained for a BEA Tuxedo server by executing tmadmin and issuing the sub-command psr. **Note:** Do not include DBBL in this list of servers since it may migrate between the master machine and the backup machine. Also, do not include BBL in the list since it is always included in the process search when executing the monitor entry point. Type and dimension: string-scalar Default: "" Example: simpserv, myserver, this server, JSL, WLS, WSH
TuxServices	Comma-delimited list of BEA Tuxedo program services to monitor. Candidate services can be obtained for a BEA Tuxedo server by executing tmadmin and issuing the sub-command psc. **Note:** The program services specified for PeopleSoft deployments using BEA Tuxedo must follow certain naming constraints. See “Registering long PeopleSoft service names ” on page 82. Type and dimension: string-scalar Default: "" Example: TOUPPER, MYSERVICE, THISSERVICE

Executing a customized monitoring program

You can configure the monitor function to execute a custom monitor utility to perform a user-defined BEA Tuxedo machine state check. The utility is executed in the context of the UNIX user that is defined in the User attribute. The environment is set by sourcing the file specified in the EnvFile attribute.

The monitor function executes MonitorProgram if:

- The MonitorProgram attribute value is set to a valid executable utility.
- The first level process check indicates that the BEA Tuxedo instance is online.
- The SecondLevelMonitor attribute is set to 1 and the second level check returns the server state as "online" or the SecondLevelMonitor attribute is set to a value greater than 1, but the second level check is deferred for this monitoring cycle.

The monitor function interprets the program exit code as follows:

110 or 0	BEA Tuxedo machine instance is online
100 or 1	BEA Tuxedo machine instance is offline
Any other value	BEA Tuxedo machine instance is unknown

To ensure that the custom monitor utility is always available to the agent, Symantec recommends storing the file in the directory where the BEA Tuxedo machine is installed.

Configuring BEA Tuxedo resources for Solaris zones support

To enable the agent for BEA Tuxedo to support Solaris zones, ensure that you perform the following configuration steps:

- Install each BEA Tuxedo on a dedicated Solaris zone.
- Preferrably, follow the Symantec recommendation of installing zones on a shared disk for convenient configuration, failover, and maintenance.
- Make sure that the name of the Solaris zone is the same as the virtual host name that you use to install and configure the BEA Tuxedo.
  For sample service groups that depict Solaris zone support:

- Ensure that you have set the value of ContainerName attribute to the name of the Solaris zone.
  By default the agent function executes in the Global zone.

Chapter 4

Clustering BEA Tuxedo

This chapter includes the following topics:

- Overview of BEA Tuxedo
- Installing BEA Tuxedo software and applications on shared disks
- Basic steps to cluster a BEA Tuxedo machine
- Configuring BEA Tuxedo using the ULGPFX attribute
- Additional requirements for multiple machine domains (MP model)
- BEA Tuxedo entities in a clustered environment
- Virtualizing BEA Tuxedo

Overview of BEA Tuxedo

The following paragraphs are excerpts from several BEA Tuxedo product documents. These documents are available on the BEA product documentation web site.

BEA Tuxedo provides the framework, or middleware, for building and deploying scalable multi-tier client/server applications across a variety of platforms. BEA Tuxedo middleware consists of software services that exist between a client or server application and the operating system. Middleware services provide a more functional set of application programming interfaces (API) than the operating system and network services, which helps to solve application connectivity and interoperability problems. The BEA Tuxedo platform supports distributed transaction processing, message-based application development, unlimited scalability, and standards-based interoperability, which are essential to a service-oriented architecture.

To understand the BEA Tuxedo system and the applications built with it, it is important to understand the following terms and concepts:

BEA Tuxedo Domain	A BEA Tuxedo domain, also known as a BEA Tuxedo application, is a set of BEA Tuxedo system, client, and server processes administered as a single unit from a single BEA Tuxedo configuration file. A BEA Tuxedo domain consists of many system processes, one or more application client processes, one or more application server processes, and one or more computer machines connected over a network. A BEA Tuxedo domain has the same meaning as a BEA Tuxedo application. A BEA Tuxedo domain may consist of one BEA Tuxedo server running on one computer (SHM model) or it may consist of multiple BEA Tuxedo servers running on multiple computers (MP model).
BEA Tuxedo Configuration File	Each BEA Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The text version of the configuration file is often referred to as UBBCONFIG, but it may have any name. The text file is converted to a binary file during the application deployment process. The binary version of the UBBCONFIG file is referred to as TUXCONFIG, although it too may be given any name.
BEA Tuxedo Installation Directory (TUXDIR)	The TUXDIR environment variable defines the product installation directory of the BEA Tuxedo software. It must be set to an absolute path name ending with the name of the product installation directory.
BEA Tuxedo Application Directory (APPDIR)	The absolute path name of the application directory in which application server files are stored and booted for a BEA Tuxedo machine. APPDIR may be set to more than one application directory.
BEA Tuxedo Master Machine	The master machine for a BEA Tuxedo domain is a machine containing the domain's UBBCONFIG file, and is designated as the master machine in the RESOURCES section of the UBBCONFIG file. Starting, stopping, and administering the machines in a BEA Tuxedo domain is done through the master machine. The master machine for a BEA Tuxedo domain also contains the master copy of the TUXCONFIG file. Copies of the TUXCONFIG file are propagated to every other machine, referred to as a non-master machine, in a BEA Tuxedo domain whenever the BEA Tuxedo domain is booted on the master machine.

BEA Tuxedo Bulletin Board	The BEA Tuxedo system uses the TUXCONFIG file to set up a bulletin board on each machine in a BEA Tuxedo domain. When a BEA Tuxedo server process becomes active, it advertises the names of its services in the bulletin board. Some information in the bulletin board is global and is replicated on every machine in the BEA Tuxedo domain (for example, the names and locations of all servers offering a particular service). Other information is local and is visible only on the local bulletin board (for example, the actual number and type of client requests currently waiting on a local server request queue). The bulletin board provides location and namespace transparency within a BEA Tuxedo domain.
Distinguished Bulletin Board Liaison (DBBL)	DBBL is a process hosted on the BEA Tuxedo machine that is the current active master. The active master is the machine on which booting, shutdown, and other administrative tasks are performed. It is used only in multiple machine domains and is hosted either by the master or the backup machine.

BEA Tuxedo applications can be deployed in a single machine domain (SHM model) or across a multiple machine domain (MP model). A single machine domain consists of one BEA Tuxedo machine on which one or more applications are deployed. It operates as a stand-alone, independent machine and does not communicate with other BEA Tuxedo machines.

A multiple machine domain consists of multiple BEA Tuxedo machines across which one or more applications are deployed. The machines function as a collective unit, communicate their status with each other periodically, and depend on the active master machine to maintain the domain bulletin board. Protecting this domain bulletin board from fault conditions by automatically performing master migrations is a primary objective of the BEA Tuxedo agent.

The diagrams that follow depict both types of domains, along with some common machine components found within each type of domain.

Figure 4-1 shows a single machine domain.

In this figure, BEA Tuxedo machine 1, comprises three administrative processes (bulletin board liaison, workstation listener, and workstation handler) and two application server processes.

Figure 4-1 Single machine domain

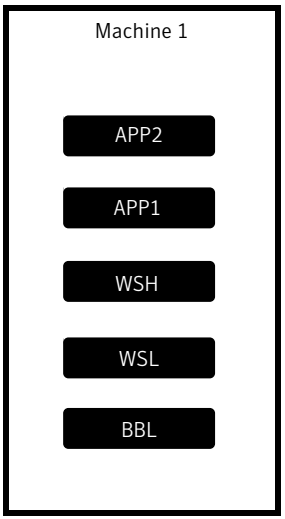
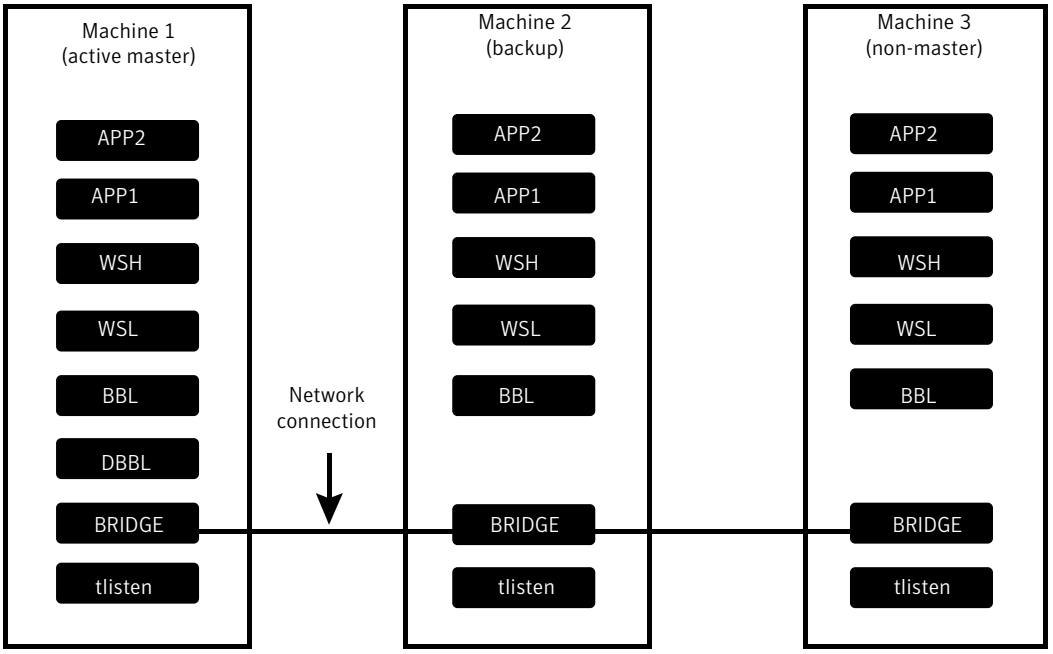


Figure 4-2 shows a multiple machine domain.

In this figure, three BEA Tuxedo machines are configured as one domain, with machine 1 currently hosting the active master (DBBL). Additional administrative processes that support network communications among the machines are required for this configuration, which include the server listener (tlisten), the BRIDGE process, and the distinguished bulletin board liaison (DBBL).

Figure 4-2 Multiple machine domain



Installing BEA Tuxedo software and applications on shared disks

The agent for BEA Tuxedo is designed to cluster BEA Tuxedo machines installed on either shared disks or on non-shared disks such as internal system drives. However, Symantec recommends installing the BEA Tuxedo program files, configuration files, application files, and data files (that is, the files contained by BEA Tuxedo environment variables TUXDIR and APPDIR) of each BEA Tuxedo machine on a shared disk that is managed by Veritas Volume Manager. Installing a BEA Tuxedo server and application on a shared disk, along with configuring the BEA Tuxedo application with a virtual host name, completely decouples the BEA Tuxedo server from any physical node in the cluster, thus providing the flexibility to run the BEA Tuxedo server on any node in the cluster.

In addition, placing the files of each BEA Tuxedo machine on its own shared disk managed by Veritas Volume Manager also simplifies the process of providing a disaster recovery solution for each BEA Tuxedo machine. By simply configuring

the BEA Tuxedo machine's volume for replication, the entire contents of the volume (that is, the entire BEA Tuxedo machine) can be replicated continuously to another volume at an alternate data center. In the event of a disaster, the volume at the alternate data center can be promoted and used to start the BEA Tuxedo machine in its last known state. Volume replication and disaster recovery processes can be completely managed and automated by VCS. Refer to the Veritas Volume Manager documentation for more information about replicating volumes for disaster recovery.

Directory structure recommendation

A well-designed directory structure for your clustered BEA Tuxedo application simplifies the cluster configuration, facilitates disaster recovery, and creates a storage environment that is more intuitive and easier to manage. Symantec recommends a directory structure similar to the one described in the following table. This sample directory structure depicts just two BEA Tuxedo machines, but the naming pattern supports an unlimited number of machines.

Table 4-1 describes the recommended directory structure.

Table 4-1 Directory structure recommendation

Directory	Purpose
/tuxedo	Base directory under which all mount point subdirectories are created.
/tuxedo/srvr1	Path used to mount the file system dedicated for BEA Tuxedo server 1 program, configuration, and application files (that is, both TUXDIR and APPDIR).
/tuxedo/srvr1/tuxedo9.1	BEA Tuxedo product installation directory (TUXDIR) dedicated for BEA Tuxedo server 1.
/tuxedo/srvr1/apps	Base directory in which all BEA Tuxedo application server files running within BEA Tuxedo server 1 are stored. Serves as the base directory for the BEA Tuxedo configuration variable APPDIR.
/tuxedo/srvr2	Path used to mount the file system dedicated for BEA Tuxedo server 2 program, configuration, and application files (that is, both TUXDIR and APPDIR).
/tuxedo/srvr2/tuxedo9.1	BEA Tuxedo product installation directory (TUXDIR) dedicated for BEA Tuxedo server 2.

Table 4-1 Directory structure recommendation *(continued)*

Directory	Purpose
/tuxedo/srvr2/apps	Base directory in which all BEA Tuxedo application server files running within BEA Tuxedo server 2 are stored. Serves as the base directory for the BEA Tuxedo configuration variable APPDIR.

Continue with the same naming pattern for BEA Tuxedo server 3, server 4, and other BEA Tuxedo servers.

Basic steps to cluster a BEA Tuxedo machine

The process of clustering a BEA Tuxedo machine is less complex for an SHM application model than an MP application model. When the model is SHM, each BEA Tuxedo machine is independent and is configured and clustered independent of other BEA Tuxedo machines in the environment. When the model is MP, all BEA Tuxedo machines depend on the active master machine and must be configured and managed relative to the active master machine. For both application models, one primary goal when clustering a BEA Tuxedo machine is to decouple it from any node in the cluster, especially the node on which the BEA Tuxedo software installation was performed. Decoupling is generally achieved by first installing the software on shared disk and then using a dedicated virtual host name to configure the BEA Tuxedo machine's various network addresses.

Regardless of the application model, use the following steps as a basic guide for clustering each BEA Tuxedo machine.

To cluster BEA Tuxedo machines

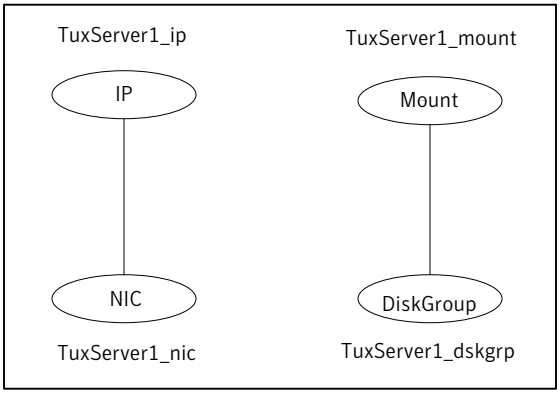
1. Allocate a shared disk resource (for example, LUN) for the BEA Tuxedo machine to be clustered.
2. Create a Veritas disk group, volume, and file system on the shared disk resource. (Although not recommended, BEA Tuxedo machines can be clustered without using Veritas Volume Manager or Veritas File System. The tight integration between VCS, Veritas Volume Manger, and Veritas File System ensures a more comprehensive and resilient high availability solution for your BEA Tuxedo machine.)
3. Obtain a dedicated virtual IP address and host name for the BEA Tuxedo machine. This network address and host name will be used exclusively by this BEA Tuxedo machine, regardless of the node in the cluster on which it is running.

4 Obtain a dedicated UNIX account for the BEA Tuxedo machine. It will use this account exclusively, regardless of the node in the cluster on which it is running.

5 Create a VCS service group that will contain the resources supporting the BEA Tuxedo machine. To maximize the flexibility of the cluster configuration, each VCS service group created for BEA Tuxedo should manage only one BEA Tuxedo machine.

6 Place all disk and network resources under VCS control. Then, bring the service group online on the system in which the BEA Tuxedo software installation will be performed.

The following figure shows such a service group.



7 Install the BEA Tuxedo software and application files. Make sure that the BEA Tuxedo installer places the files on the shared disk file system previously established for this machine.

The following excerpt from a BEA Tuxedo installation session illustrates this step:

```
==================================================
Choose BEA Home Directory
-------------------------
    1- Create a New BEA Home
    2- Use Existing BEA Home
Enter a number: 1
Specify a New BEA Home: /tuxedo/srvr1
==================================================
Choose Product Directory
------------------------
    1- Modify Current Selection (/tuxedo/srvr1/tuxedo9.1)
    2- Use Current Selection (/tuxedo/srvr1/tuxedo9.1)
Enter a number: 2
==================================================
Installing...
```

8 When creating the BEA Tuxedo configuration file for the application (TUXCONFIG), ensure that all parameters requiring a network address are specified using the virtual host name dedicated to the BEA Tuxedo machine. Do not use the administrative host name or node name of one of the nodes in the cluster for these parameters.

The following parameters, if used in your application, should be configured with the virtual host name:

NADDR	The complete network address to be used by the BRIDGE process, that is, the listening address on the LMID.
NLSADDR	The network address used by the tlisten process servicing the network on the node identified by the LMID.
WSL	A Workstation Listener process, running on the BEA Tuxedo machine, which accepts connection requests from ATMI clients and assigns connections to a Workstation Handler (WSH).
WSH	A Workstation Handler (gateway) process, running on the BEA Tuxedo machine, which handles communications between ATMI clients and the BEA Tuxedo server application.
JSL	A Jolt Server Listener process, running on the BEA Tuxedo machine, which accepts connection requests from Jolt clients and assigns connections to a Jolt Server Handler (JSH).
JSH	A Jolt Server Handler (gateway) process, running on the BEA Tuxedo machine, which handles communications between Jolt clients and the BEA Tuxedo server application.
ISL	An IIOP Listener process, running on the BEA Tuxedo machine, which accepts connection requests from CORBA clients and assigns connections to an IIOP Server Handler (ISH).
ISH	An IIOP Server Handler (gateway) process, running on the BEA Tuxedo machine, which handles communications between CORBA clients and the BEA Tuxedo server application.

It is important to specify virtual host names instead of virtual IP addresses for these parameters if the BEA Tuxedo server volumes are replicated to an alternate computing site using Veritas Volume Replicator (to provide a disaster recovery solution). The IP subnets at both computing sites are likely to be different; thus, the virtual IP address for a given BEA Tuxedo machine will be different on the two sites, but the virtual host name may be the same. When switching a BEA Tuxedo machine to run on the alternate computing site, the domain name server (DNS) will need to be updated to reflect the change in the BEA Tuxedo machine's virtual IP address. This ensures that all

network traffic targeted for this BEA Tuxedo machine is directed to the alternate computing site. This DNS update task can be performed automatically by VCS using the Global Cluster Option.

9 In the MACHINE section of the BEA Tuxedo configuration file, set the PMID (physical machine identifier) and the LMID (logical machine identifier) for each machine to the same logical value. Since the BEA Tuxedo machine is completely decoupled from any physical node in the cluster, both the PMID and LMID can now represent the logical names for the BEA Tuxedo machine. Thus, Symantec recommends setting both PMID and LMID to the same value to minimize the number of different names for the machine. Some sample names for these logical identifiers are machine1, machine2, server1, and server2.

PMID and LMID are names internal to a BEA Tuxedo application and are not used for TCP/IP network communications.

10 Add the variable PMID to the BEA Tuxedo machine's environment file (EnvFile).

The following is a sample excerpt from an environment file (Bourne shell syntax):

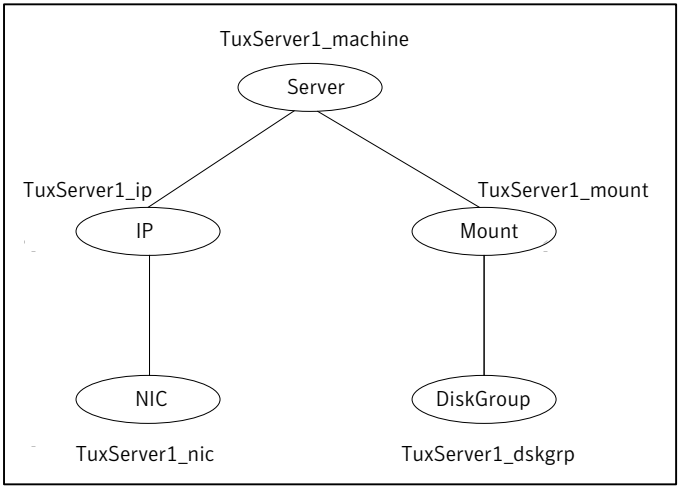
```
PMID=server1; export PMID
APPDIR=/tuxedo/srvr1/apps/simpapp; export APPDIR
TUXCONFIG=/tuxedo/srvr1/apps/tuxconfig; export TUXCONFIG
TUXDIR=/tuxedo/srvr1/tuxedo9.1; export TUXCONFIG
```

11 Load the BEA Tuxedo configuration file into its binary format using tmloadcf. Be sure the PMID environment variable is set before executing tmloadcf.

12 Before placing the BEA Tuxedo machine under VCS control, verify the BEA Tuxedo machine can be successfully started and stopped manually. Be sure to use the environment file established for this BEA Tuxedo machine (EnvFile) and the dedicated user account (User).

13 Using the Tuxedo type, place the BEA Tuxedo machine under VCS control by creating a resource for it. Test the BEA Tuxedo resource and the entire service group by bringing it online on each node in the cluster one at a time.

The following figure shows such a service group.



14 If the BEA Tuxedo machine is part of a multiple machine domain, create a VCS resource to manage the BEA Tuxedo listener that supports the BEA Tuxedo machine.

For detailed instructions:

See "Clustering BEA Tuxedo listeners" on page 57.

15 If the BEA Tuxedo machine is part of a multiple machine domain, be sure to carefully read all the configuration notes in the following sections and implement those that are relevant.

See "Additional requirements for multiple machine domains (MP model)" on page 52.

Configuring BEA Tuxedo using the ULGPFX attribute

By default, BEA Tuxedo logs are located at the following location:

AppDir/log/ULOG.*mmddyy*

The default location can be overridden using the ULGPFX attribute in the UBBCONFIG file. The TUXCONFIG file generated from this file is then used for booting the BEA Tuxedo machine.

The following is an excerpt from a UBBCONFIG file that makes use of the ULGPFX attribute:

```
*RESOURCES
IPCKEY          111223
DOMAINID        simpapp
MASTER          tux91sol
MAXACCESSERS    10
MAXSERVERS      5
MAXSERVICES     10
MODEL           SHM
LDBAL           N

*MACHINES
DEFAULT:
                APPDIR="/bea/tux/tuxedo91/app/simpapp"
                ULGPFX="/bea/tux/tuxedo91/backup/log/ULOG"
                TUXCONFIG="/bea/tux/tuxedo91/app/simpapp/tuxconfig"
                TUXDIR="/bea/tux/tuxedo91"
```

If your BEA Tuxedo machine makes use of the ULGPFX attribute, set the value of AppDir as follows:

```
$ dirname ULGPFX
```

For example, in this case the value is set as follows:

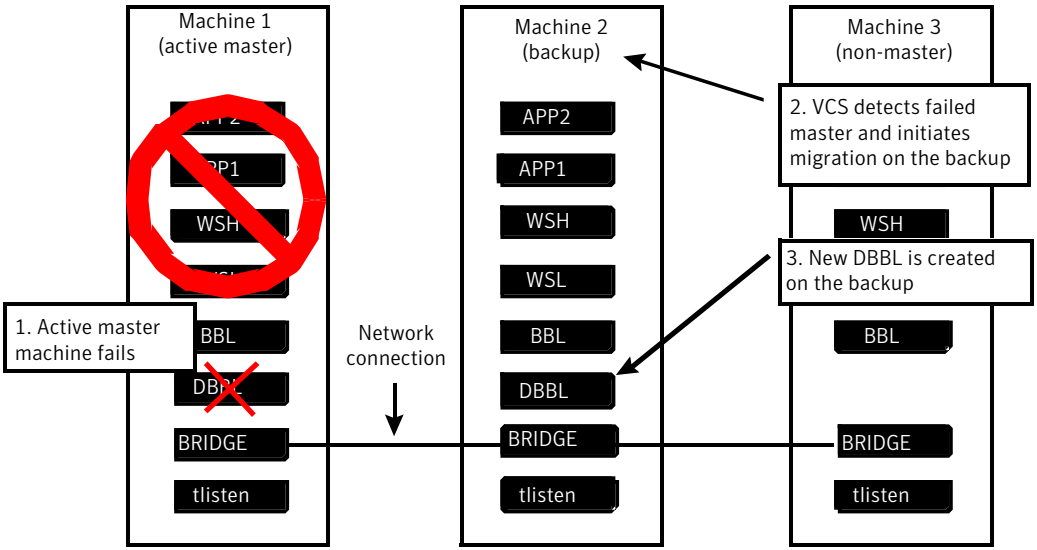
```
$ dirname /bea/tux/tuxedo91/backup/log/ULOG
/bea/tux/tuxedo91/backup/log
```

Use the value returned by the command to set the value for the AppDir agent attribute. The agent will not be able to detect such an instance of BEA Tuxedo unless the value of AppDir has been set as specified.

For information about the AppDir attribute:

See Table 3-1 on page 29.

Figure 4-3 Automated master migration



High availability scenarios supported

The following sections describe two high availability scenarios relevant to the complexities just discussed and how the BEA Tuxedo agent handles them.

Scenario 1: Active master machine faults

In the initial state of this scenario, the master, backup, and one or more non-master machines are online and functioning properly.

The following holds true for this scenario:

- The active master machine faults unexpectedly and abruptly (that is, DBBL and BBL no longer exist).
- During its next monitor interval, the VCS resource for the backup machine detects that the active master is missing and initiates a master migration process in order to salvage the domain bulletin board.
- While the backup machine performs the master migration, the faulted active master machine switches over to another node in the cluster and starts as the new backup machine. Note that the faulted active master machine does not attempt to restart until the backup machine completes the master migration process.

- Also, while the backup machine performs the master migration, the non-master machines may detect that the active master is missing. On detection, they start a wait and search again algorithm for the remainder of the MonitorTimeout period, which provides ample time for the backup machine to complete the master migration. When the migration completes, the non-master machines detect the new active master.
- The backup machine completes the master migration and the domain bulletin board survives.
- The faulted active master machine now serves as the backup machine.

Scenario 2: Active master machine faults, then backup machine faults (cascading fault)

In the initial state of this scenario, the master, backup, and one or more non-master machines are online and functioning properly.

The following holds true for this scenario:

- The active master machine faults unexpectedly and abruptly (that is, DBBL and BBL no longer exist). For whatever reason, the faulted active master machine is not able to automatically switch and restart on another system in the cluster.
- During its next monitor interval, the VCS resource for the backup machine detects that the active master is missing and initiates a master migration process in order to salvage the domain bulletin board.
- The non-master machines wait patiently for the backup to become the new active master.
- The backup machine completes the master migration and the domain bulletin board survives.
- Now the backup machine, which is hosting the active master, also faults unexpectedly and abruptly (that is, DBBL and BBL no longer exist).
- Since the original machine hosting the active master failed to switch and restart (refer to step 2), the domain bulletin board is now lost and a master migration process is not possible. But VCS detects this second fault and automatically switches the faulted backup machine to another node in the cluster.
- The faulted backup machine is successfully restarted and a new domain bulletin board is established.
- As the domain bulletin board was lost, the remaining non-master machines are faulted by VCS and are restarted or switched to other systems. The

non-master machines become online and join the new domain bulletin board. The BEA Tuxedo application thus remains available.

About service group dependencies with the MP model

A VCS service group dependency provides a mechanism by which two service groups can be linked by a dependency rule, similar to the way resources are linked. A common use of service group dependencies is to ensure that an application's database server is started before its application servers.

VCS service group dependencies should not be created among BEA Tuxedo service groups that are part of a multiple machine domain. This restriction is necessary due to BEA Tuxedo's ability to migrate the active master machine (that is, the DBBL) back and forth between the master and backup machines. In a normal scenario, to start a BEA Tuxedo application configured as MP, the master machine is started first to establish the DBBL, then the remaining machines are started. But if the previous shutdown of the application had been performed while the active master was hosted on the backup machine (for example, due to a fault of the master machine or due to a manual master migration), then the first machine that starts during the next application startup should be the backup machine, since it was the last machine to host the active master. When the BEA Tuxedo application is offline, the BEA Tuxedo agent is only aware that the active master may be hosted by either the master or backup machine.

In the event that multiple BEA Tuxedo service groups are started simultaneously (including the machine with the active master), the BEA Tuxedo agent ensures that the master machine is started first. The startup process of the remaining machines is delayed sufficiently until the active master is established, at which time they may join the domain as a backup or non-master machine.

As service group dependencies cannot be used with multiple machine domains, a VCS administrator can inadvertently take a service group currently hosting the active master machine offline while the non-master machines are still online. If the backup machine is online, VCS will detect the missing active master and initiate a master migration process on the backup machine. Thus, the inadvertent offline operation of the active master machine does not shut down the entire BEA Tuxedo application, but forces a master migration to occur.

Configuring Remote Shell or OpenSSH

To manage a multiple machine domain, the BEA Tuxedo agent requires either Remote Shell (rsh) or OpenSSH (ssh) to be installed on each node in the cluster that runs a BEA Tuxedo machine. In addition, each dedicated account (User) created for the BEA Tuxedo machines must be configured to use the remote access solution. Both remote access solutions enable the agent entry points to remotely

execute BEA Tuxedo administrative commands on the active master machine from the other non-master machines in the domain, which are likely running on different systems from the active master. Remotely executed commands include tmboot (to start a non-master machine), tmshutdown (to stop a non-master machine), and tmadmin (to monitor the state of a running non-master machine). For example, if the active master machine is online on systemA, and if a non-master machine is to be brought online on systemB, then the online entry point running on systemB must issue a tmboot command for the non-master machine that executes remotely on systemA.

Using OpenSSH, the command to be executed on systemB is as follows:

```
$ ssh systemA 'tmboot -B NonMasterMachine -l NonMasterMachine'
```

Using Remote Shell, the command to shut down the same non-master machine is as follows:

```
$ rsh systemA 'tmshutdown -B NonMasterMachine -l NonMasterMachine'
```

The sample commands are simplified to illustrate how the agent entry points use the remote commands. The actual remote commands issued also incorporate the appropriate BEA Tuxedo user account (User) and environment file (EnvFile) under which the command should be executed. In addition, if OpenSSH is used, the -x option is included in the remote command to disable X11 forwarding during execution, which avoids potential delays caused by X11 authentication.

When configuring the remote access solution, ensure that commands execute on remote nodes without attaching to a controlling terminal. This configuration would also allow cron jobs to utilize rsh and ssh in its scripts. In other words, the remote command setup should allow background jobs, but the jobs must not attempt to gain a controlling terminal. This configuration can be verified by creating a test script that uses rsh or ssh to execute a simple command on a remote server and enabling the script using a cron job.

The remote access solution that is used by the agent is specified for each BEA Tuxedo resource using the attribute RemoteShell.

Tuning MonitorTimeout value

MonitorTimeout is an attribute of the type Tuxedo that specifies the maximum number of seconds within which the monitor entry point must complete or be terminated. Since MonitorTimeout is an attribute for the type Tuxedo, the value of MonitorTimeout affects the behavior of all resources created with Tuxedo. The default MonitorTimeout value for Tuxedo is 300 seconds, which is significantly larger than the 60 second default value used by other VCS agents. This longer than normal timeout period is essential to provide the necessary tolerance to

effect a master machine migration if the active master faults. Choosing 300 seconds as a default value assumes that a master migration process requires no more than approximately 55 seconds to complete (that is, executing the sub-command master -yes within tmadmin completes within 55 seconds). The MonitorTimeout value in your environment should be analyzed and tuned to reflect the maximum time required to complete a master migration process should the active master machine become unavailable unexpectedly.

You can determine the maximum period of time to effect a master migration by performing a simple test. Take the service group hosting the active master machine offline while the backup machine and one or more non-master machines are still online. This scenario causes the agent to detect the missing active master and to respond by recreating the DBBL from its bulletin board information, which requires the longest migration period. Symantec recommends testing this scenario and others to fully understand the longest time required for a master migration, and then set MonitorTimeout to support this maximum migration time. In addition, keep in mind that, in the event of a true fault, a longer MonitorTimeout period also means a longer delay until the automated switch begins.

Configuring ToleranceLimit attribute

ToleranceLimit is an attribute of the type Tuxedo which specifies the number of times that the monitor entry point responds with the message that the BEA Tuxedo resource is offline before declaring the resource faulted. As an attribute for the Tuxedo type, the value specified for it applies to all VCS resources created using Tuxedo. The default value is 0, which means a BEA Tuxedo resource is faulted the first time the monitor entry point determines that the resource became offline unexpectedly.

You should consider increasing the ToleranceLimit value of the Tuxedo type to 1 or 2. Increasing the ToleranceLimit allows the agent to collaborate with the self-monitoring and self-recovery features built into the BEA Tuxedo product. For example, if a BBL, BRIDGE, or application server process was terminated accidentally, the surviving BEA Tuxedo server administrative processes can detect its absence and re-spawn it. Increasing the ToleranceLimit provides the BEA Tuxedo machine the latitude to self-recover before the agent finally faults the BEA Tuxedo machine and initiates a restart or switch.

Clustering BEA Tuxedo listeners

In a multiple machine domain (application model is MP), a BEA Tuxedo listener process (tlisten) must be started for each machine in the networked BEA Tuxedo application before the application is booted. The listener process enables the BEA

Tuxedo active master machine to start, shut down, and administer BEA Tuxedo processes running on the non-master machines.

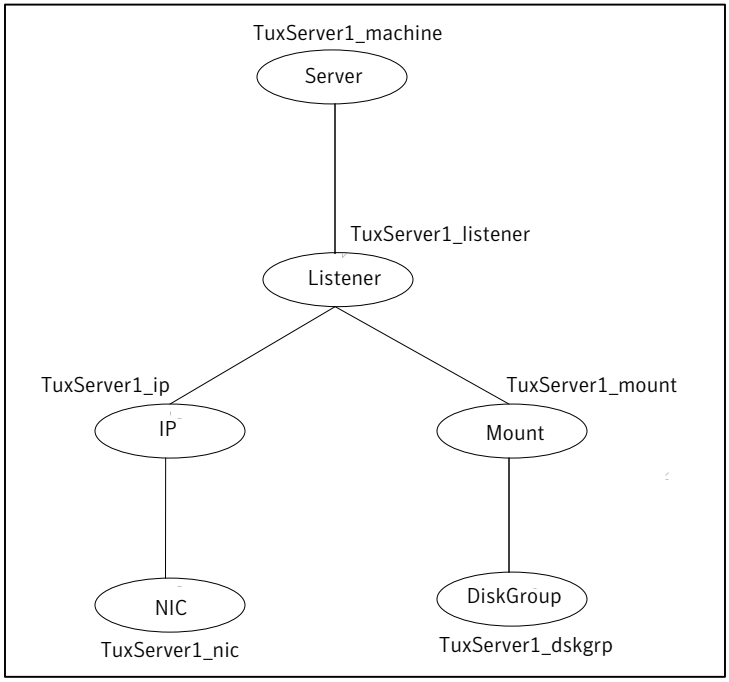
Symantec recommends placing each listener process under VCS control using the application agent that comes bundled with VCS. The listener resource should be created in the same VCS service group as the BEA Tuxedo machine it supports. In addition, the listener resource must be configured as a child of the BEA Tuxedo machine it supports.

A VCS resource definition for the listener process that can serve as a model for your environment is shown as follows.

```
       Application TuxServer1_listener (
                User = tuxsrvr1
                StartProgram =
"/tuxedo/srvr1/tuxedo9.1/bin/startListener"
                StopProgram =
"/tuxedo/srvr1/tuxedo9.1/bin/stopListener"
                MonitorProcesses = { "tlisten -l //tuxsrvr1:5555" }
                )
        TuxServer1_machine requires TuxServer1_listener
```

Figure 4-4 shows how a service group appears after the listener is configured.

Figure 4-4 Sample service group with a listener configured



The scripts startListener and stopListener referenced in this example are not included with the BEA Tuxedo software or the agent for BEA Tuxedo. They must be created by the VCS administrator.

For your convenience, sample scripts are provided as follows.

The script startListener is as follows:

```
#!/bin/sh
. /tuxedo/srvr1/tuxedo9.1/tux.env ; tlisten -l //tuxsrvr1:5555
```

This script starts the listener process configured to listen on a virtual host 'tuxsrvr1' at port 5555.

The script stopListener is as follows:

```
#!/bin/sh
pid='ps -ef | grep tlisten | grep -v grep | grep tuxsrvr1 | awk
'{print $2}''
if [ "X$pid" != "X" ]; then
               kill $pid
```

```
fi
exit 0
```

The start script is simply a one-line program that sources the BEA Tuxedo environment file for the BEA Tuxedo machine and then executes the tlisten program with the appropriate network address for the listener process. Note that in the VCS resource definition, the application agent ensures that tlisten is executed with the account specified in the attribute User (for example, tuxsrvr1).

The stop script is also a short program that simply identifies the listener's process identifier (PID) using the search strings "tlisten" and the user under which tlisten is executed (for example, tuxsrvr1). Once the PID is identified, it is used to send a SIGTERM signal (kill $pid) to the process, which is the recommended method to stop the listener.

BEA Tuxedo entities in a clustered environment

A service group containing all resources that can support a BEA Tuxedo machine instance in a clustered environment forms a basic setup to cluster the server.

The required resources are as follows:

Disk Group	A disk group contains a volume and a file system, which is a mount resource containing theBEA Tuxedo machine installation files. Use the DiskGroup resource type to create this resource. Also, create the resource on a shared disk so that you can import the group into any system in the cluster.
Mount	The mount resource mounts, monitors, and unmounts the file system that is dedicated to the BEA Tuxedo machine installation files. Use the Mount resource type to create this resource.
Network Interface	The Network Interface resource monitors the Network Interface Card (NIC) through which the BEA Tuxedo machine instances communicates with the other services. Use the NIC resource type to create this resource.
Virtual IP	The Virtual IP resource configures the virtual IP address dedicated to the BEA Tuxedo machine instance. The external services, programs and clients use this address to communicate with this instance. Use the IP resource type to create this resource.

BEA Tuxedo Server	The BEA Tuxedo Server resource starts, stops, and monitors the BEA Tuxedo Server instance. Use the BEA Tuxedo resource type to create this resource.

Virtualizing BEA Tuxedo

To ensure that your BEA Tuxedo machine can function properly on any node of the cluster, you need to virtualize all the parameters that could be dependent on a particular node.

Review the following basic notes for virtualization:

Host names	When installing and configuring the BEA Tuxedo machine, ensure that you enter the virtual host name associated with the IP address used to configure the IP resource. This ensures that if the application needs to be migrated, you are not tied down by the physical IP address given to the BEA Tuxedo machine. Refer to the definition of the EnvFile and MachineName attributes for information on how to set the LMID environment variable.
Path names	Ensure that your application gets installed on a shared disk so that it is not constrained by anything that is local to the node. If this is not possible every time, make sure that the local data is available on each configured node.

Chapter 5

Clustering PeopleSoft Enterprise Servers

This chapter includes the following topics:

- Overview of PeopleSoft
- Creating the BEA Tuxedo environment file
- Testing the Tuxedo environment file
- Sample resource definition
- Obtaining values for attributes MachineName and TuxDomain
- Additional PeopleSoft configuration notes
- Registering long PeopleSoft service names

Overview of PeopleSoft

This section provides additional instructions and examples to cluster PeopleSoft Enterprise middle-tier servers using the BEA Tuxedo agent. It is highly recommended that you enlist the assistance of your PeopleSoft application administration team to complete this task. The PeopleSoft administration team will be familiar with the programs and files referenced in the following sections.

The agent for BEA Tuxedo can be used to cluster PeopleSoft Enterprise application servers and process schedulers. The PeopleSoft Enterprise application architecture (also known as the PeopleSoft Internet Architecture) utilizes BEA Tuxedo servers for its middle-tier application servers and process schedulers; thus, these middle-tier services can be made highly available using the BEA Tuxedo agent.

Normally, PeopleSoft application servers and process schedulers are created, configured, started, and stopped using the PeopleSoft administrative utility called psadmin. These PeopleSoft servers can also be started and stopped using BEA Tuxedo's administrative utility called tmadmin, which is the interface used by the BEA Tuxedo agent. Creating the correct UNIX environment file for the BEA Tuxedo server is one of the most important tasks towards successfully using tmadmin and the BEA Tuxedo agent to start, stop, and monitor a PeopleSoft server. It is essential to source a complete and accurate environment file before executing tmadmin to start or stop a BEA Tuxedo server instance. When defining a VCS resource using the BEA Tuxedo agent, this environment file is identified by the attribute EnvFile.

Creating the BEA Tuxedo environment file

Since the PeopleSoft environment does not provide an environment file sufficiently complete to be used with tmadmin, you need to create the file manually.

The required environment variables and values to be included in the BEA Tuxedo environment file for a server instance can be obtained easily using one of following two methods:

- Start the PeopleSoft server outside of VCS control and use the Berkeley version of the UNIX process status (ps) command to display the environment in which the server is running.
- Use the psadmin utility to capture the environment in which the server is running.

The first method is easier and should be used on all platforms except HP-UX (due to a ps command output limitation). The second method is more involved but is the best method on HP-UX. Both methods are described in complete detail in the following sections.

Method 1: Capturing the environment using the ps command

This method must be used for all platforms except HP-UX. The output from the ps command on HP-UX is limited to 1024 characters; thus, some environment variables may be excluded due to truncation.

Perform the following steps to obtain the required environment variables and values for a PeopleSoft server instance using the output of the ps command and to construct the final file. The steps are applicable to both application servers and process schedulers.

To capture the environment using the ps command

1. Start the PeopleSoft Tuxedo server instance outside of VCS control. Be sure to start the server using the dedicated account that has been assigned to the PeopleSoft Tuxedo server (refer to the description of the Tuxedo agent attribute User for important requirements regarding this dedicated, unique account).
2. Once the server is fully instantiated, execute the process status (ps) command against the Tuxedo server's BBL process. The command should include the option to display the environment of the process.

   The ps command syntax is as follows:

Solaris	`/usr/ucb/ps -auxeww` *`PID for BBL process`*
AIX	`/bin/ps auxew` *`PID for BBL process`*
Linux	`/bin/ps auxeww | grep BBL`

3. Copy the output from the column labeled COMMAND to your preferred text editor. Place each variable/value pair on a separate line. Be sure to add the "set" and "export" commands appropriate for the shell on each line in your environment file.

   The following is a sample session from a Solaris system that follows the process.

```
# su psftapp
$ cd /bea/psft8/PT8.49/
$ . ./psconfig.sh
$ cd appserv/
$ ./psadmin -c boot -d HR84
psappsrv.cfg has changed archiving old one...
Copying HR84/Archive/psappsrv.cfg to HR84/Archive
/psappsrv_051608_1245_24.cfg
Attempting to boot bulletin board...
tmadmin - Copyright (c) 1996-1999 BEA Systems, Inc.
Portions * Copyright 1986-1997 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
Tuxedo is a registered trademark.
No bulletin board exists. Entering boot mode.
INFO: BEA Tuxedo, Version 9.1, 64-bit, Patch Level 036
INFO: Serial #: 650522264137-2065448083901,
Expiration NONE, Maxusers 1000000
```

```
INFO: Licensed to: Oracle-Peoplesoft-ISV
Booting admin processes ...
exec BBL -A :
        process id=26543 ... Started.
1 process started.
Attempting to boot ...
INFO: BEA Tuxedo, Version 9.1, 64-bit, Patch Level 036
INFO: Serial #: 650522264137-2065448083901,
Expiration NONE, Maxusers 1000000
INFO: Licensed to: Oracle-Peoplesoft-ISV
Booting server processes ...
[Remaining boot details omitted]
13 processes started.
Request to boot domain HR84 was submitted successfully.
$ ps -ef | grep BBL
 psftapp 26543     1   0 14:15:32 pts/5    0:00 BBL
-C dom=TESTSERV_34687 -g 30002 -i 0 -u psftapp -U
 /bea/psft8/PT8.49/appserv
 psftapp 27125 26363  0 14:16:33 pts/5    0:00 grep BBL
$ /usr/ucb/ps -auxeww 26543
USER       PID %CPU %MEM   SZ  RSS TT       S    START  TIME COMMAND
psftapp  26543  0.0  0.410976 7664 pts/5    S 14:15:32  0:00 BBL
-C dom=TESTSERV_34687 -g 30002 -i 0 -u psftapp -U /bea/psft8
/PT8.49/appserv/HR84/LOGS/TUXLOG -m 0 -A APPDIR=/bea/psft8/PT8.49
/appserv/HR84:/bea/psft8/PT8.49/bin APP_PW=PS
BDMCONFIG=/bea/psft8/PT8.49/appserv/HR84/PSBDMCFG
CLASSPATH=/bea/psft8/PT8.49/appserv/classes
COBPATH=/bea/psft8/PT8.49/cblbin
CONF=/etc/VRTSvcs/conf/config CVS_RSH
=ssh CVS_SSH=ssh DISPLAY=vcslin5:0.0 EDITOR=/bin/vi
ELOG=/var/VRTSvcs/log/engine_A.log FIELDTBLS32=t
padm fielded buffer not aligned HOME=/root
HOSTNAME=vcssun68 HOSTTYPE=sparc HZ=100 ICU_DATA=NOTUSED
INFORMIXSERVER= IPC_TERMINATE_PROCESS=1 IS_PS_PLT=Y
LD_LIBRARY_PATH=/bea/psft8/PT8.49/appserv/HR84:/
bea/psft8/PT8.49/bin:/bea/psft8/tux/tuxedo91/lib:
/lib:/user/lib:/bea/psft8/PT8.49/jre/lib/sparcv9/
native_threads:/bea/psft8/PT8.49/jre/lib/sparcv9/server:
/bea/psft8/PT8.49/jre/lib/sparcv9::/usr/lib/mp
s/sasl2/64:/bea/psft8/PT8.49/bin:
/bea/psft8/PT8.49/bin/interfacedrivers:
/bea/psft8/PT8.49/bin/sqr/OR
A/bin:/bea/psft8/PT8.49/optbin:/bea/psft8/PT8.49/verity
```

```
/solaris/_ssol26/bin:/bea/psft8/PT8.49/ora92/
lib:/bea/psft8/tux/tuxedo91/lib:/bea/psft8/PT8.49
/verity/solaris/_ssol26/bin LIBPATH=:/bea/psft8/PT8
.49/verity/solaris/_ssol26/bin LOGNAME=root
MACHTYPE=sparc-sun-solaris2.9 MAIL=/var/mail/root MANPAT
H=/usr/man:/usr/share/man:/opt/freeware/man:
/opt/csm/man:/opt/pssp_to_csm/man:/opt/VRTS/man:/opt/VRT
Svlic/man:. NLS_LANG=AMERICAN_AMERICA.UTF8
OLDPWD=/bea/psft8/PT8.49 ORACLE_HOME=/bea/psft8/PT8.49/or
a92 ORACLE_SID=PSFDBSOL OSTYPE=solaris2.9
PAGER=/bin/less PATH=/bea/psft8/PT8.49/appserv/HR84:/bea/p
sft8/PT8.49/bin:/bea/psft8/tux/tuxedo91/bin:/bin:
/bea/psft8/PT8.49/bin:.:/bea/psft8/PT8.49/verity/so
laris/_ssol26/bin PERL5LIB=/opt/VRTSperl/lib/site_perl
PMID=psftapp PSAPPSERVSTARTUP=Y PSJLIBPATH=/b
ea/psft8/PT8.49/jre/lib/sparcv9/native_threads:
/bea/psft8/PT8.49/jre/lib/sparcv9/server:/bea/psft8/P
T8.49/jre/lib/sparcv9 PS_DB=ORA PS_DBVER=8.1.x
PS_DM_DATA=/bea/psft8/PT8.49/data PS_DM_LOG=/bea/psft
8/PT8.49/log PS_DM_SCRIPT=/bea/psft8/PT8.49/scripts
PS_HOME=/bea/psft8/PT8.49 PS_HOSTTYPE=solaris-8-
sparc PS_JRE=/bea/psft8/PT8.49/jre
PS_LIBPATH=/bea/psft8/PT8.49/bin
PS_MACH=vcssun68 PS_SERVDIR=/bea
/psft8/PT8.49/appserv/HR84 PS_SERVER_CFG=/bea/psft8/PT8.49
/appserv/HR84/psappsrv.cfg PS_TUXDEV=/dev/
tcp PWD=/bea/psft8/PT8.49/appserv SHELL=/sbin/sh
SHLIB_PATH=:/bea/psft8/PT8.49/verity/solaris/_ssol2
6/bin SHLVL=2 SQRDIR=/bea/psft8/PT8.49/bin/sqr/ORA/bin
SQR_HOME=/bea/psft8/PT8.49/bin/sqr/ORA TERM=v
t100 TM_BOOTPRESUMEDFAIL=Y TM_BOOTTIMEOUT=60
TM_GP_AUTOSPAWNEXIT_FIX=yes TM_RESTARTSRVTIMEOUT=60 TUX
CONFIG=/bea/psft8/PT8.49/appserv/HR84/PSTUXCFG
TUXDIR=/bea/psft8/tux/tuxedo91 TZ=Asia/Calcutta
USER=root VCS_HOME=/opt/VRTSvcs _=./psadmin
path=/bea/psft8/PT8.49/bin;
/bea/psft8/PT8.49/jre/bin:/bin:/sb
in:/usr/sbin:/usr/lbin:/usr/local/bin:/usr/openwin/bin:
/usr/xpg4/bin:/opt/VRTSvcs/bin:/opt/VRTS/bin:
/etc/vx/diag.d:/usr/X11R6/bin:/usr/aix/bin:
/usr/artic960/bin:/usr/bin:/usr/ccs/bin:/usr/java14/.priv
ate142/bin:/usr/java14/.private142/jre/bin:
/usr/java14/bin:/usr/java14/jre/bin:/usr/lib/boot/bin:
```

/usr/lib/ncs/bin:/usr/lib/smit/bin:/usr/linux/bin:
/usr/lpp/Tivoli_Management_Agent.client/bin:/usr/lpp/
Tivoli_Management_Agent.client/inst_root/opt/Tivoli/
lcf/bin:/usr/lpp/X11/Xamples/bin:/usr/lpp/X11/bi
n:/usr/lpp/X11/custom/bin:/usr/lpp/bos.alt_disk_install/bin:
/usr/lpp/cnls/bin:/usr/lpp/devices.artic
960/bin:/usr/lpp/devices.ssa.disk/bin:
/usr/lpp/diagnostics/bin:/usr/lpp/tls/bin:
/usr/lpp/zhls/bin:/usr/opt/ifor/bin:/usr/opt/ifor/ls/bin:
/usr/opt/ifor/ls/os/aix/bin:/usr/opt/perl5/bin:
/usr/sbin/rsct/bin:/usr/sbin/rsct/install/bin:
/usr/sbin/rsct/perl5/bin:/usr/ssa/disk/bin:
/usr/ssa/ssaraid/bin:/usr/suma/bin:
/usr/sysv/bin:/usr/websm/bin:/opt/IBMinvscout/bin:
/opt/Tivoli/lcf/bin:/opt/csm/bin:/opt/diag
nostics/bin:/opt/freeware/bin:
/usr/lpp/bos.net/inst_root/sbin:.
:/bea/psft8/PT8.49/bin:/bea/psft8/PT8
.49/bin/sqr/ORA/bin:/bea/psft8/PT8.49/
ora92/bin:/bea/psft8/PT8.49/verity/
solaris/_ssol26/bin:/bea/psft8/tux/tuxedo91/bin
TUXOFFSET=0
[Environment variables and values are emphasized in *italics*.]

After the environment file has been formatted and the “set” and “export” commands added, be sure to test the environment file.

Method 2: Capture the environment using the psadmin utility

This method must be used only on the HP-UX platform.

For all other operating systems:

See “Method 1: Capturing the environment using the ps command” on page 64.

Perform the following steps to obtain the list of required environment variables and values using the PeopleSoft psadmin utility and to construct the final file.

To capture the environment using the psadmin utility

1. Establish a telnet session to the system on which the PeopleSoft server to be clustered currently resides. Log in using the dedicated account that has been assigned to the PeopleSoft Tuxedo server (refer to the description of the Tuxedo agent attribute User for important requirements regarding this dedicated, unique account).
2. Once logged in, set the environment for the session in preparation to run the PeopleSoft psadmin utility. This is done by sourcing the environment file created by psadmin when the PeopleSoft server was defined. A PeopleSoft administrator will be familiar with this environment file and its location in the Tuxedo directory structure. Note that this file is sufficient to run psadmin, but it is an incomplete environment in which to run tmadmin and should not be used for the EnvFile attribute. Additional variables are required.

   The following is a sample session in which the environment is being set in preparation to run psadmin.

   ```
   # su psftapp
   $ cd /bea/psft8/PT8.49/
   $ . ./psconfig.sh
   $ cd appserv/
   ```

3 As an optional test, start the psadmin utility, navigate its menu structure, and start the PeopleSoft server outside of VCS control. A successful startup of the PeopleSoft server indicates that the proper PeopleSoft environment file was sourced. If the test succeeded, the PeopleSoft server should then be stopped using psadmin before proceeding to the next steps.

The following is a sample session in which psadmin is used to start a PeopleSoft application server.

```
$ ./psadmin
PSADMIN -- Tools Release: 8.49
Copyright (c) 1988, 2003, Oracle. All rights reserved.
---------------------------------
PeopleSoft Server Administration
---------------------------------
  1) Application Server
  2) Process Scheduler
  3) Search Server
  q) Quit
Command to execute (1-3, q): 1
-------------------------------------------
PeopleSoft Application Server Administration
-------------------------------------------
  1) Administer a domain
  2) Create a domain
  3) Delete a domain
  4) Import domain configuration
  q) Quit
Command to execute (1-4, q) : 1
---------------------------------
PeopleSoft Server Administration
---------------------------------
  1) Application Server
  2) Process Scheduler
  3) Search Server
  q) Quit
Command to execute (1-3, q): 1
-------------------------------------------
PeopleSoft Application Server Administration
-------------------------------------------
  1) Administer a domain
  2) Create a domain
  3) Delete a domain
  4) Import domain configuration
```

```
  q) Quit
Command to execute (1-4, q) : 1
Tuxedo domain list:
  1)  HR84
Select domain number to administer: 1
--------------------------------
PeopleSoft Domain Administration
--------------------------------
     Domain Name: HR84
  1) Boot this domain
  2) Domain shutdown menu
  3) Domain status menu
  4) Configure this domain
  5) TUXEDO command line (tmadmin)
  6) Edit configuration/log files menu
  7) Messaging Server Administration menu
  8) Purge Cache
  9) Preload File Cache
 10) Clean IPC resources of this domain
  q) Quit
Command to execute (1-10, q) : 1
-------------------------------
PeopleSoft Domain Boot Menu
-------------------------------
     Domain Name: HR84
  1) Boot (Serial Boot)
  2) Parallel Boot
  q) Quit
Command to execute (1-2, q) [q]: 1
psappsrv.cfg has changed archiving old one...
Copying HR84/Archive/psappsrv.cfg to HR84/Archive/
psappsrv_050908_1021_24.cfg
Attempting to boot bulletin board...
tmadmin - Copyright (c) 1996-1999 BEA Systems, Inc.
Portions * Copyright 1986-1997 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
Tuxedo is a registered trademark.
No bulletin board exists. Entering boot mode.
INFO: BEA Tuxedo, Version 9.1, 64-bit, Patch Level 036
INFO: Serial #: 650522264137-2065448083901,
Expiration NONE, Maxusers 1000000
INFO: Licensed to: Oracle-Peoplesoft-ISV
```

```
Booting admin processes ...
exec BBL -A :
        process id=18718 ... Started.
1 process started.
Attempting to boot ...
INFO: BEA Tuxedo, Version 9.1, 64-bit, Patch Level 036
INFO: Serial #: 650522264137-2065448083901,
Expiration NONE, Maxusers 1000000
INFO: Licensed to: Oracle-Peoplesoft-ISV
Booting server processes ...
exec PSWATCHSRV -A -- -ID 34687 -C psappsrv.cfg
-D TESTSERV -S PSWATCHSRV :
        process id=18744 ... Started.
exec PSAPPSRV -s@../psappsrv.lst -s@../psqcksrv.lst -sICQuery
-sSqlQuery:SqlRequest -- -C psappsrv.cfg
-D TESTSERV -S PSAPPSRV :
        process id=18745 ... Started.
exec PSAPPSRV -s@../psappsrv.lst -s@../psqcksrv.lst -sICQuery
-sSqlQuery:SqlRequest -- -C psappsrv.cfg
-D TESTSERV -S PSAPPSRV :
        process id=18804 ... Started.
exec PSAPPSRV -s@../psappsrv.lst -s@../psqcksrv.lst -sICQuery
-sSqlQuery:SqlRequest -- -C psappsrv.cfg
-D TESTSERV -S PSAPPSRV :
        process id=18811 ... Started.
exec PSSAMSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSSAMSRV :
        process id=18858 ... Started.
exec PSANALYTICSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSANALYTICSRV :
        process id=18865 ... Started.
exec PSANALYTICSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSANALYTICSRV :
        process id=18921 ... Started.
exec PSANALYTICSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSANALYTICSRV :
        process id=18940 ... Started.
exec PSRENSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSRENSRV :
        process id=18947 ... Started.
exec PSMONITORSRV -A -- -ID 34687 -C psappsrv.cfg
-D TESTSERV -S PSMONITORSRV :
        process id=18970 ... Started.
```

```
exec WSL -A -- -n //psftapp:7000 -z 0 -Z 0
-d /dev/tcp -I 5 -T 60 -m 1 -M
3 -x 40 -c 5000 -p 7001 -P  7003 :
        process id=19001 ... Started.
exec JSL -A -- -d /dev/tcp -n //psftapp:9000
-m 5 -M 7 -I 5 -j ANY
-x 40 -S 10 -c 1000000 -w JSH :
        process id=19003 ... Started.
exec JREPSVR -A -- -W -P /bea/psft8/PT8.49/
appserv/HR84/jrepository :
        process id=19009 ... Started.
13 processes started.
```

4 Extract the Tuxedo environment variables from the PeopleSoft system. First, within the environment set by the previous steps, execute the command psadmin –env and redirect the output to a file. This command captures most of the environment variables and values that are set internally by psadmin when it is executing tmadmin commands.

The following command illustrates this step:

```
# su psftapp
$ cd /bea/psft8/PT8.49/
$ . ./psconfig.sh
$ cd appserv/
$ ./psadmin –env > ./psadmin_env_cmd
```

5 This step is applicable only for PeopleSoft application servers. Skip this step if you are configuring a process scheduler.

Next, start the psadmin utility again and navigate to the PeopleSoft Domain Administration menu. Select the option TUXEDO command line to start a tmadmin session from within psadmin. Then, within the tmadmin session, run the command !env and redirect the output to another file. This captures additional variables not captured by the previous step.

The following excerpt illustrates a sample session:

```
bash-2.05$ ./psadmin
PSADMIN -- Tools Release: 8.49
Copyright (c) 1988, 2003, Oracle. All rights reserved.
---------------------------------
PeopleSoft Server Administration
---------------------------------
  1) Application Server
  2) Process Scheduler
  3) Search Server
  q) Quit
Command to execute (1-3, q): 1
-------------------------------------------
PeopleSoft Application Server Administration
-------------------------------------------
  1) Administer a domain
  2) Create a domain
  3) Delete a domain
  4) Import domain configuration
  q) Quit
Command to execute (1-4, q) : 1
Tuxedo domain list:
  1)  HR84
Select domain number to administer: 1
---------------------------------
PeopleSoft Domain Administration
---------------------------------
     Domain Name: HR84
  1) Boot this domain
  2) Domain shutdown menu
  3) Domain status menu
  4) Configure this domain
  5) TUXEDO command line (tmadmin)
  6) Edit configuration/log files menu
  7) Messaging Server Administration menu
```

```
  8) Purge Cache
  9) Preload File Cache
 10) Clean IPC resources of this domain
  q) Quit
Command to execute (1-10, q) : 5
Loading command line administration utility ...
tmadmin - Copyright (c) 1996-1999 BEA Systems, Inc.
Portions * Copyright 1986-1997 RSA Data Security, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
Tuxedo is a registered trademark.
> !env > ./psadmin_env_menu
> quit
```

6 You will now have two files containing environment variables and values. (You will have only one file if you are configuring a process scheduler.) These files should now be consolidated into one file.

Use the appropriate tools or utilities at your disposal to perform the following steps:

- Combine the lines from the two files into one file.
- Sort the lines in the consolidated file to easily identify duplicate lines.
- If two lines are exactly the same (that is, two lines specify the same variable name and value), then delete one of the two lines. If two lines have the same variable but the values are different, then merge the values from the two variables into one line and delete the other line. This process of merging values from two lines usually applies only to path variables (for example, PATH and LIBPATH).

- Finally, be sure to add the “set” and “export” commands appropriate for the shell on each line in your consolidated file. The outputs from running psadmin do not include set or export statements.

7 This step applies only to process schedulers.

If you are configuring an application server, proceed to verify that the newly created environment file is ready to be used with Tuxedo’s tmadmin utility:

See “Testing the Tuxedo environment file” on page 76.

In some PeopleSoft Enterprise configurations, for some unknown reason, the output from executing psadmin –env (refer to step 4) sometimes fails to include the following environment variables:

PS_SERVER_CFG=$PS_HOME/appserv/prcs/*DomainName*/psprcs.cfg

APPDIR=$PS_HOME/appserv/prcs/*DomainName*

PS_SERVDIR=$PS_HOME/appserv/prcs/*DomainName*

These variables are essential to successfully starting a process scheduler and should be added to the environment file you construct for the Tuxedo resource—replacing $PS_HOME and *DomainName* with the appropriate values for your configuration.

Another method that you can use to ensure your process scheduler environment file is comprehensive and accurate is to compare it with a process scheduler log file located in the following directory:

$PS_HOME/appserv/prcs/*DomainName*/log_output/_PSDSTSRVLOG

This log file, which is created when the process scheduler is started, usually contains a list of all the environment variables set prior to starting the Tuxedo server. Be sure to use a log file that reflects a successful startup of the server.

Testing the Tuxedo environment file

To verify that the newly created environment file is ready to be used with Tuxedo’s tmadmin utility, perform the following steps:

To test the Tuxedo environment file

1 Log in as the root user to the system on which the PeopleSoft server to be clustered currently resides.

2 Switch user to the account dedicated to the PeopleSoft Tuxedo server (that is, to the account specified in attribute User). When performing the switch, do not take the user’s environment (that is, exclude the hyphen option of the su command so that the root user’s environment is retained). The environment will be set in the next step.

3 Set the environment by sourcing your newly created environment file.

4 Start the Tuxedo server by executing the command tmboot –y. If a startup errors occur, it is usually due to missing or incorrectly set environment variables. The error messages should indicate which variable is missing or is incorrect. Correct your environment file until tmadmin is able to successfully start the Tuxedo server without errors.

5 Stop the Tuxedo server by executing the command tmshutdown –y.

The following excerpt provides a sample session demonstrating this test.

```
# su psftapp
$ cd /bea/psft8/PT8.49
$ . ./psftappserv.env
$ tmboot -y
Booting all admin and server processes in
/bea/psft8/PT8.49/appserv/HR84/PSTUXCFG
INFO: BEA Tuxedo, Version 9.1, 64-bit, Patch Level 036
INFO: Serial #: 650522264137-2065448083901,
Expiration NONE, Maxusers 1000000
INFO: Licensed to: Oracle-Peoplesoft-ISV
Booting admin processes ...
exec BBL -A :
        process id=23601 ... Started.
Booting server processes ...
exec PSWATCHSRV -A -- -ID 34687 -C psappsrv.cfg
-D TESTSERV -S PSWATCHSRV :
        process id=23620 ... Started.
exec PSAPPSRV -s@../psappsrv.lst -s@../psqcksrv.lst
-sICQuery -sSqlQuery:SqlRequest -- -C psappsrv.c
fg -D TESTSERV -S PSAPPSRV :
        process id=23621 ... Started.
exec PSAPPSRV -s@../psappsrv.lst -s@../psqcksrv.lst
-sICQuery -sSqlQuery:SqlRequest -- -C psappsrv.c
fg -D TESTSERV -S PSAPPSRV :
        process id=23636 ... Started.
exec PSAPPSRV -s@../psappsrv.lst -s@../psqcksrv.lst
-sICQuery -sSqlQuery:SqlRequest -- -C psappsrv.c
fg -D TESTSERV -S PSAPPSRV :
        process id=23692 ... Started.
exec PSSAMSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSSAMSRV :
        process id=23699 ... Started.
exec PSANALYTICSRV -A -- -C psappsrv.cfg
```

```
-D TESTSERV -S PSANALYTICSRV :
        process id=23710 ... Started.
exec PSANALYTICSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSANALYTICSRV :
        process id=23766 ... Started.
exec PSANALYTICSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSANALYTICSRV :
        process id=23785 ... Started.
exec PSRENSRV -A -- -C psappsrv.cfg
-D TESTSERV -S PSRENSRV :
        process id=23796 ... Started.
exec PSMONITORSRV -A -- -ID 34687 -C psappsrv.cfg
-D TESTSERV -S PSMONITORSRV :
        process id=23866 ... Started.
exec WSL -A -- -n //psftapp:7000 -z 0 -Z 0
-d /dev/tcp -I 5 -T 60 -m 1 -M 3 -x 40
-c 5000 -p 7001 -P 7003 :
        process id=23874 ... Started.
exec JSL -A -- -d /dev/tcp -n //psftapp:9000 -m 5
-M 7 -I 5 -j ANY -x 40 -S 10 -c 1000000 -w JSH :
        process id=23876 ... Started.
exec JREPSVR -A -- -W
-P /bea/psft8/PT8.49/appserv/HR84/jrepository :
        process id=23882 ... Started.
14 processes started.
$ tmshutdown -y
Shutting down all admin and server processes in
/bea/psft8/PT8.49/appserv/HR84/PSTUXCFG
Shutting down server processes ...
Server Id = 250 Group Id = JREPGRP Machine = psftapp
:shutdown succeeded
Server Id = 200 Group Id = JSLGRP Machine = psftapp
:shutdown succeeded
Server Id = 20 Group Id = BASE Machine = psftapp
:shutdown succeeded
Server Id = 1 Group Id = MONITOR Machine = psftapp
:shutdown succeeded
Server Id = 101 Group Id = RENGRP Machine = psftapp
:shutdown succeeded
Server Id = 2 Group Id = ANALYTICGRP Machine = psftapp
:shutdown succeeded
Server Id = 3 Group Id = ANALYTICGRP Machine = psftapp
:shutdown succeeded
```

```
Server Id = 1 Group Id = ANALYTICGRP Machine = psftapp
:shutdown succeeded
Server Id = 100 Group Id = APPSRV Machine = psftapp
:shutdown succeeded
Server Id = 2 Group Id = APPSRV Machine = psftapp
:shutdown succeeded
Server Id = 1 Group Id = APPSRV Machine = psftapp
:shutdown succeeded
Server Id = 3 Group Id = APPSRV Machine = psftapp
:shutdown succeeded
Server Id = 1 Group Id = WATCH Machine = psftapp
:shutdown succeeded
Shutting down admin processes ...
Server Id = 0 Group Id = psftapp Machine = psftapp
:shutdown succeeded
14 processes stopped.
```

Assuming the test succeeded, you may now place the newly created environment file somewhere within the Tuxedo directory structure (usually in the directory specified by attribute TuxDir) and set the attribute EnvFile to point to it.

Sample resource definition

The following sample resource definitions configure two PeopleSoft application servers and a process scheduler to run within the same cluster. Note that each resource has a unique, dedicated value for the attributes User, TuxDir, and EnvFile. Also note that the attribute MachineType is set to Single, which is the more common Tuxedo configuration for PeopleSoft servers.

```
Tuxedo PsftAppSrvr1_tux (
  TuxDir = "/psft/app1/tuxedo"
  User = psftapp1
  EnvFile = "/psft/app1/setenv_psadmin.sh"
  MachineName = machine1
  MachineType = Single
  TmbootOptNonMaster = "-y"
  TmshutdownOptNonMaster = "-y"
  TuxDomain = psftapp1
  TuxServers = "PSAPPSRV,WSL"
  )
Tuxedo PsftAppSrvr2_tux (
  TuxDir = "/psft/app2/tuxedo"
```

```
  User = psftapp2
  EnvFile = "/psft/app2/setenv_psadmin.sh"
  MachineName = machine2
  MachineType = Single
  TmbootOptNonMaster = "-y"
  TmshutdownOptNonMaster = "-y"
  TuxDomain = psftapp2
  TuxServers = "PSAPPSRV,WSL"
  )
Tuxedo PsftScheduler1_tux (
  TuxDir = "/psft/sched1/tuxedo"
  User = psftsch1
  EnvFile = "/psft/sched1/tuxenv"
  MachineName = sps-sun3
  MachineType = Single
  TmbootOptNonMaster = "-y"
  TmshutdownOptNonMaster = "-y"
  TuxDomain = PSUNX
  TuxServers = PSPRCSRV
  )
```

Obtaining values for attributes MachineName and TuxDomain

The machine and domain names for a PeopleSoft Tuxedo server instance can be identified in a variety of ways. One of the simplest methods is to search or view the text version of the Tuxedo server's configuration file.

The following is a sample session that performs the search for an application server instance (TuxDomain is psftapp1; MachineName is machine1).

```
# grep DOMAINID /psft/app1/appserv/psftapp1/psappsrv.ubb
DOMAINID psftapp1
# grep LMID /psft/app1/appserv/psftapp1/psappsrv.ubb
"machine1" LMID="machine1" # Machine name must be upper case
```

The following is another sample session that performs the search for a process scheduler instance (TuxDomain is PSUNX; MachineName is sps-sun3).

```
# find /psft/sched1 -name *.ubb
/psft/sched1/appserv/prcs/PSFT/psprcsrv.ubb
/psft/sched1/appserv/prcs/PSFT/backup/psprcsrv.ubb
# grep DOMAINID /psft/sched1/appserv/prcs/PSFT/psprcsrv.ubb
```

```
DOMAINID PSUNX
# grep LMID /psft/sched1/appserv/prcs/PSFT/psprcsrv.ubb
"sps-sun3" LMID="sps-sun3" # Machine name must be upper case
```

Additional PeopleSoft configuration notes

Incorporate the following instructions when using psadmin to create and configure PeopleSoft application servers and process schedulers.

- Use all upper-case letters to set a process scheduler's database connection variable (DBNAME).
- When setting the network address variables for the various listeners (for example, Workstation and Jolt), be sure to use the virtual IP host name assigned to the server instead of the IP address.
- For PeopleSoft application servers, ensure that the variable UseLocalOracleDB is set to 0 if you use Oracle as your database.
- PeopleSoft process schedulers must be configured as "Maintained by Tuxedo".

The following is an excerpt from a session in which a PeopleSoft application server is created. It depicts how some of the server variables are set.

```
[/home/psftapp1] ./psadmin
PSADMIN -- Tools Release: 8.49
Copyright (c) 1988, 2003, Oracle. All rights reserved.
---------------------------------
PeopleSoft Server Administration
---------------------------------
1) Application Server
2) Process Scheduler
3) Web Components
q) Quit
Command to execute (1-3, q): 1
-------------------------------------------
PeopleSoft Application Server Administration
-------------------------------------------
1) Administer a domain
2) Create a domain
3) Delete a domain
q) Quit
Command to execute (1-3, q) : 2
Please enter name of domain to create :test
Configuration templates:
1) developer
```

```
2) large
3) medium
4) small
Select config template number: 4
Creating domain...
Copying application server configuration files...
Copying Jolt repository file...
Copying Java Query client files...
Domain created.
Would you like to configure this domain now? (y/n) [y] :y
-----------------------------------------------
Quick-configure menu -- domain: test
-----------------------------------------------
Features Settings
========== ==========
1) Pub/Sub Servers: No 8) DBNAME :[test]
2) Quick Servers : No 9) DBTYPE :[]
3) Query Servers : No 10) OPRID :[PTDMO]
4) Jolt : Yes 11) OPRPSWD :[PTDMO]
5) Jolt Relay : No 12) DomainID :[PT81]
13) AddToPATH :[.]
14) ConnectID :[people]
15) ConnectPswd:[peop1e]
16) ServerName :[]
Actions 17) WSL Port :[7000]
========= 18) JSL Port :[9000]
6) Load config as shown 19) JRAD Port :[9100]
7) Custom configuration
h) Help for this menu
q) Return to previous menu
HINT: Enter 8 to edit DBNAME, then 6 to load
Enter selection (1-19, h, or q):
[End of excerpt]
```

Registering long PeopleSoft service names

The names of PeopleSoft program services that are deployed using BEA Tuxedo are displayed up to the first 12 characters only. Longer names are truncated to the first 11 characters and end with a plus '+' sign.

The following sample tmadmin psc output excerpt from a PeopleSoft HRMS deployment illustrates this behavior.

```
GetAppImages GetAppImages PSAPPSRV   APPSRV     3    psftapp     0 AVAIL
PsmGetBOEIn+ PsmGetBOEIn+ PSAPPSRV   APPSRV     1    psftapp     0 AVAIL
PsmChkUserR+ PsmChkUserR+ PSAPPSRV   APPSRV     1    psftapp     0 AVAIL
PsmChkRptAu+ PsmChkRptAu+ PSAPPSRV   APPSRV     1    psftapp     0 AVAIL
```

The actual services deployed using the UBBCONFIG psappsrv.ubb file are PsmGetBOEInfo, PsmChkUserRole, and PsmChkRptAuth respectively.

When registering such services with the agent using the TuxServices attribute, ensure that you use only the first 11 characters of the service name (without the '+' sign). In the illustrated sample, the value of TuxServices must be set to "PsmGetBOEIn, PsmChkUserR, PsmChkRptAu".

For more information on the "TuxServices" attribute:

See Table 3-2 on page 34.

Chapter 6

Troubleshooting the agent for BEA Tuxedo

This chapter includes the following topics:

- Using correct software and operating system versions
- Meeting prerequisites
- Configuring BEA Tuxedo resources
- Verifying virtualization
- Starting the BEA Tuxedo instance outside a cluster
- Reviewing error log files

Using 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 BEA Tuxedo, double check that you meet the prerequisites.

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

See "Before you install the Veritas agent for BEA Tuxedo" on page 21.

Configuring BEA Tuxedo resources

Before using a BEA Tuxedo resource, ensure that you configure the resource properly. For a list of attributes used to configure all BEA Tuxedo resources, refer to the agent attributes.

Verifying virtualization

Verify that your application does not use anything that ties it down to a particular node of the cluster.

See "Virtualizing BEA Tuxedo " on page 61.

Starting the BEA Tuxedo 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 BEA Tuxedo instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the BEA Tuxedo 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 Tuxedo instance outside the cluster framework, is illustrated as follows.

Note that all the following examples in this section assume that the prerequisite environment has been sourced through the environment file, declared through the EnvFile attribute.

When MachineType is Single (SHM mode), NonMaster (MP mode), or Backup (MP mode), the commands to start or stop the Tuxedo instance outside the cluster are as follows:

- To start the Tuxedo instance outside the cluster:

  `$ TuxDir/bin/tmboot TmbootOptNonMaster`

- To stop the Tuxedo instance outside the cluster:

  `$ TuxDir/bin/tmshutdown TmshutdownOptNonMaster`

For starting the active master Tuxedo machine (MP mode), the commands are as follows:

- To start the Tuxedo instance outside the cluster:

  ```
  $ TuxDir/bin/tmboot TmbootOptMaster
  ```

- To stop the Tuxedo instance outside the cluster (first, migrate the DBBL to the backup Tuxedo machine, then execute the command):

  ```
  $ TuxDir/bin/tmshutdown TmshutdownOptNonMaster
  ```

  This command may fail if the migration was not successful.

To monitor the Tuxedo instance

1. First, verify if the Tuxedo processes are running as the user defined through the attribute User for all servers defined through the TuxServers attribute.
2. The agent then uses the monitor command:

   ```
   $ TuxDir/bin/tmadmin -r << EOF
   psr -m MachineName
   psc -m MachineName
   quit
   EOF
   ```

   It then attempts to locate BBL and at least one instance of the servers defined using the TuxServers attribute, and at least one instance of the services defined using the TuxServices attribute, in the command output, in order to verify that the Tuxedo machine is available.

   The command is executed in the context of the user defined through the attribute User. Execute this command manually and check if the Tuxedo machine is available. In the MP mode, execute the above command on the machine where the active master is online, using rsh or ssh.

Reviewing error log files

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

The common reasons for issues are as follows:

Insufficient privileges	Files that need to be created, written to, would be created as User. Check if necessary privileges have been set.
Incorrect port, environment or parameter settings	Verify that ports have been properly configured and declared. Typically, ports from 1 through 1024 are reserved for the superuser. Also ensure that parameters to the agent are correctly defined.

Expired licenses	Check the application log files for any error messages related to expired licenses. Ensure the license keys/files have been placed at the appropriate location, as needed by the application.
Broken symlinks, missing files, and libraries	Verify your installation. Make sure nothing is broken, and all dependencies for the executables are met.
Insufficient disk space or system parameters	Ensure that the file-system has sufficient space for creation of temporary files that the application might need. Verify that the kernel has been tuned for sufficient IPC resources, file descriptors and meets the hardware requirement. Consult your product documentation for these details.

Consult your application expert if needed.

Using trace level logging

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

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

Note: Starting with version 5.1.1.0 of the ACC library, the TRACE level logs for any ACCLib based agent are generated locally at the location /var/VRTSvcs/log/*Agent*_A.log.

To localize ResLogLevel attribute for a resource

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

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

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

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

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

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

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

6 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.

Appendix A

Sample Configurations

This appendix includes the following topics:

- About sample configurations for the agent for BEA Tuxedo
- Sample configuration file (SHM model)
- Sample configuration file for PSFT (SHM mode)
- Sample configuration file (MP model)
- Sample agent type definition (VCS 4.x)
- Sample agent type definition (VCS 5.0)
- Sample agent type definition (VCS 5.0 under Solaris 10 zone)
- Sample configuration for Solaris zone support

About sample configurations for the agent for BEA Tuxedo

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 BEA Tuxedo. For more information about these resource types, see the *Veritas Cluster Server Bundled Agents Reference Guide*.

Sample configuration file (SHM model)

The following section from the VCS main.cf file is a service group definition for a Tuxedo machine running as a stand-alone machine where the application model is SHM:

```
include "types.cf"
include "TuxedoTypes.cf"
cluster tux91sol (
   UserNames = { admin = aHIaHChEIdIIgQIcHF }
   Administrators = { admin }
   CredRenewFrequency = 0
   CounterInterval = 5
   )
system nodeA (
   )
system nodeB (
   )
group tux91shm (
   SystemList = { nodeA = 0, nodeB = 1 }
   )
DiskGroup tux91_dg (
   DiskGroup = tux91dg
   )
NIC tux91_nic (
   Device = bge0
   )
IP tux91_ip (
   Device = bge0
   Address = "10.212.98.188"
   NetMask = "255.255.254.0"
   )
Mount tux91_mnt (
   MountPoint = "/bea/tux"
   BlockDevice = "/dev/vx/dsk/tux91dg/tuxedo"
   FSType = vxfs
   FsckOpt = "-y"
   )
Tuxedo tux91srvr (
   EnvFile = "/bea/tux/tuxedo9.1/tux.env"
   MachineName = tux91sol
   TmbootOptNonMaster = "-y"
   TmshutdownOptNonMaster = "-y"
   AppDir = "/bea/tux/tuxedo9.1/app/simpapp"
   TuxDir = "/bea/tux/tuxedo9.1"
   TuxDomain = simpapp
   TuxServers = simpserv
   TuxServices = TOUPPER
   User = tux91adm
```

```
   SecondLevelMonitor = 3
   )
tux91_mnt requires tux91_dg
tux91_ip requires tux91_nic
tux91srvr requires tux91_ip
tux91srvr requires tux91_mnt

// resource dependency tree
//
// group tux91
// {
//    Tuxedo tux91srvr
//    {
//    Mount tux91_mnt
//       {
//       DiskGroup tux91_dg
//       }
//    IP tux91_ip
//    }
// }
```

Sample configuration file for PSFT (SHM mode)

The following section from the VCS main.cf file is a service group definition for a PeopleSoft HRMS deployment, making use of a Tuxedo machine under SHM mode.

```
include "types.cf"
include "TuxedoTypes.cf"

cluster PSFTHRMS (
        UserNames = { admin = aHIaHChEIdIIgQIcHF }
 Administrators = { admin }
 CredRenewFrequency = 0
        CounterInterval = 5
 )

system nodeA (
 )

system nodeB (
 )
```

```
group psftapp (
 SystemList = { nodeA = 0, nodeB = 1 }
 )

 DiskGroup psftapp_dg (
  Critical = 0
  DiskGroup = psftapp
  )

 DiskGroup psfttux_dg (
  Critical = 0
  DiskGroup = psft8tux
  )

 IP psftapp_ip (
  Critical = 0
  Device = bge0
  Address = "10.212.98.122"
  NetMask = "255.255.254.0"
  )

 Mount psftapp_mnt (
  MountPoint = "/bea/psft8/PT8.49"
  BlockDevice = "/dev/vx/dsk/psftapp/mobagent"
  FSType = vxfs
  FsckOpt = "-y"
  SecondLevelMonitor = 1
  )

 Mount psfttux_mnt (
  MountPoint = "/bea/psft8/tux"
  BlockDevice = "/dev/vx/dsk/psft8tux/tuxedo"
  FSType = vxfs
  FsckOpt = "-y"
  SecondLevelMonitor = 1
  )

 Tuxedo psft_appsrvr (
  EnvFile = "/bea/psft8/PT8.49/psftappserv.env"
  MachineName = psftapp
  TmbootOptNonMaster = "-y"
  TmshutdownOptNonMaster = "-y"
```

```
 AppDir = "/bea/psft8/PT8.49/appserv/HR84"
 TuxDir = "/bea/psft8/tux/tuxedo91"
 TuxDomain = TESTSERV_50645
 TuxServers = "PSMONITORSRV, PSANALYTICSRV, PSAPPSRV,
 PSWATCHSRV, WSL, PSSAMSRV, PSRENSRV, JREPSVR, JSL"
 TuxServices = "SqlQuery, SqlRequest, Publish, RemoteCall"
 User = psftapp
 SecondLevelMonitor = 5
 )

psft_appsrvr requires psftapp_ip
psft_appsrvr requires psftapp_mnt
psft_appsrvr requires psfttux_mnt
psftapp_mnt requires psftapp_dg
psfttux_mnt requires psfttux_dg


// resource dependency tree
//
// group psftapp
// {
// Tuxedo psft_appsrvr
//     {
//     IP psftapp_ip
//     Mount psftapp_mnt
//         {
//         DiskGroup psftapp_dg
//         }
//     Mount psfttux_mnt
//         {
//         DiskGroup psfttux_dg
//         }
//     }
// }
```

Sample configuration file (MP model)

The following section from the VCS main.cf file are service group definitions for four Tuxedo machines that are part of the same multiple machine domain where the application model is MP. tux91sol is the master machine, tux91bksol is the backup machine, and tux91nm1sol and tux91nm2sol are the non-master machines.

```
include "types.cf"
include "TuxedoTypes.cf"
cluster tux91sol (
   UserNames = { admin = aHIaHChEIdIIgQIcHF }
   Administrators = { admin }
   CredRenewFrequency = 0
   CounterInterval = 5
   )
system nodeA (
   )
system nodeB (
   )
group tux91 (
   SystemList = { nodeA = 0, nodeB = 1 }
   )
   Application tux91lsnr (
      User = tux91adm
      StartProgram = "/bea/tux/tuxedo9.1/bin/startlsnr"
      StopProgram = "/bea/tux/tuxedo9.1/bin/stoplsnr"
      CleanProgram = "/bea/tux/tuxedo9.1/bin/stoplsnr"
      MonitorProcesses = {
      "/bea/tux/tuxedo9.1/bin/tlisten -l //tux91sol:22222" }
      )
   DiskGroup tux91_dg (
      DiskGroup = tux91dg
      )
   NIC tux91_nic (
      Device = bge0
      )
   IP tux91_ip (
      Device = bge0
      Address = "10.212.98.188"
      NetMask = "255.255.254.0"
      )
   Mount tux91_mnt (
      MountPoint = "/bea/tux"
      BlockDevice = "/dev/vx/dsk/tux91dg/tuxedo"
      FSType = vxfs
      FsckOpt = "-y"
      )
   Tuxedo tux91srvr (
      EnvFile = "/bea/tux/tuxedo9.1/tux.env"
      MachineName = tux91sol
```

```
        MachineType = Master
        RemoteShell = "/usr/bin/ssh"
        TmbootOptMaster = "-Ml tux91sol"
        TmbootOptNonMaster = "-y"
        TmshutdownOptMaster = "-Ml tux91sol"
        TmshutdownOptNonMaster = "-y"
        AppDir = "/bea/tux/tuxedo9.1/app/simpapp"
        TuxDir = "/bea/tux/tuxedo9.1"
        TuxDomain = simpapp
        TuxServers = simpserv
        TuxServices = TOUPPER
        User = "+tux91adm"
        SecondLevelMonitor = 1
        )
tux91_mnt requires tux91_dg
tux91_ip requires tux91_nic
tux91lsnr requires tux91_ip
tux91lsnr requires tux91_mnt
tux91srvr requires tux91lsnr

// resource dependency tree
//
// group tux91
//    {
//    Tuxedo tux91srvr
//       {
//       Application tux91lsnr
//          {
//          Mount tux91_mnt
//             {
//             DiskGroup tux91_dg
//             }
//          IP tux91_ip
//          }
//       }
//    }
group tux91bkp (
   SystemList = { nodeA = 0, nodeB = 1 }
   )
   Application tux91bkplsnr (
      User = tux91adm
      StartProgram = "/bea/tuxbk/tuxedo9.1/bin/startlsnr"
      StopProgram = "/bea/tuxbk/tuxedo9.1/bin/stoplsnr"
```

```
        CleanProgram = "/bea/tuxbk/tuxedo9.1/bin/stoplsnr"
        MonitorProcesses = {
        "/bea/tuxbk/tuxedo9.1/bin/tlisten -l
        //tux91bksol:22222" }
        )
    DiskGroup tux91bkp_dg (
        DiskGroup = tux91bkpdg
        )
    NIC tux91bkp_nic (
        Device = bge1
        )
    IP tux91bkp_ip (
        Device = bge1
        Address = "10.212.98.187"
        NetMask = "255.255.254.0"
        )
    Mount tux91bkp_mnt (
        MountPoint = "/bea/tuxbk"
        BlockDevice = "/dev/vx/dsk/tux91bkpdg/tuxedo"
        FSType = vxfs
        FsckOpt = "-y"
        )
    Tuxedo tux91bkpsrvr (
        EnvFile = "/bea/tuxbk/tuxedo9.1/tux.env"
        MachineName = tux91bksol
        MachineType = Backup
        RemoteShell = "/usr/bin/ssh"
        TmbootOptMaster = "-Ml tux91bksol"
        TmbootOptNonMaster = "-B tux91bksol -l tux91bksol"
        TmshutdownOptMaster = "-Ml tux91bksol"
        TmshutdownOptNonMaster = "-B tux91bksol -l tux91bksol"
        AppDir = "/bea/tuxbk/tuxedo9.1/app/simpapp"
        TuxDir = "/bea/tuxbk/tuxedo9.1"
        TuxDomain = simpapp
        TuxServers = simpservbk
        TuxServices = TOUPPER
        User = "+tux91adm"
        SecondLevelMonitor = 1
        )
tux91bkp_mnt requires tux91bkp_dg
tux91bkp_ip requires tux91bkp_nic
tux91bkplsnr requires tux91bkp_ip
tux91bkplsnr requires tux91bkp_mnt
```

```
tux91bkpsrvr requires tux91bkplsnr

// resource dependency tree
//
// group tux91bkp
// {
// Tuxedo tux91bkpsrvr
//    {
//    Application tux91bkplsnr
//       {
//       IP tux91bkp_ip
//       Mount tux91bkp_mnt
//          {
//          DiskGroup tux91bkp_dg
//          }
//       }
//    }
// }
group tux91nm1 (
   SystemList = { nodeA = 0, nodeB = 1 }
   )
   Application tux91nm1lsnr (
      User = tuxadm2
      StartProgram = "/bea/tuxm2/tuxedo9.1/bin/startlsnr"
      StopProgram = "/bea/tuxm2/tuxedo9.1/bin/stoplsnr"
      CleanProgram = "/bea/tuxm2/tuxedo9.1/bin/stoplsnr"
      MonitorProcesses = {
      "/bea/tuxm2/tuxedo9.1/bin/tlisten -l
      //tux91nm1sol:22222" }
      )
   DiskGroup tux91nm1_dg (
      DiskGroup = tux91bdg
      )
   NIC tux91nm1_nic (
      Device = bge2
      )
   IP tux91nm1_ip (
      Device = bge2
      Address = "10.212.98.189"
      NetMask = "255.255.254.0"
      )
   Mount tux91nm1_mnt (
      MountPoint = "/bea/tuxm2"
```

```
        BlockDevice = "/dev/vx/dsk/tux91bdg/tuxedo"
        FSType = vxfs
        FsckOpt = "-y"
        )
    Tuxedo tux91nm1srvr (
        EnvFile = "/bea/tuxm2/tuxedo9.1/tux.env"
        MachineName = tux91nm1sol
        MachineType = NonMaster
        RemoteShell = "/usr/bin/ssh"
        TmbootOptNonMaster = "-B tux91nm1sol -l tux91nm1sol"
        TmshutdownOptNonMaster = "-B tux91nm1sol -l tux91nm1sol"
        AppDir = "/bea/tuxm2/tuxedo9.1/app/simpapp"
        TuxDir = "/bea/tuxm2/tuxedo9.1"
        TuxDomain = simpapp
        TuxServers = simpservnm1
        TuxServices = TOUPPER
        User = tuxadm2
        SecondLevelMonitor = 1
        )
tux91nm1_mnt requires tux91nm1_dg
tux91nm1_ip requires tux91nm1_nic
tux91nm1lsnr requires tux91nm1_ip
tux91nm1lsnr requires tux91nm1_mnt
tux91nm1srvr requires tux91nm1lsnr

// resource dependency tree
//
// group tux91nm1
// {
// Tuxedo tux91nm1srvr
//     {
//     Application tux91nm1lsnr
//         {
//         IP tux91nm1_ip
//         Mount tux91nm1_mnt
//             {
//             DiskGroup tux91nm1_dg
//             }
//         }
//     }
// }
group tux91nm2 (
    SystemList = { nodeA = 0, nodeB = 1 }
```

```
    )
    Application tux91nm2lsnr (
        User = tux91adm
        StartProgram = "/bea/tuxm3/tuxedo9.1/bin/startlsnr"
        StopProgram = "/bea/tuxm3/tuxedo9.1/bin/stoplsnr"
        CleanProgram = "/bea/tuxm3/tuxedo9.1/bin/stoplsnr"
        MonitorProcesses = {
        "/bea/tuxm3/tuxedo9.1/bin/tlisten -l
        //tux91nm2sol:22222" }
        )
    DiskGroup tux91nm2_dg (
        DiskGroup = tux91nm2dg
        )
    NIC tux91nm2_nic (
        Device = bge3
        )
    IP tux91nm2_ip (
        Device = bge3
        Address = "10.212.98.190"
        NetMask = "255.255.254.0"
        )
    Mount tux91nm2_mnt (
        MountPoint = "/bea/tuxm3"
        BlockDevice = "/dev/vx/dsk/tux91nm2dg/tuxedo"
        FSType = vxfs
        FsckOpt = "-y"
        )
    Tuxedo tux91nm2srvr (
        EnvFile = "/bea/tuxm3/tuxedo9.1/tux.env"
        MachineName = tux91nm2sol
        MachineType = NonMaster
        RemoteShell = "/usr/bin/ssh"
        TmbootOptNonMaster = "-B tux91nm2sol -l tux91nm2sol"
        TmshutdownOptNonMaster = "-B tux91nm2sol -l tux91nm2sol"
        AppDir = "/bea/tuxm3/tuxedo9.1/app/simpapp"
        TuxDir = "/bea/tuxm3/tuxedo9.1"
        TuxDomain = simpapp
        TuxServers = simpservnm2
        TuxServices = TOUPPER
        User = "+tux91adm"
        SecondLevelMonitor = 5
        )
tux91nm2_mnt requires tux91nm2_dg
```

```
tux91nm2_ip requires tux91nm2_nic
tux91nm2lsnr requires tux91nm2_ip
tux91nm2lsnr requires tux91nm2_mnt
tux91nm2srvr requires tux91nm2lsnr

// resource dependency tree
//
// group tux91nm2
// {
// Tuxedo tux91nm2srvr
//    {
//    Application tux91nm2lsnr
//       {
//       Mount tux91nm2_mnt
//          {
//          DiskGroup tux91nm2_dg
//          }
//       IP tux91nm2_ip
//       }
//    }
// }
```

Sample agent type definition (VCS 4.x)

The sample configuration file for VCS 4.x is as follows:

```
type Tuxedo (
    static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
    MachineName, MachineType, RemoteShell, TmbootOptMaster,
    TmbootOptNonMaster, TmshutdownOptMaster, TmshutdownOptNonMaster,
    AppDir, TuxDir, TuxDomain, TuxServers, TuxServices, User,
    SecondLevelMonitor, MonitorProgram }
    str ResLogLevel = INFO
    str EnvFile
    str MachineName
    str MachineType = Single
    str RemoteShell
    str TmbootOptMaster
    str TmbootOptNonMaster = "-y"
    str TmshutdownOptMaster
    str TmshutdownOptNonMaster = "-y"
    str AppDir
    str TuxDir
```

```
    str TuxDomain
    str TuxServers
    str TuxServices
    str User
    int SecondLevelMonitor = 0
    str MonitorProgram
)
```

Sample agent type definition (VCS 5.0)

The sample configuration file for VCS 5.0 is as follows:

```
type Tuxedo (
   static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
   static str AgentDirectory = "/opt/VRTSagents/ha/bin/Tuxedo"
   static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
   MachineName, MachineType, RemoteShell, TmbootOptMaster,
   TmbootOptNonMaster, TmshutdownOptMaster, TmshutdownOptNonMaster,
   AppDir, TuxDir, TuxDomain, TuxServers, TuxServices, User,
   SecondLevelMonitor, MonitorProgram }
   str ResLogLevel = INFO
   str EnvFile
   str MachineName
   str MachineType = Single
   str RemoteShell
   str TmbootOptMaster
   str TmbootOptNonMaster = "-y"
   str TmshutdownOptMaster
   str TmshutdownOptNonMaster = "-y"
   str AppDir
   str TuxDir
   str TuxDomain
   str TuxServers
   str TuxServices
   str User
   int SecondLevelMonitor = 0
   str MonitorProgram
)
```

Sample agent type definition (VCS 5.0 under Solaris 10 zone)

The sample agent type definition for Solaris zone support, for Tuxedo is as follows:

```
type Tuxedo (
 static str ContainerType = Zone
 static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/Tuxedo"
 static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
 MachineName, MachineType, RemoteShell, TmbootOptMaster,
 TmbootOptNonMaster, TmshutdownOptMaster, TmshutdownOptNonMaster,
 AppDir, TuxDir, TuxDomain, TuxServers,
 TuxServices, User, SecondLevelMonitor,
 MonitorProgram }
 str ResLogLevel = INFO
 str EnvFile
 str MachineName
 str MachineType = Single
 str RemoteShell
 str TmbootOptMaster
 str TmbootOptNonMaster = "-y"
 str TmshutdownOptMaster
 str TmshutdownOptNonMaster = "-y"
 str AppDir
 str TuxDir
 str TuxDomain
 str TuxServers
 str TuxServices
 str User
 int SecondLevelMonitor = 0
 str MonitorProgram
 str ContainerName
)
```

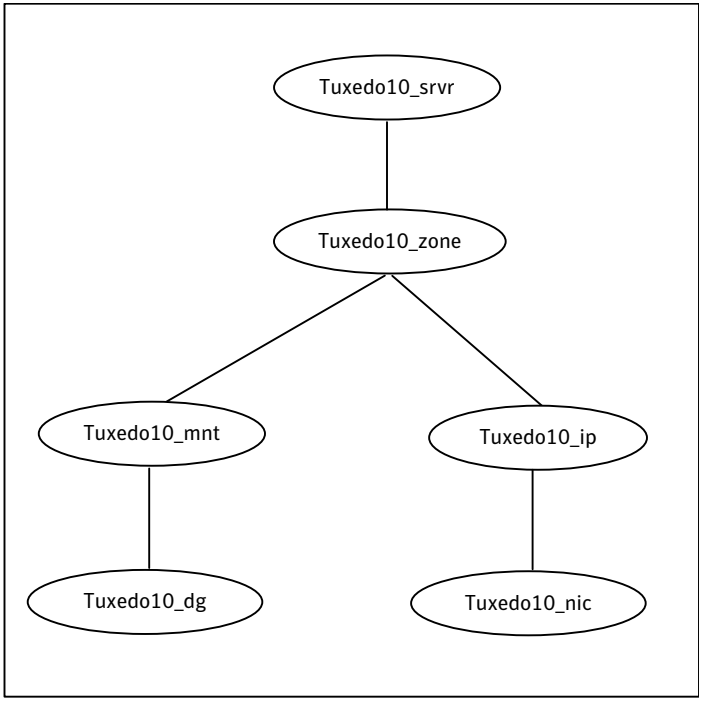
Sample configuration for Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure A-1 shows a service group with a Tuxedo instance running in a local zone, if the zone binaries are present on the local disk.

The service group also includes a DiskGroup resource, a NIC resource, an IP resource and a Mount resource.

Figure A-1 Sample service group with a Tuxedo instance configured under Solaris 10 zones



Index

A

about ACC library 22
about BEA Tuxedo 13
    See also about BEA Tuxedo server; BEA Tuxedo, overview
about BEA Tuxedo server 13
    See also about BEA Tuxedo; BEA Tuxedo, overview
ACC library
    installing 22
    removing 25
agent
    features 12
    importing agent types files 27
    installing, VCS environment 23
    optional attributes 33
    overview 11
    required attributes 28
    supported software 12
    uninstalling, VCS environment 24
    upgrading 25
    what's new 12
agent attributes
    AppDir 29
    EnvFile 29
    MachineName 29
    MachineType 30
    MonitorProgram 34
    RemoteShell 30
    ResLogLevel 34
    SecondLevelMonitor 35
    TmbootOptMaster 31
    TmbootOptNonMaster 35
    TmshutdownOptMaster 32
    TmshutdownOptNonMaster 36
    TuxDir 32
    TuxDomain 33
    TuxServers 36
    TuxServices 36
    User 33
agent configuration file
    importing 27
agent functions
    configuring monitor function. *See* executing custom monitor program
agent installation
    general requirements 21
    requirements for Solaris zones 22
    steps to install 23
agent operations
    clean 18
    monitor 16
    offline 15
    online 14

B

BEA Tuxedo
    additional requirements for MP model 52
    clustering 45
    configuration using ULGPFX 50
    configuring resources 86
    configuring resources for Solaris zones 37
    creating environment file 64
    directory structure 44
    installing on shared disks 43
    MP model 41
    SHM model 41
    starting instance outside cluster 86
    testing environment file 76
    virtualization 61
        Host names 61
        Path names 61
BEA Tuxedo machine
    clustering
        basic resources 60
BEA Tuxedo, overview 39
    See also about BEA Tuxedo; about BEA Tuxedo server
BEA Tuxedo, shared disk installation 43

C

clustering BEA Tuxedo machine
    basic resources 60
clustering BEA Tuxedo machines 45
configuration file, MP model 95
configuration file, SHM model 91
configuring monitor function 37
creating BEA Tuxedo environment file
    using ps command 64
    using psadmin utility 68

E

executing custom monitor program 37

L

logs
    reviewing error log files 87
    using trace level logging 88

M

MP model
    clustering BEA Tuxedo listeners 57
    configuring OpenSSH 55
    configuring Remote Shell 55
    configuring ToleranceLimit 57
    sample configuration file 95
    tuning MonitorTimout value 56

P

PeopleSoft
    about 63
    configuration notes 81
    obtaining values for MachineName 80
    obtaining values for TuxDomain 80
    registering long service names 82
    sample configuration file 93
    sample resource definition 79
PeopleSoft, overview 63

R

removing agent, VCS environment 24

S

sample agent type definition
    Tuxedo, Solaris zone support 104
sample agent type definition, VCS 4.x 102
sample agent type definition, VCS 5.0 103
service group
    sample configurations, Solaris zone support 104
service group dependencies, MP model 55
SHM model
    sample configuration file 91
Solaris zone support
    configuring BEA Tuxedoresources 37
    installation requirements 22
    sample agent type definition 104
    Tuxedo, sample service group
        configurations 104
starting the BEA Tuxedo instance outside a cluster 86
supported high availability scenarios 53
supported software 12

T

troubleshooting
    meeting prerequisites 85
    reviewing error log files 87
        using trace level logging 88
    using correct software 85
    verifying virtualization 86
Tuxedo
    sample service group configurations, Solaris
        zone support 104

U

uninstalling agent, VCS environment 24
upgrading agent 25