

Cluster Server Agent for Oracle Tuxedo Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

5.0

Cluster Server Agent for Oracle 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

Document version: 5.0 Rev 0

Legal Notice

Copyright © 2015 Symantec Corporation. All rights reserved.

Symantec, the Symantec Logo, the Checkmark Logo, Veritas, the Veritas Logo, InfoScale, CommandCentral, NetBackup, Enterprise Vault, and LiveUpdate are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, "Rights in Commercial Computer Software or Commercial Computer Software Documentation", as applicable, and any successor regulations. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

<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 support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

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

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

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

Contacting Technical Support

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

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

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

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

- Product release level
- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
 - Error messages and log files
 - Troubleshooting that was performed before contacting Symantec
 - Recent software configuration changes and network changes

Licensing and registration

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

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

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

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

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

Asia-Pacific and Japan customercare_apj@symantec.com

Europe, Middle-East, and Africa semea@symantec.com

North America and Latin America [supportolutions@symantec.com](mailto:supportsolutions@symantec.com)

Documentation

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

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

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

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

<https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation>

About Symantec Connect

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

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

Contents

Technical Support	3	
Chapter 1	Introducing the agent for Oracle Tuxedo	9
	About the Cluster Server agent for Oracle Tuxedo	9
	Supported software	10
	Features of the agent	10
	How the agent makes Oracle Tuxedo highly available	10
	High availability for Oracle Tuxedo instances running in Solaris zones	11
	Oracle Tuxedo agent operations	11
	Online	11
	Offline	12
	Monitor	13
	Clean	15
	Setting up Oracle Tuxedo in a VCS cluster	16
Chapter 2	Installing and configuring Oracle Tuxedo for high availability	17
	About Oracle Tuxedo	17
	About the Oracle Tuxedo server	18
	Virtualizing Oracle Tuxedo	18
	Clustering Oracle Tuxedo	19
	Overview of Oracle Tuxedo	19
	Installing Oracle Tuxedo software and applications on shared disks	22
	Basic steps to cluster an Oracle Tuxedo machine	25
	Configuring Oracle Tuxedo using the ULOGPFX attribute	30
	Additional requirements for multiple machine domains (MP model)	31
	Oracle Tuxedo entities in a clustered environment	45
	Clustering PeopleSoft Enterprise Servers	46
	Overview of PeopleSoft	46
	Creating the Oracle Tuxedo environment file	46
	Testing the Tuxedo environment file	60
	Sample resource definition	63

	Obtaining values for attributes MachineName and TuxDomain	64
	Additional PeopleSoft configuration notes	65
	Registering long PeopleSoft service names	66
Chapter 3	Installing, upgrading, and removing the agent for Oracle Tuxedo	68
	Before you install the Cluster Server agent for Oracle Tuxedo	68
	Prerequisites for enabling i18n support	69
	Prerequisites for installing the agent to support Solaris zones	69
	About the ACC library	70
	Installing the ACC library	70
	Installing the ACC library IPS package on Oracle Solaris 11 systems	71
	Installing the ACC library package on Solaris brand non-global zones	72
	Installing the agent in a VCS environment	73
	Installing the agent IPS package on Oracle Solaris 11 systems	74
	Installing agent packages on Solaris brand non-global zones	75
	Uninstalling the agent in a VCS environment	76
	Removing the ACC library	77
	Upgrading the agent in a VCS environment	78
Chapter 4	Configuring the agent for Oracle Tuxedo	80
	About configuring the Cluster Server agent for Oracle Tuxedo	80
	Importing the agent types files in a VCS environment	80
	Agent attributes	82
	Executing a customized monitoring program	90
Chapter 5	Configuring the service groups for Oracle Tuxedo using the CLI	91
	About configuring service groups for Oracle Tuxedo	91
	Before configuring the service groups for Oracle Tuxedo	92
	Configuring Oracle Tuxedo resources for Solaris zones support	92

Chapter 6	Troubleshooting the agent for Oracle Tuxedo	93
	Using the correct software and operating system versions	93
	Meeting prerequisites	93
	Configuring Oracle Tuxedo resources	94
	Verifying virtualization	94
	Starting the Oracle Tuxedo instance outside a cluster	94
	Reviewing error log files	95
	Using trace level logging	96
Appendix A	Sample Configurations	98
	About sample configurations for the agents for Oracle Tuxedo	98
	Sample agent type definition for Oracle Tuxedo	98
	For VCS 4.x	98
	For VCS 5.x	99
	For VCS 5.0 under Solaris zones	100
	For VCS 5.1 under Solaris zones	101
	Sample configuration file for Oracle Tuxedo	101
	For SHM model	101
	For PSFT (SHM mode)	103
	For MP model	105
	Sample configuration for Solaris zone support	112
Index	114

Introducing the agent for Oracle Tuxedo

This chapter includes the following topics:

- [About the Cluster Server agent for Oracle Tuxedo](#)
- [Supported software](#)
- [Features of the agent](#)
- [How the agent makes Oracle Tuxedo highly available](#)
- [Oracle Tuxedo agent operations](#)
- [Setting up Oracle Tuxedo in a VCS cluster](#)

About the Cluster Server agent for Oracle Tuxedo

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

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

See the Agent Pack Release Notes for the latest updates or software issues for this agent.

Supported software

For information on the software versions that the Cluster Server agent for Oracle Tuxedo supports, see the Symantec Operations Readiness Tools (SORT) site: <https://sort.symantec.com/agents>.

Features of the agent

The following are the features of the Cluster Server agent for Oracle Tuxedo:

- Support for validation of attributes that are based on the agent functions
The agent can validate attributes in each agent function before the actual data processing starts.
- Support for First Failure Data Capture (FFDC)
In case of a fault, the agent generates a huge volume of the debug logs that enable troubleshooting of the fault.
- Support for Fast First Level Monitor (FFLM)
The agent maintains PID files based on search patterns to expedite the monitoring process.
- Support for external user-supplied monitor utilities
The agent enables user-specified monitor utilities to be plugged in, in addition to the built-in monitoring logic. This enables administrators to completely customize the monitoring of the application.
- Delayed agent function
The agent manages the first monitor after online for slow initializing applications.

How the agent makes Oracle Tuxedo highly available

The agent provides the following levels of application monitoring:

- Primary or Basic monitoring
This mode has Process check and Health check monitoring options. With the default Process check option, the agent verifies that the Oracle Tuxedo instance processes are present in the process table. Process check cannot detect whether processes are in the hung or stopped states.
- Secondary or Detail monitoring
In this mode, the agent runs a utility to verify the status of the Oracle Tuxedo instance. The agent detects application failure if the monitoring routine reports an improper function of the Oracle Tuxedo instance processes. When this

application failure occurs, the Oracle Tuxedo instance service group fails over to another node in the cluster.

Thus, the agent ensures high availability for Oracle Tuxedo instances.

High availability for Oracle Tuxedo instances running in Solaris zones

Solaris provides a means of virtualizing operating system services, allowing one or more processes to run in isolation from other activity on the system. Such a 'sandbox' is called a 'non-global zone'. Each zone can provide a rich and customized set of services. The processes that run in a 'global zone' have the same set of privileges that are available on a Solaris system today.

VCS provides high availability to applications running in non-global zones by extending the failover capability to zones. VCS is installed in a global zone, and all the agents and the engine components run in the global zone. For applications running within non-global zones, agents run script entry points inside the zones. If a zone configured under VCS control faults, VCS fails over the entire service group containing the zone.

For more details, refer to the *Cluster Server Administrator's Guide*.

The Cluster Server agent for Oracle Tuxedo is zone-aware and can monitor Tuxedo instances running in non-global zones.

Oracle Tuxedo agent operations

The agent for Oracle 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 Oracle 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 Oracle Tuxedo machine instance.
- Boots the stand-alone Oracle 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 tmbboot 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 tadmin 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 Oracle 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 Oracle Tuxedo machine by remotely executing tmboot on the active master machine and specifying the non-master machine to be booted. Remote execution of tadmin is implemented with either rsh or ssh.

Offline

The offline operation is responsible for stopping a single Oracle Tuxedo machine instance on a node in the cluster. The steps to take an Oracle 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 Oracle 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 Oracle 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 Oracle 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 tadmin and issuing the pclean sub-command for the target machine.

If MachineType is not Single and the Oracle 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 Oracle 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 Oracle 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 Oracle 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 Oracle 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 Oracle Tuxedo machine's bulletin board using `tadmin`. 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 Oracle 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 Oracle 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 `tadmin` 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 Oracle 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 90.

Clean

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

The operation cleans the system by performing the following tasks:

- Identifies and removes remaining processes pertaining to the Oracle Tuxedo machine being clustered.
- Identifies and removes remaining IPC resources pertaining to the Oracle 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 Oracle Tuxedo

highly recommends installing each Oracle 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 Oracle 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 Oracle Tuxedo utility to list the IPC resources that belong to the current Oracle 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 *Oracle Tuxedo Command Reference Guide* available on the Oracle 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 Oracle Tuxedo machine lies with the system administrator. Failure to do so can prevent the Oracle Tuxedo machine from restarting on this cluster node.

Setting up Oracle Tuxedo in a VCS cluster

Follow the steps below to set up Oracle Tuxedo in a cluster:

- Set up a VCS cluster.

For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Install and configure Oracle Tuxedo for High Availability.

See [“About configuring the Cluster Server agent for Oracle Tuxedo”](#) on page 80.
- Install the Cluster Server agent for Oracle Tuxedo.

See [“Installing the agent in a VCS environment”](#) on page 73.
- Configure the service groups for Oracle Tuxedo.

See [“About configuring service groups for Oracle Tuxedo”](#) on page 91.

Installing and configuring Oracle Tuxedo for high availability

This chapter includes the following topics:

- [About Oracle Tuxedo](#)
- [About the Oracle Tuxedo server](#)
- [Virtualizing Oracle Tuxedo](#)
- [Clustering Oracle Tuxedo](#)
- [Clustering PeopleSoft Enterprise Servers](#)

About Oracle Tuxedo

The Oracle 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 Oracle 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 an Oracle Tuxedo application.
- The fundamental underpinnings for creating, managing, and maintaining reliable, high performance, easily managed distributed systems.

About the Oracle Tuxedo server

An Oracle 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.

Oracle 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 Oracle 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.

Virtualizing Oracle Tuxedo

To ensure that your Oracle 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 Oracle 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 Oracle 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.

Clustering Oracle Tuxedo

This section provides excerpts from several Oracle Tuxedo product documents. These documents are available on the Oracle product documentation web site.

Overview of Oracle Tuxedo

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

Oracle Tuxedo provides the framework, or middleware, for building and deploying scalable multi-tier client/server applications across a variety of platforms. Oracle 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 Oracle 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 Oracle Tuxedo system and the applications built with it, it is important to understand the following terms and concepts:

Oracle Tuxedo Domain An Oracle Tuxedo domain, also known as an Oracle Tuxedo application, is a set of Oracle Tuxedo system, client, and server processes administered as a single unit from a single Oracle Tuxedo configuration file. An Oracle 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. An Oracle Tuxedo domain has the same meaning as an Oracle Tuxedo application. An Oracle Tuxedo domain may consist of one Oracle Tuxedo server running on one computer (SHM model) or it may consist of multiple Oracle Tuxedo servers running on multiple computers (MP model).

Oracle Tuxedo Configuration File	Each Oracle 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.
Oracle Tuxedo Installation Directory (TUXDIR)	The TUXDIR environment variable defines the product installation directory of the Oracle Tuxedo software. It must be set to an absolute path name ending with the name of the product installation directory.
Oracle Tuxedo Application Directory (APPDIR)	The absolute path name of the application directory in which application server files are stored and booted for an Oracle Tuxedo machine. APPDIR may be set to more than one application directory.
Oracle Tuxedo Master Machine	The master machine for an Oracle 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 an Oracle Tuxedo domain is done through the master machine. The master machine for an Oracle 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 an Oracle Tuxedo domain whenever the Oracle Tuxedo domain is booted on the master machine.
Oracle Tuxedo Bulletin Board	The Oracle Tuxedo system uses the TUXCONFIG file to set up a bulletin board on each machine in an Oracle Tuxedo domain. When an Oracle 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 Oracle 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 an Oracle Tuxedo domain.
Distinguished Bulletin Board Liaison (DBBL)	DBBL is a process hosted on the Oracle 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.

Oracle 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 Oracle Tuxedo machine on which one or more applications are

deployed. It operates as a stand-alone, independent machine and does not communicate with other Oracle Tuxedo machines.

A multiple machine domain consists of multiple Oracle 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 Oracle Tuxedo agent.

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

Figure 2-1 shows a single machine domain.

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

Figure 2-1 Single machine domain

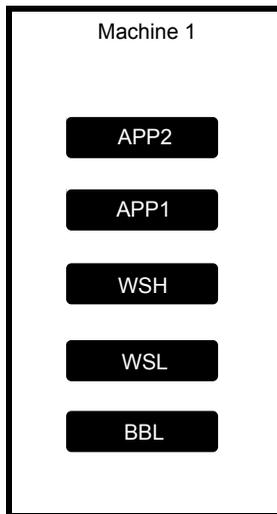
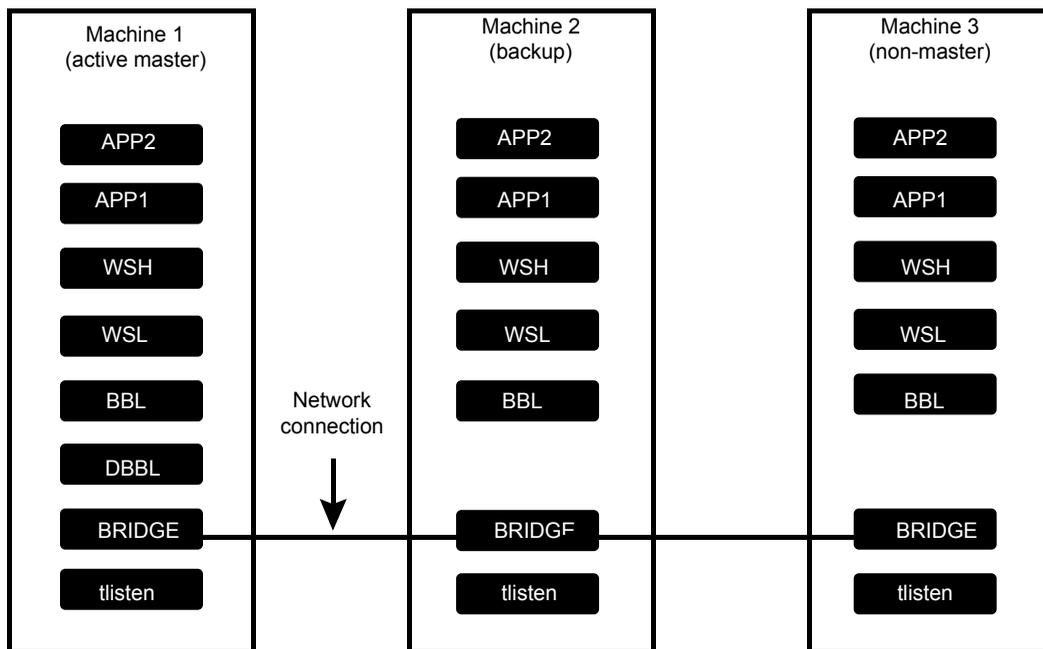


Figure 2-2 shows a multiple machine domain.

In this figure, three Oracle 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 2-2 Multiple machine domain



Installing Oracle Tuxedo software and applications on shared disks

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

In addition, placing the files of each Oracle Tuxedo machine on its own shared disk managed by Veritas Volume Manager also simplifies the process of providing a disaster recovery solution for each Oracle Tuxedo machine. By simply configuring the Oracle Tuxedo machine's volume for replication, the entire contents of the volume (that is, the entire Oracle 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 Oracle 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 Oracle Tuxedo application simplifies the cluster configuration, facilitates disaster recovery, and creates a storage environment that is more intuitive and easier to manage. recommends a directory structure similar to the one described in the following table. This sample directory structure depicts just two Oracle Tuxedo machines, but the naming pattern supports an unlimited number of machines.

[Table 2-1](#) describes the recommended directory structure.

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

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

Directory structure recommendation

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

[Table 2-1](#) describes the recommended directory structure.

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

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

Basic steps to cluster an Oracle Tuxedo machine

The process of clustering an Oracle Tuxedo machine is less complex for an SHM application model than an MP application model. When the model is SHM, each Oracle Tuxedo machine is independent and is configured and clustered independent of other Oracle Tuxedo machines in the environment. When the model is MP, all Oracle 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 an Oracle Tuxedo machine is to decouple it from any node in the cluster, especially the node on which the Oracle 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 Oracle Tuxedo machine's various network addresses.

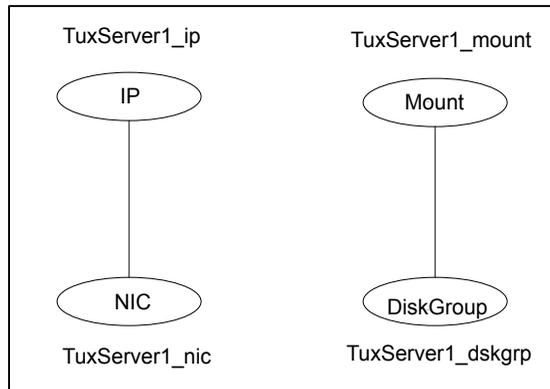
Regardless of the application model, use the following steps as a basic guide for clustering each Oracle Tuxedo machine.

To cluster Oracle Tuxedo machines

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

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

The following figure shows such a service group.



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

The following excerpt from an Oracle 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 Oracle 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 Oracle 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 Oracle 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 Oracle Tuxedo machine, which handles communications between ATMI clients and the Oracle Tuxedo server application.
JSL	A Jolt Server Listener process, running on the Oracle 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 Oracle Tuxedo machine, which handles communications between Jolt clients and the Oracle Tuxedo server application.
ISL	An IIOp Listener process, running on the Oracle 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 Oracle Tuxedo machine, which handles communications between CORBA clients and the Oracle Tuxedo server application.

It is important to specify virtual host names instead of virtual IP addresses for these parameters if the Oracle Tuxedo server volumes are replicated to an alternate computing site using 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 Oracle Tuxedo machine will be different on the two sites, but the virtual host name may be the same. When switching an Oracle 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 Oracle Tuxedo machine's virtual IP address. This ensures that all network traffic

targeted for this Oracle 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 Oracle 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 Oracle 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 Oracle Tuxedo machine. Thus, 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 an Oracle Tuxedo application and are not used for TCP/IP network communications.

- 10 Add the variable PMID to the Oracle 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 Oracle Tuxedo configuration file into its binary format using `tmloadcf`. Be sure the PMID environment variable is set before executing `tmloadcf`.
- 12 The 'TuxDir/bin/tmipcrm' utility, which the Tuxedo agent relies on to identify IPC resources pertaining to a particular Tuxedo instance, is not shipped along with Tuxedo 6.5. Hence, in a Tuxedo 6.5 environment, if multiple Tuxedo instances are sharing the same administrative UNIX login, you must create a dummy executable 0-byte file in the following location: `TuxDir/bin`.

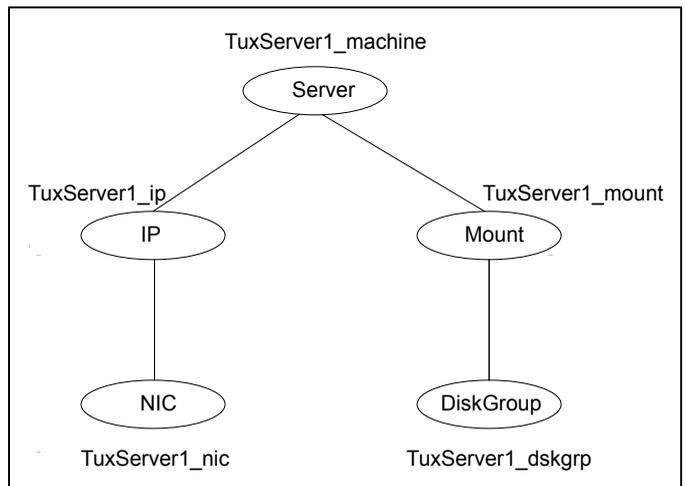
To create the dummy executable file, as the Tuxedo user, run the following commands:

- `$ touch TuxDir/bin/tmipcrm`
- `$ chmod +x TuxDir/bin/tmipcrm`

Note: You do not need to perform this step if you are using a dedicated UNIX login per Tuxedo instance.

- 13 Before placing the Oracle Tuxedo machine under VCS control, verify that the Oracle Tuxedo machine can be successfully started and stopped manually. Be sure to use the environment file established for this Oracle Tuxedo machine (EnvFile) and the dedicated user account (User).
- 14 Using the Tuxedo type, place the Oracle Tuxedo machine under VCS control by creating a resource for it. Test the Oracle 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.



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

For detailed instructions:

See [“Clustering Oracle Tuxedo listeners”](#) on page 43.

- 16 If the Oracle 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 31.

Configuring Oracle Tuxedo using the ULOGPFX attribute

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

```
AppDir/log/ULOG.mmdyy
```

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

The following is an excerpt from a UBBCONFIG file that makes use of the ULOGPFX 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"
ULOGPFX="/bea/tux/tuxedo91/backup/log/ULOG"
TUXCONFIG="/bea/tux/tuxedo91/app/simpapp/tuxconfig"
TUXDIR="/bea/tux/tuxedo91"
```

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

```
$ dirname ULOGPFX
```

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 Oracle Tuxedo unless the value of AppDir has been set as specified.

For information about the AppDir attribute:

See [Table 4-1](#) on page 82.

Additional requirements for multiple machine domains (MP model)

Supporting multiple machine domains with the Oracle Tuxedo agent introduces additional complexities and high availability scenarios compared to the SHM model. This complexity is mostly due to interdependencies among the Oracle Tuxedo machines in the domain. The most important dependency is that each non-master machine depends on the machine hosting the active master. Thus, when starting the entire Oracle Tuxedo application, the machine hosting the active master must always be started before the non-masters (the active master first establishes the domain bulletin board and then the non-masters join this bulletin board). Conversely, the active master machine should be the last machine to be stopped when shutting down the entire application.

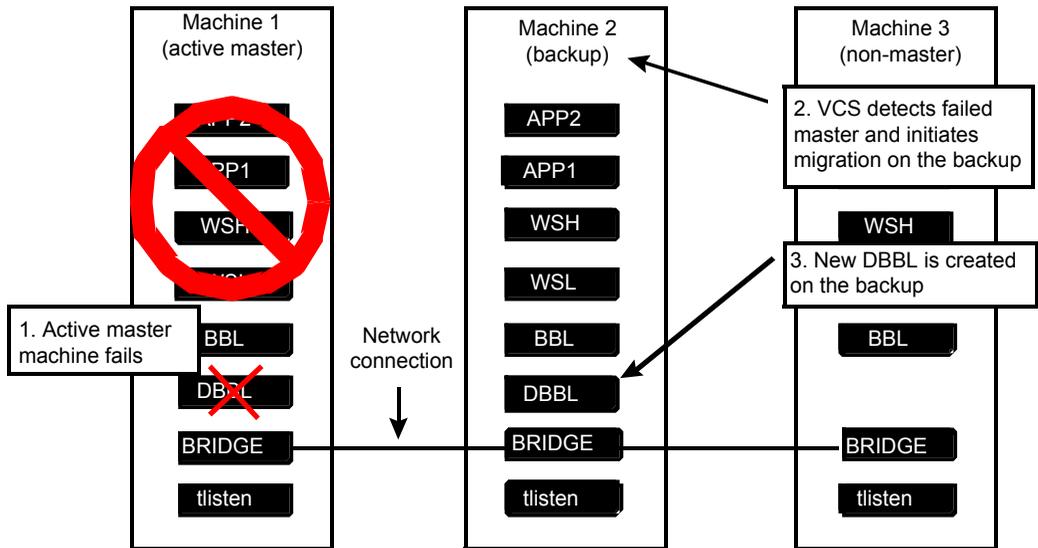
The Oracle Tuxedo agent must also be aware that the active master can be hosted on either the master machine or the backup machine. This built-in high availability feature of Oracle Tuxedo enables an Oracle Tuxedo administrator to manually migrate the active master between the master and the backup machine for performing maintenance on a system or in response to a fault of the active master. The purpose of this feature is to maximize the availability of the domain bulletin board and avoid a restart of the entire Oracle Tuxedo application. Without VCS, the master migration process must be performed manually by the Oracle Tuxedo administrator; the Oracle Tuxedo software does not automatically migrate the master. And delaying the master migration process may jeopardize the state of the domain bulletin board.

Using VCS and the Oracle Tuxedo agent solves this problem and significantly enhances the availability of an Oracle Tuxedo application by automatically detecting a faulted active master machine (whether it is hosted on the master or backup machine) and then initiating a master migration process on the machine currently acting as the backup. During the migration process, the non-master machines continue to run without interruption and wait patiently for the active master to be migrated to its new machine.

[Figure 2-3](#) illustrates this process.

Figure 2-3

Automated master migration



High availability scenarios supported

The following sections describe two high availability scenarios relevant to the complexities just discussed and how the Oracle 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 Oracle 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 Oracle Tuxedo service groups that are part of a multiple machine domain. This restriction is necessary due to Oracle 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 an Oracle 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 Oracle Tuxedo application is offline, the Oracle Tuxedo agent is only aware that the active master may be hosted by either the master or backup machine.

In the event that multiple Oracle Tuxedo service groups are started simultaneously (including the machine with the active master), the Oracle 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 Oracle Tuxedo application, but forces a master migration to occur.

Configuring Remote Shell or OpenSSH

To manage a multiple machine domain, the Oracle Tuxedo agent requires either Remote Shell (rsh) or OpenSSH (ssh) to be installed on each node in the cluster that runs an Oracle Tuxedo machine. In addition, each dedicated account (User) created for the Oracle Tuxedo machines must be configured to use the remote access solution. Both remote access solutions enable the agent entry points to remotely execute Oracle 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 `tadmin` (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 `tmbot` 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 'tmbot -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 Oracle 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 Oracle 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 `tadmin` 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. 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 Oracle 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 an Oracle 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 Oracle Tuxedo product. For example, if a BBL, BRIDGE, or application server process was terminated accidentally, the surviving Oracle Tuxedo server administrative processes can detect its absence and re-spawn it. Increasing the `ToleranceLimit` provides the Oracle Tuxedo machine the latitude to self-recover before the agent finally faults the Oracle Tuxedo machine and initiates a restart or switch.

Clustering Oracle Tuxedo listeners

In a multiple machine domain (application model is MP), an Oracle Tuxedo listener process (`tlisten`) must be started for each machine in the networked Oracle Tuxedo application before the application is booted. The listener process enables the Oracle Tuxedo active master machine to start, shut down, and administer Oracle Tuxedo processes running on the non-master machines.

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 Oracle Tuxedo machine it supports. In addition,

the listener resource must be configured as a child of the Oracle 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.

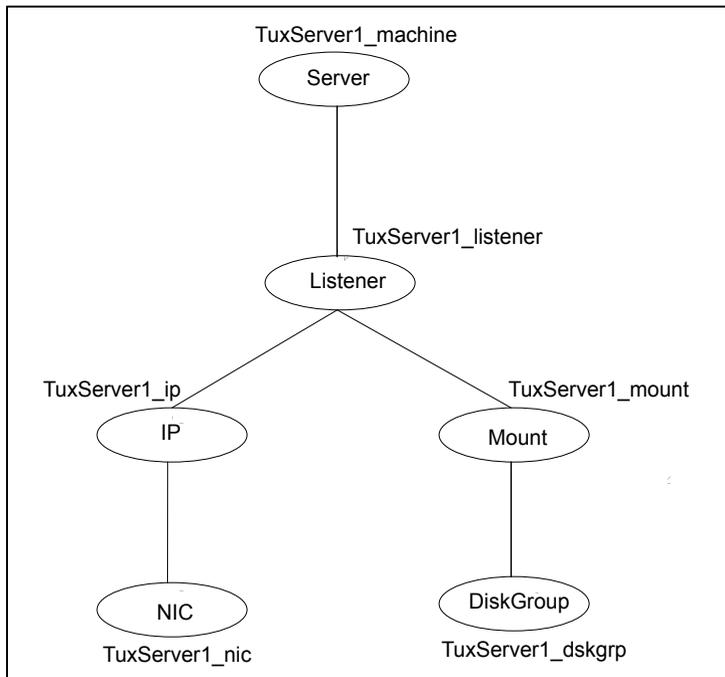
```

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 2-4 shows how a service group appears after the listener is configured.

Figure 2-4 Sample service group with a listener configured



The scripts `startListener` and `stopListener` referenced in this example are not included with the Oracle Tuxedo software or the agent for Oracle 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 $2}')
if [ "$pid" != "X" ]; then
    kill $pid
fi
exit 0
```

The start script is simply a one-line program that sources the Oracle Tuxedo environment file for the Oracle 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.

High availability scenarios supported

The following sections describe two high availability scenarios relevant to the complexities just discussed and how the Oracle 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 Oracle 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 Oracle Tuxedo service groups that are part of a multiple machine domain. This restriction is necessary due to Oracle 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 an Oracle 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 Oracle Tuxedo application is offline, the Oracle Tuxedo agent is only aware that the active master may be hosted by either the master or backup machine.

In the event that multiple Oracle Tuxedo service groups are started simultaneously (including the machine with the active master), the Oracle 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 Oracle Tuxedo application, but forces a master migration to occur.

Configuring Remote Shell or OpenSSH

To manage a multiple machine domain, the Oracle Tuxedo agent requires either Remote Shell (rsh) or OpenSSH (ssh) to be installed on each node in the cluster that runs an Oracle Tuxedo machine. In addition, each dedicated account (User) created for the Oracle Tuxedo machines must be configured to use the remote access solution. Both remote access solutions enable the agent entry points to remotely execute Oracle 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 Oracle 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 Oracle 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. 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 Oracle 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 an Oracle 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 Oracle Tuxedo product. For example, if a BBL, BRIDGE, or application server process was terminated accidentally, the surviving Oracle Tuxedo server administrative processes can detect its absence and re-spawn it. Increasing the ToleranceLimit provides the

Oracle Tuxedo machine the latitude to self-recover before the agent finally faults the Oracle Tuxedo machine and initiates a restart or switch.

Clustering Oracle Tuxedo listeners

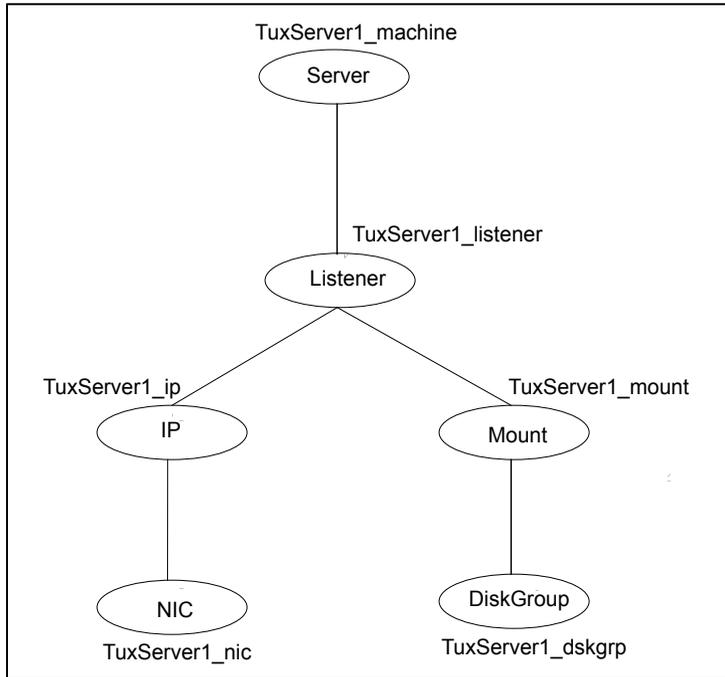
In a multiple machine domain (application model is MP), an Oracle Tuxedo listener process (tlisten) must be started for each machine in the networked Oracle Tuxedo application before the application is booted. The listener process enables the Oracle Tuxedo active master machine to start, shut down, and administer Oracle Tuxedo processes running on the non-master machines.

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 Oracle Tuxedo machine it supports. In addition, the listener resource must be configured as a child of the Oracle 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 2-4](#) shows how a service group appears after the listener is configured.

Figure 2-5 Sample service group with a listener configured

The scripts `startListener` and `stopListener` referenced in this example are not included with the Oracle Tuxedo software or the agent for Oracle 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 Oracle Tuxedo environment file for the Oracle 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.

Oracle Tuxedo entities in a clustered environment

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

The required resources are as follows:

Disk Group	<p>A disk group contains a volume and a file system, which is a mount resource containing the Oracle Tuxedo machine installation files.</p> <p>Use the <code>DiskGroup</code> 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.</p>
Mount	<p>The mount resource mounts, monitors, and unmounts the file system that is dedicated to the Oracle Tuxedo machine installation files.</p> <p>Use the <code>Mount</code> resource type to create this resource.</p>
Network Interface	<p>The Network Interface resource monitors the Network Interface Card (NIC) through which the Oracle Tuxedo machine instances communicate with the other services.</p> <p>Use the <code>NIC</code> resource type to create this resource.</p>
Virtual IP	<p>The Virtual IP resource configures the virtual IP address dedicated to the Oracle Tuxedo machine instance. The external services, programs and clients use this address to communicate with this instance.</p> <p>Use the <code>IP</code> resource type to create this resource.</p>

Oracle Tuxedo Server The Oracle Tuxedo Server resource starts, stops, and monitors the Oracle Tuxedo Server instance.

Use the Oracle Tuxedo resource type to create this resource.

Clustering PeopleSoft Enterprise Servers

This section provides additional instructions and examples to cluster PeopleSoft Enterprise middle-tier servers using the Oracle 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.

Overview of PeopleSoft

The agent for Oracle 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 Oracle Tuxedo servers for its middle-tier application servers and process schedulers; thus, these middle-tier services can be made highly available using the Oracle 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 Oracle Tuxedo's administrative utility called `tadmin`, which is the interface used by the Oracle Tuxedo agent. Creating the correct UNIX environment file for the Oracle Tuxedo server is one of the most important tasks towards successfully using `tadmin` and the Oracle Tuxedo agent to start, stop, and monitor a PeopleSoft server. It is essential to source a complete and accurate environment file before executing `tadmin` to start or stop an Oracle Tuxedo server instance. When defining a VCS resource using the Oracle Tuxedo agent, this environment file is identified by the attribute `EnvFile`.

Creating the Oracle Tuxedo environment file

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

The required environment variables and values to be included in the Oracle 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 `ps` command does not list the environment variables of the processes running on HP-UX.

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	<code>/usr/ucb/ps -auxeww PID for BBL process</code>
AIX	<code>/bin/ps auxew PID for BBL process</code>
Linux	<code>/bin/ps auxeww grep BBL</code>

- 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/sas12/64:/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=PSFDBSQL 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/sbin:/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  
atel142/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 47.

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 tadmin 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 (tadmin)
  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...
tadmin - 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 This step is applicable only for PeopleSoft application servers. Skip this step if you are configuring a process scheduler.

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 lenv and redirect the output to another file. This captures additional variables missed, if any, in 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
```

5 Extract the Tuxedo environment variables from the PeopleSoft system.

You can either extract these variables manually or can use the `genenv` utility.

■ Manual method:

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` while it executes the `tmadmin` commands.

The following command illustrates this step:

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

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.
- genenv utility method

Use the genenv utility to create the environment files that can be used with the agent for C-shell and Bourne-shell variants.

This utility utilizes the file containing the environment created by the psadmin interactive session, and consolidates it with the one created from,

`$PS_HOME/appserv/psadmin -env`

The genenv utility is located at the following path:

VCS 4.x /opt/VRTSvc/bin/Tuxedo/genenv

VCS 5.x /opt/VRTSagents/ha/bin/Tuxedo/genenv

To create the environment file using the genenv utility, execute the utility in the same login terminal in which the environment was dumped from the psadmin interactive session. This will ensure that all PeopleSoft and Tuxedo specific environment has been correctly set, to administer a PeopleSoft domain.

The following example shows the setup for VCS 5.x

```
$ /opt/VRTSagents/ha/bin/Tuxedo/genenv -h

genenv - Generate PSFT environment file for VCS Tuxedo Agent

Usage:

genenv -f infile [-ahv][-o outfile]

-a --all          Dump all environment variables
-f --infile       Environment dump file from tadmin prompt
-h --help         Help (this message)
-o --outfile      Output environment file
-v --verbose      Be verbose

$ /opt/VRTSagents/ha/bin/Tuxedo/genenv -f ./psadmin_env_menu \
-o ./envfile
```

As shown in the above example, the “psadmin_env_menu” file, that was generated earlier using the psadmin interactive session is used as an input

file and two environment files, namely `envfile_sh` (for Bourne-shell and its variants) and `envfile_csh` (for C-shell) are created in the current working directory. However, if the `'-o'` option is not used, the utility creates the files `"envfile_sh"` and `"envfile_csh"` under `$PS_HOME`.

By default, the following environment variables gets omitted in the resulting output files:

```
_ MANPATH SSH_TTY COLUMNS EDITOR ERASE SHELL TERM PWD PAGER  
LINES MAIL OLDPWD SSH_CLIENT SSH_CONNECTION
```

However, if you want to have this variable as a part of the environment files, use the `'-a'` flag to dump the entire environment.

Additionally, note that

- For C-shell, the value of the `"PATH"` environment variable is appended to the `"path"` variable. Similarly, for Bourne-shell, the value of the C-shell specific `"path"` variable is appended to the `"PATH"` environment variables. Also, only the values of variables containing `":"` are merged and the duplicates, if found are removed..
- If two instances of a variable having different values are encountered and do not have a `":"`, the value specified in the input file is used.

- You must verify the resultant output files before use.
- 6 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 `tadmin` utility:

See "Testing the Tuxedo environment file" on page 60.

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 `tadmin` 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 = PSPRC_SRV
)

```

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/psftappl] ./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:[people]
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 Oracle 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 tadmin 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 4-2](#) on page 87.

Installing, upgrading, and removing the agent for Oracle Tuxedo

This chapter includes the following topics:

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

Before you install the Cluster Server agent for Oracle Tuxedo

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

Before you install the agent for Oracle Tuxedo, ensure that the following prerequisites are met.

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

- Remove any previous version of this agent.
To remove the agent,
See [“Uninstalling the agent in a VCS environment”](#) on page 76.
- Install the latest version of ACC Library.
To install or update the ACC Library package, locate the library and related documentation in the Agent Pack tarball,
See [“Installing the ACC library”](#) on page 70.

Prerequisites for enabling i18n support

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

- Install ACCLib version 5.1.3.0 or later.
See [“Installing the ACC library”](#) on page 70.
- For VCS 5.0 and earlier releases, copy the latest `ag_i18n_inc.pm` module from the following location on the agent pack disc.

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

VCS 5.0 `cd1/platform/arch_dist/vcs/application/i18n_support/5.0`

VCS 4.1 `cd1/platform/arch_dist/vcs/application/i18n_support/4.1`

VCS 4.0 `cd1/platform/arch_dist/vcs/application/i18n_support/4.0`

where `arch_dist` takes the following values:
'sol_sparc' for Solaris SPARC

Prerequisites for installing the agent to support Solaris zones

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

- Install Tuxedo to support Solaris zones. For details, refer to the Tuxedo user documentation.
- Install and configure the VCS 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 the ACC library

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

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

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

Installing the ACC library

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

To install the ACC library

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

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

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

AIX `cd1/aix/vcs/application/acc_library/version_library/pkggs`

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

Linux `cd1/linux/generic/vcs/application/acc_library/version_library/rpms`

Solaris `cd1/solaris/dist_arch/vcs/application/acc_library/version_library/pkggs`

where `dist_arch` is `sol_sparc`.

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

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

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

Installing the ACC library IPS package on Oracle Solaris 11 systems

To install the ACC library IPS package on an Oracle Solaris 11 system

- 1 Copy the VRTSacclib.p5p package from the pkgs directory to the system in the /tmp/install directory.
- 2 Disable the publishers that are not reachable as package install may fail, if any, of the already added repositories are unreachable.

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

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

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

- 4 Install the package.

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

- 5 Remove the publisher from the system.

```
# pkg unset-publisher Symantec
```

- 6 Enable the publishers that were disabled earlier.

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

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

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

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

1 Ensure that the SMF service

`svc:/application/pkg/system-repository:default` and
`svc:/application/pkg/zones-proxyd:default` are online on the global zone.

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

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

3 Ensure that the SMF service

`svc:/application/pkg/zones-proxy-client:default` is online inside the non-global zone:

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

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

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

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

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

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

7 Install the package.

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

8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Symantec
```

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

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

10 Enable the publishers that were disabled earlier.

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

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

Installing the agent in a VCS environment

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

To install the agent in a VCS environment

- 1 Download the agent from the Symantec Operations Readiness Tools (SORT) site: <https://sort.symantec.com/agents>.

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

- 2 Uncompress the file to a temporary location, say /tmp.
- 3 If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

AIX `cd1/aix/vcs/application/tuxedo_agent/
 vcs_version/version_agent/pkg`

HP-UX `cd1/hpux/generic/vcs/application/tuxedo_agent/
 vcs_version/version_agent/pkg`

Linux `cd1/linux/generic/vcs/application/tuxedo_agent/
 vcs_version/version_agent/rpms`

Solaris `cd1/solaris/dist_arch/vcs/application/tuxedo_agent/
 vcs_version/version_agent/pkg`

where, *dist_arch* is sol_sparc.

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

- 4 Log in as a superuser.

5 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
```

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

Installing the agent IPS package on Oracle Solaris 11 systems

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

- 1** Copy the `VRTStuxedo.p5p` package from the `pkgs` directory to the system in the `/tmp/install` directory.
- 2** Disable the publishers that are not reachable as package install may fail if any of the already added repositories are unreachable.

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

where the publisher name is obtained using the `pkg publisher` command.
- 3** Add a file-based repository in the system.

```
# pkg set-publisher -g /tmp/install/VRTStuxedo.p5p Symantec
```
- 4** Install the package.

```
# pkg install --accept VRTStuxedo
```
- 5** Remove the publisher from the system.

```
# pkg unset-publisher Symantec
```
- 6** Enable the publishers that were disabled earlier.

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

Installing agent packages on Solaris brand non-global zones

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

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

1 Ensure that the SMF service

`svc:/application/pkg/system-repository:default` and `svc:/application/pkg/zones-proxyd:default` are online on the global zone.

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

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

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

3 Ensure that the SMF service

`svc:/application/pkg/zones-proxy-client:default` is online inside non-global zone:

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

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

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

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

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

```
# pkg set-publisher -g/tmp/install/VRTStuxedo.p5p Symantec
```

7 Install the package.

```
# pkg install --accept VRTStuxedo
```

8 Remove the publisher on the non-global zone.

```
# pkg unset-publisher Symantec
```

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

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

10 Enable the publishers that were disabled earlier.

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

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

Uninstalling the agent in a VCS environment

You must uninstall the agent for Oracle 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 running the following command from any node in the cluster:

```
# haconf -makerw
```

- 3 Remove all Oracle Tuxedo resources from the cluster. Run 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 running 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 running the following command from any node in the cluster:

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

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

Run the following command to uninstall the agent:

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

```
HP-UX # swremove VRTStuxedo
```

```
Linux # rpm -e VRTStuxedo
```

```
Solaris # pkgrm VRTStuxedo
```

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

```
# pkg uninstall 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
```

```
HP-UX # swremove VRTSacclib
```

```
Linux # rpm -e VRTSacclib
```

```
Solaris # pkgrm VRTSacclib
```

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

```
# pkg uninstall VRTSacclib
```

Upgrading the agent in a VCS environment

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

To upgrade the agent in a VCS environment

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

```
# hagr -freeze GroupName -persistent
```
- 2 Stop the cluster services forcibly.

```
# hastop -all -force
```
- 3 Ensure that the agent operations are stopped on all the nodes.

```
# ps -ef | grep Tuxedo
```
- 4 Uninstall the agent package from all the nodes. Use the platform's native software management program to remove the agent for Oracle Tuxedo from each node in the cluster.

Run the following command to uninstall the agent:

```
AIX          # installp -u VRTStuxedo.rte
HP-UX       # swremove VRTStuxedo
Linux       # rpm -e VRTStuxedo
Solaris     For Solaris 10:
            # pkgrm VRTStuxedo
            For Solaris 11:
            # pkg uninstall VRTStuxedo
```

- 5 Install the new agent on all the nodes.
See [“Installing the agent in a VCS environment”](#) on page 73.
- 6 Copy the new `TuxedoTypes.cf` file from the agent's conf directory, to the VCS conf directory `/etc/VRTSvcS/conf/config`.

```
VCS 4.x      ■ AIX      /etc/VRTSvcS/conf/sample_Tuxedo/
              ■ HP-UX   TuxedoTypes.cf
              ■ Linux
              ■ Solaris
```

VCS 5.x or later	■ AIX	/etc/VRTSagents/ha/conf/Tuxedo/
	■ HP-UX	TuxedoTypes.cf
	■ Linux	
VCS 5.0	■ Solaris SPARC	/etc/VRTSagents/ha/conf/Tuxedo/ TuxedoTypes50.cf
VCS 5.1 or later	■ Solaris SPARC	/etc/VRTSagents/ha/conf/Tuxedo/ TuxedoTypes51.cf

Note: If you are using Solaris SPARC, copy the `TuxedoTypes50.cf` file for VCS 5.0 (and its intermediate Maintenance Packs) and `TuxedoTypes51.cf` file for VCS 5.1 or later.

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

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

- 8 Start the cluster services.

```
# hastart
```

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

```
# haagent -start Tuxedo -sys SystemName
```

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

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

Configuring the agent for Oracle Tuxedo

This chapter includes the following topics:

- [About configuring the Cluster Server agent for Oracle Tuxedo](#)
- [Importing the agent types files in a VCS environment](#)
- [Agent attributes](#)
- [Executing a customized monitoring program](#)

About configuring the Cluster Server agent for Oracle Tuxedo

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

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

See [“About sample configurations for the agents for Oracle Tuxedo”](#) on page 98.

Importing the agent types files in a VCS environment

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

You can import the agent types file using the VCS graphical user interface or using the command line interface.

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

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`.

VCS 4.x	<ul style="list-style-type: none"> ■ AIX ■ HP-UX ■ Linux ■ Solaris 	<pre>/etc/VRTSvcs/conf/sample_Tuxedo/ TuxedoTypes.cf</pre>
VCS 5.x or later	<ul style="list-style-type: none"> ■ AIX ■ HP-UX ■ Linux 	<pre>/etc/VRTSagents/ha/conf/Tuxedo/ TuxedoTypes.cf</pre>
VCS 5.0	<ul style="list-style-type: none"> ■ Solaris SPARC 	<pre>/etc/VRTSagents/ha/conf/Tuxedo/ TuxedoTypes50.cf</pre>
VCS 5.1 or later	<ul style="list-style-type: none"> ■ Solaris SPARC 	<pre>/etc/VRTSagents/ha/conf/Tuxedo/ TuxedoTypes51.cf</pre>

The following example assumes VCS 5.0 is installed on AIX:

```
# 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 Oracle Tuxedo agent type is now imported to the VCS engine.

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

Agent attributes

[Table 4-1](#) lists the required attributes for the agent for Oracle Tuxedo.

Table 4-1 Required attributes

Attribute name	Description
AppDir	<p>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 Oracle Tuxedo instance share the same application directory.</p> <p>Note: If this instance of Oracle Tuxedo makes use of the ULOGPFX attribute, set the value of AppDir as specified in the guide.</p> <p>See "Configuring Oracle Tuxedo using the ULOGPFX attribute" on page 30.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /bea/tux91mach1/apps/simpapp</p>

Table 4-1 Required attributes (*continued*)

Attribute name	Description
EnvFile	<p>Absolute path of the file to source for setting the environment prior to executing Oracle Tuxedo administrative programs such as tmbboot, tmsshutdown, and tmadmin.</p> <p>The following shell environments are supported: Bourne, Korn, and C shell.</p> <p>recommends using the Bourne shell script included in the Oracle Tuxedo installation (\$TUXDIR/tux.env) and adding additional variables as required by your environment. also recommends storing the environment file on a shared disk along with the Oracle Tuxedo software and application files (that is, files in \$TUXDIR and \$APPDIR).</p> <p>Also consider setting the PMID environment variable to the MachineName attribute value in this file, which is the same as defined in the MACHINES section of the UBBCONFIG file.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /tuxedo/srvr1/tuxedo9.1/tux.env</p>
MachineName	<p>Logical machine name (LMID) assigned to this Oracle Tuxedo machine as specified in the Oracle Tuxedo configuration file. Although not required for clustering, to avoid confusion in the Oracle Tuxedo configuration file, consider setting the machine's physical machine name (PMID) and the LMID to the same value.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: Machine1</p>
MachineType	<p>Identifies the bulletin board architecture and machine type of the Oracle Tuxedo machine. The agent entry points use this value and other resource state information to determine how to boot, shut down, and monitor the Oracle Tuxedo machine.</p> <p>The valid values and their definitions are as follows:</p> <ul style="list-style-type: none"> ■ Value-Single; Model-SHM; MachineType-Master ■ Value-Master; Model- MP; MachineType-Master ■ Value-Backup; Model-MP; MachineType-Backup ■ Value-NonMaster; Model-MP; MachineType-NonMaster <p>Type and dimension: string-scalar</p> <p>Default: Single</p> <p>Example: Master</p>

Table 4-1 Required attributes (*continued*)

Attribute name	Description
RemoteShell	<p>Absolute path name of the remote shell program used by agent entry points to remotely issue administrative Oracle 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).</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1: /bin/remsh</p> <p>Example 2: /usr/local/bin/ssh</p>
TmbootOptMaster	<p>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.</p> <p>The program tmboot starts an Oracle Tuxedo machine in whole or in part depending on the options specified. Refer to the Oracle Tuxedo documentation for a complete list of options and their descriptions. If the resource is an Oracle 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.</p> <p>If the resource is an Oracle Tuxedo machine defined as the backup in TUXCONFIG and the machine will host application servers, then follow Example 2. The options "-M -I MachineName" start both administrative and application server processes on the master machine.</p> <p>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.</p> <p>The tmboot option -y should not be used because this would cause all Oracle Tuxedo machines configured through the TUXCONFIG file to be booted outside VCS control.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1: -M</p> <p>Example 2: -M -I machine1</p>

Table 4-1 Required attributes (*continued*)

Attribute name	Description
TmshutdownOptMaster	<p>Command-line options passed to tmshutdown to stop an Oracle 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.</p> <p>The program tmshutdown stops an Oracle Tuxedo application in whole or in part, depending on the options specified. Refer to the Oracle Tuxedo documentation for a complete list of options and their descriptions.</p> <p>If the Oracle 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.</p> <p>If the Oracle 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 -I MachineName" stop both administrative and application processes on the master machine.</p> <p>The tmboot option -y should not be used because this would cause all Oracle Tuxedo machines configured through the TUXCONFIG file to be shut down outside VCS control.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1: -M</p> <p>Example 2: -M -I machine1</p>
TuxDir	<p>Absolute path name to the Oracle Tuxedo software base installation directory. The agent entry points use TuxDir to locate programs such as tmboot, tmshutdown, and tmsadmin.</p> <p>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.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: /tuxedo/srvr1/tuxedo9.1</p>

Table 4-1 Required attributes (*continued*)

Attribute name	Description
TuxDomain	<p>Name of the Oracle Tuxedo domain to which the machine belongs, which is the same name specified by the parameter DOMAINID in the RESOURCES section of the Oracle Tuxedo configuration file. When the application model is MP, the agent entry points use the TuxDomain value to associate Oracle 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.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: finance</p>
User	<p>UNIX login name assigned to the Oracle Tuxedo machine. strongly recommends that the login name be unique within the cluster. Sharing a UNIX login name across Oracle Tuxedo machines could compromise the high availability of the Oracle Tuxedo setup. In case the UNIX login name is not dedicated to the Oracle Tuxedo machine, it has to be registered with the agent by adding a + at the start of the actual login name.</p> <p>For more information about the behavior of the agent when a shared login is registered: See "Identifying IPC resources pertaining to Oracle Tuxedo" on page 15.</p> <p>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.</p> <p>With this user, the agent entry points execute Oracle Tuxedo administrative programs such as tmboot, tmshutdown, tmipcrm, and tmsadmin. The user's login shell must be Bourne, Korn, or C shell.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1 (dedicated UNIX login): tux91adm</p> <p>Example 2 (shared UNIX login): +tux91adm</p>

[Table 4-2](#) lists the optional attributes for the agent for Oracle Tuxedo.

Table 4-2 Optional attributes

Attribute name	Description
MonitorProgram	<p>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 Oracle Tuxedo machine state check. There are no restrictions for what actions the external monitor program performs to determine the state of an Oracle Tuxedo machine instance.</p> <p>The only constraint is that the external monitor program must return one of the following integer values:</p> <ul style="list-style-type: none"> ■ 110 or 0: Oracle Tuxedo machine is online ■ 100 or 1: Oracle Tuxedo machine is offline ■ Any other value: Oracle Tuxedo machine status is unknown <p>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.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example 1: /tuxedo/srvr1/tuxedo9.1/bin/myMonitor.pl</p> <p>Example 2: /tuxedo/srvr1/tuxedo9.1/bin/myMonitor.sh arg1 arg2</p>
ResLogLevel	<p>Logging details performed by the agent for the resource.</p> <p>The valid values are as follows:</p> <ul style="list-style-type: none"> ■ 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. <p>Type and dimension: string-scalar</p> <p>Default: INFO</p> <p>Example: TRACE</p>

Table 4-2 Optional attributes (*continued*)

Attribute name	Description
SecondLevelMonitor	<p>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 Oracle 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.</p> <p>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.</p> <p>Type and dimension: integer-scalar Default: 0 Example: 5</p>
TmbootOptNonMaster	<p>Command-line options passed to tmboot to start any type of Oracle 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.</p> <p>The attribute is also used to specify command-line options passed to tmboot to start a stand-alone Oracle Tuxedo machine (that is, where the resource MachineType is Single). Follow example 2 to support this scenario.</p> <p>Review the definition of attribute TmbootOptMaster to specify consistent values between these two attributes.</p> <p>Type and dimension: string-scalar Default: -y Example 1 (MP mode): -B machine1 -I machine1 Example 2 (SHM mode): -y</p>

Table 4-2 Optional attributes (*continued*)

Attribute name	Description
TmshutdownOptNonMaster	<p>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 1 for this scenario.</p> <p>If MachineType is Single, the options specified here are passed to tmshutdown to stop a stand-alone, independent Oracle Tuxedo machine. Follow example 2 for this scenario. Read the definition of attribute TmshutdownOptMaster to specify consistent values between these two attributes.</p> <p>Type and dimension: string-scalar</p> <p>Default: -y</p> <p>Example 1 (MP mode): -B machine1 -I machine1</p> <p>Example 2 (SHM mode): -y</p>
TuxServers	<p>Comma-delimited list of Oracle Tuxedo servers to monitor. Candidate server names can be obtained for an Oracle Tuxedo server by executing tmadm and issuing the sub-command psr.</p> <p>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.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: simpserv, myserver, this server, JSL, WLS, WSH</p>
TuxServices	<p>Comma-delimited list of Oracle Tuxedo program services to monitor. Candidate services can be obtained for an Oracle Tuxedo server by executing tmadm and issuing the sub-command psc.</p> <p>Note: The program services specified for PeopleSoft deployments using Oracle Tuxedo must follow certain naming constraints.</p> <p>See "Registering long PeopleSoft service names" on page 66.</p> <p>Type and dimension: string-scalar</p> <p>Default: ""</p> <p>Example: TOUPPER, MYSERVICE, THISSERVICE</p>

Executing a customized monitoring program

You can configure the monitor function to execute a custom monitor utility to perform a user-defined Oracle 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 Oracle 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	Oracle Tuxedo machine instance is online
100 or 1	Oracle Tuxedo machine instance is offline
Any other value	Oracle Tuxedo machine instance is unknown

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

Configuring the service groups for Oracle Tuxedo using the CLI

This chapter includes the following topics:

- [About configuring service groups for Oracle Tuxedo](#)
- [Before configuring the service groups for Oracle Tuxedo](#)
- [Configuring Oracle Tuxedo resources for Solaris zones support](#)

About configuring service groups for Oracle Tuxedo

Configuring the Oracle Tuxedo service group involves creating the Oracle Tuxedo service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the service groups using one of the following:

- The Cluster Manager (Java console)
- Veritas Infoscale Operations Manager
- The command line

Before configuring the service groups for Oracle Tuxedo

Before you configure the Oracle Tuxedo service group, you must:

- Verify that Cluster Server is installed and configured on all nodes in the cluster where you will configure the service group.
For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Verify that the Cluster Server agent for Oracle Tuxedo is installed on all nodes in the cluster.
See [“Installing the agent in a VCS environment”](#) on page 73.

Configuring Oracle Tuxedo resources for Solaris zones support

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

- Install Oracle Tuxedo on dedicated Solaris zones.
- Preferably, follow the 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 Oracle Tuxedo.
- In a VCS environment, 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.

Troubleshooting the agent for Oracle Tuxedo

This chapter includes the following topics:

- [Using the correct software and operating system versions](#)
- [Meeting prerequisites](#)
- [Configuring Oracle Tuxedo resources](#)
- [Verifying virtualization](#)
- [Starting the Oracle Tuxedo instance outside a cluster](#)
- [Reviewing error log files](#)

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

For information on the software versions that the agent for Oracle Tuxedo supports, see the Symantec Operations Readiness Tools (SORT) site:

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

Meeting prerequisites

Before installing the agent for Oracle Tuxedo, ensure that the following prerequisites are met.

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

See [“Before you install the Cluster Server agent for Oracle Tuxedo”](#) on page 68.

Configuring Oracle Tuxedo resources

Before using Oracle Tuxedo resources, ensure that you configure the resources properly. For a list of attributes used to configure all Oracle 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 Oracle Tuxedo”](#) on page 18.

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

You can then restart the Oracle 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 described 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 Oracle Tuxedo or the agent for Oracle Tuxedo, use the log files described in this section to investigate the problems.

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	<p>Verify that ports have been properly configured and declared. Typically, ports from 1 through 1024 are reserved for the superuser.</p> <p>Ensure that parameters to the agent are correctly defined.</p>
Expired licenses	<p>Check the application log files for any error messages related to expired licenses.</p> <p>Ensure that the license keys/files have been placed at the appropriate location, as needed by the application.</p>
Broken symlinks, missing files, and libraries	<p>Verify your installation.</p> <p>Make sure that nothing is broken, and all dependencies for the executables are met.</p>
Insufficient disk space or system parameters	<p>Ensure that the file-system has sufficient space for creation of temporary files that the application might need.</p> <p>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.</p>

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 Oracle 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. 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.

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

To localize ResLogLevel attribute for a resource

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

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

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

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

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

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

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

- 6 Save the configuration changes.

```
# haconf -dump
```

- 7 Review the contents of the log file.

You can also contact support for more help.

To enable debug logs for all resources of type Tuxedo

- ◆ Enable the debug log.

```
# hatype -modify Tuxedo LogDbg DBG_5
```

To override the LogDbg attribute at resource level

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

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

Sample Configurations

This appendix includes the following topics:

- [About sample configurations for the agents for Oracle Tuxedo](#)
- [Sample agent type definition for Oracle Tuxedo](#)
- [Sample configuration file for Oracle Tuxedo](#)
- [Sample configuration for Solaris zone support](#)

About sample configurations for the agents for Oracle Tuxedo

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

Sample agent type definition for Oracle Tuxedo

This section lists the sample agent type definition for Oracle Tuxedo agent on different versions of VCS.

For VCS 4.x

```
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  
)  
)
```

For VCS 5.x

```
type Tuxedo (  
    static boolean AEPTIMEOUT = 1  
    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
)
```

For VCS 5.0 under Solaris zones

```
type Tuxedo (
    static boolean AEPTIMEOUT = 1
    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
)
```

For VCS 5.1 under Solaris zones

```
type Tuxedo (  
    static int ContainerOpts{} = { RunInContainer=1, PassCInfo=0 }  
    static boolean AEPTIMEOUT = 1  
    static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"  
    static str AgentDirectory = "/opt/VRTSagents/ha/bin/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 configuration file for Oracle Tuxedo

This section lists the VCS main.cf file for a Tuxedo machine under various application models.

For 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
//     }
// }
// }
```

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, JREPSRV, 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  
// }  
// }  
// }
```

For 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/stoplslnr"
```

```
CleanProgram = "/bea/tuxbk/tuxedo9.1/bin/stoplusr"
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 = simpservebk
  TuxServices = TOUPPER
  User = "+tux91adm"
  SecondLevelMonitor = 1
)
tux91bkp_mnt requires tux91bkp_dg
tux91bkp_ip requires tux91bkp_nic
tux91bkplusr requires tux91bkp_ip
tux91bkplusr 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/stoplusr"
  CleanProgram = "/bea/tuxm2/tuxedo9.1/bin/stoplusr"
  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 tux91nmlsrvr (
  EnvFile = "/bea/tuxm2/tuxedo9.1/tux.env"
  MachineName = tux91nmlsol
  MachineType = NonMaster
  RemoteShell = "/usr/bin/ssh"
  TmbootOptNonMaster = "-B tux91nmlsol -l tux91nmlsol"
  TmshutdownOptNonMaster = "-B tux91nmlsol -l tux91nmlsol"
  AppDir = "/bea/tuxm2/tuxedo9.1/app/simpapp"
  TuxDir = "/bea/tuxm2/tuxedo9.1"
  TuxDomain = simpapp
  TuxServers = simpservernml
  TuxServices = TOUPPER
  User = tuxadm2
  SecondLevelMonitor = 1
)
tux91nml_mnt requires tux91nml_dg
tux91nml_ip requires tux91nml_nic
tux91nml1lsnr requires tux91nml_ip
tux91nml1lsnr requires tux91nml_mnt
tux91nmlsrvr requires tux91nml1lsnr

// resource dependency tree
//
// group tux91nml
// {
//   Tuxedo tux91nmlsrvr
//   {
//     Application tux91nml1lsnr
//     {
//       IP tux91nml_ip
//       Mount tux91nml_mnt
//       {
//         DiskGroup tux91nml_dg
//       }
//     }
//   }
// }
group tux91nml2 (
  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 = simpservernm2
    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
//     }
//   }
// }
```

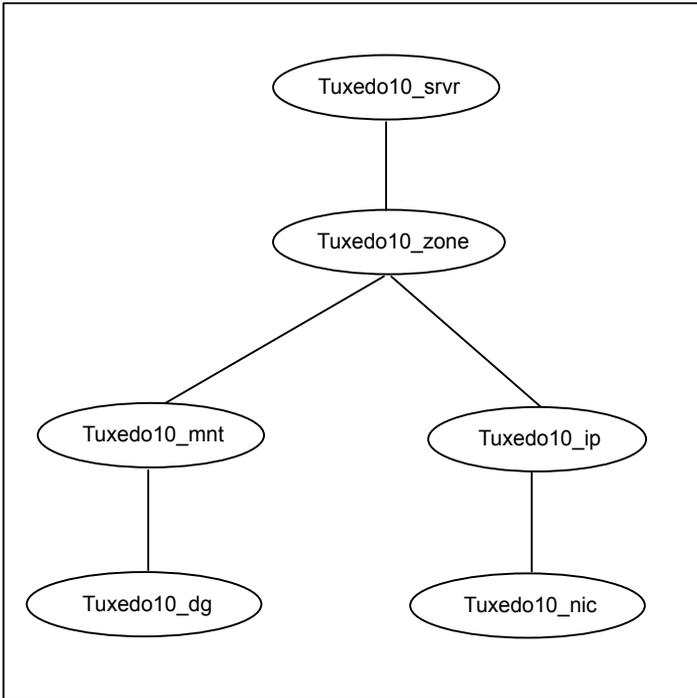
Sample configuration for Solaris zone support

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 zones



Index

A

- about
 - configuring service groups 91
- about ACC library 70
- about Oracle Tuxedo 17
 - See *also* about Oracle Tuxedo server; Oracle Tuxedo, overview
- about Oracle Tuxedo server 18
 - See *also* about Oracle Tuxedo; Oracle Tuxedo, overview
- ACC library
 - installing 70
 - removing 77
- agent
 - features 10
 - i18n support 69
 - importing agent types files 80
 - installing, VCS environment 73
 - optional attributes 86
 - overview 9
 - required attributes 82
 - uninstalling, VCS environment 76
 - upgrading 78
- agent attributes
 - AppDir 82
 - EnvFile 83
 - MachineName 83
 - MachineType 83
 - MonitorProgram 87
 - RemoteShell 84
 - ResLogLevel 87
 - SecondLevelMonitor 88
 - TmbootOptMaster 84
 - TmbootOptNonMaster 88
 - TmshutdownOptMaster 85
 - TmshutdownOptNonMaster 89
 - TuxDir 85
 - TuxDomain 86
 - TuxServers 89
 - TuxServices 89
 - User 86

- agent configuration file
 - importing 80
- agent installation
 - general requirements 68
 - requirements for Solaris zones 69
 - steps to install 73
- agent operations
 - clean 15
 - monitor 13
 - offline 12
 - online 11

B

- before
 - configuring the service groups 92

C

- Clustering
 - Oracle Tuxedo 19
 - PeopleSoft Enterprise Servers 46
- clustering Oracle Tuxedo machine
 - basic resources 45
- clustering Oracle Tuxedo machines 25
- configuration file, MP model 105
- configuration file, SHM model 101
- configuring monitor function 90
- creating Oracle Tuxedo environment file
 - using ps command 47
 - using psadmin utility 51

E

- executing custom monitor program 90

L

- logs
 - reviewing error log files 95
 - using trace level logging 96

M

- MP model
 - clustering Oracle Tuxedo listeners 36, 43
 - configuring OpenSSH 34, 41
 - configuring Remote Shell 34, 41
 - configuring ToleranceLimit 36, 42
 - sample configuration file 105
 - tuning MonitorTimeout value 35, 42

O

- Oracle Tuxedo
 - additional requirements for MP model 31
 - clustering 25
 - configuration using ULOGPFX 30
 - configuring resources 94
 - configuring resources for Solaris zones 92
 - creating environment file 46
 - directory structure 23–24
 - installing on shared disks 22
 - MP model 21
 - SHM model 21
 - starting instance outside cluster 94
 - testing environment file 60
 - virtualization 18
 - Host names 18
 - Path names 19
- Oracle Tuxedo machine
 - clustering
 - basic resources 45
- Oracle Tuxedo, overview 19
 - See *also* about Oracle Tuxedo; about Oracle Tuxedo server
- Oracle Tuxedo, shared disk installation 22

P

- PeopleSoft
 - about 46
 - configuration notes 65
 - obtaining values for MachineName 64
 - obtaining values for TuxDomain 64
 - registering long service names 66
 - sample configuration file 103
 - sample resource definition 63
- PeopleSoft, overview 46

S

- sample agent type definition
 - Tuxedo, Solaris zone support 100

- sample agent type definition, VCS 4.x 98
- sample agent type definition, VCS 5.1 101
- sample agent type definition, VCS 5.x 99
- service group
 - sample configurations, Solaris zone support 112
- service group dependencies, MP model 34, 40
- setting
 - Oracle Tuxedo in a cluster 16
- SHM model
 - sample configuration file 101
- Solaris zone support
 - configuring Oracle Tuxedo resources 92
 - installation requirements 69
 - sample agent type definition 100
 - Tuxedo, sample service group configurations 112
- starting the Oracle Tuxedo instance outside a cluster 94
- supported high availability scenarios 32, 38

T

- troubleshooting
 - meeting prerequisites 93
 - reviewing error log files 95
 - using trace level logging 96
 - using correct software 93
 - verifying virtualization 94
- Tuxedo
 - sample service group configurations, Solaris zone support 112

U

- uninstalling agent, VCS environment 76
- upgrading agent 78