Oracle® Tuxedo Message Queue (OTMQ)

Installation Guide

12*c* (12.1.1.0.0)

March 2012

Installing Oracle Tuxedo Message Queue, 12*c* (12.1.1.0.0)

Contents

1. Preparing to Install Oracle Tuxedo Message Queue

2. Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation

6. Performing Post-Installation Tasks

A. Oracle Tuxedo Message Queue 12*c* (12.1.1.0.0) Platform Data Sheets

B. Starting the Oracle Tuxedo Message Queue Administration Console

C. File and Database Management and Disk Space Allocation

D. IPC Resource Configuration on a UNIX System

Preparing to Install Oracle Tuxedo Message Queue

The following sections provide information that you need to know before installing the Oracle Tuxedo Message Queue (OTMQ)12*c* (12.1.1.0.0) product software:

Note: The installation of Oracle Tuxedo Message Queue 12c (12.1.1.0.0) is *integrated within* Oracle Tuxedo12c (12.1.1.0.0) installation. When you install Oracle Tuxedo 12c (12.1.1.0.0), you are asked if you want to install Oracle Tuxedo Message Queue 12c (12.1.1.0.0).

- Oracle Tuxedo Web Distribution
- Oracle Tuxedo Software Components
- LLE and SSL Levels of Encryption
- Hardware and Software Prerequisites
- Interprocess Communication Resources Configuration
- Oracle Installation Program
- Oracle Home Directory
- Installation Road Map

Oracle Tuxedo Software Components

Included with the Oracle Tuxedo 12*c* (12.1.1.0.0) software distribution is the Oracle Jolt product software, which enables Java clients to call Oracle Tuxedo services, and the Oracle SNMP Agent product software, which enables Oracle Tuxedo applications to be managed from a centralized enterprise management console.

The Oracle Tuxedo 12*c* (12.1.1.0.0) distribution contains the following components:

- Server components:
  - Oracle Application-to-Transaction Monitor Interface (ATMI) server software
  - Oracle Common Object Request Broker Architecture (CORBA) C++ server software
  - Oracle Jolt 12*c* (12.1.1.0.0) server software
  - Oracle SNMP Agent 12*c* (12.1.1.0.0) software
  - Oracle Tuxedo Administration Console software
  - Link-Level Encryption (LLE) and Secure Sockets Layer (SSL) encryption software
- Client components:
  - Oracle ATMI Workstation (/WS) client software
  - Oracle CORBA C++ client software—includes the C++ client Object Request Broker (ORB) and the environmental objects
  - Oracle Jolt 12*c* (12.1.1.0.0) client software
  - LLE and SSL encryption software

For descriptions of terms such as *remote client*, *native client*, *LLE*, and *SSL*, see “Client and Server Components” in *Oracle Tuxedo Product Overview*.

LLE and SSL Levels of Encryption

LLE and SSL are available with two levels of encryption: 56-bit and 128/256-bit.

Hardware and Software Prerequisites

The Oracle Tuxedo software must be installed on each server machine that will participate in a Oracle Tuxedo application, also known as a Tuxedo domain. A Tuxedo application is a business software program, built upon the Tuxedo system, that is defined and controlled by a single configuration file known as the `UBBCONFIG` file. The Tuxedo configuration file is described in reference page `UBBCONFIG(5)` in *Oracle Tuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*.

A Tuxedo application consists of many Tuxedo system processes, one or more application client processes, one or more application server processes, and one or more computer machines connected over a network. In a multi-machine Tuxedo application running different releases of the Oracle Tuxedo software, the *master machine*—designated via the `MASTER` parameter in the `RESOURCES` section of the `UBBCONFIG` file—must run the highest release of the Oracle Tuxedo software in the application. For more information about Tuxedo applications, see "Domains" in *Oracle Tuxedo Product Overview*.

Note: Oracle advises against trying to share the Oracle Tuxedo system executables across remote filesystems; this practice has proven to be unreliable in the past.

System Requirements

The system requirements for Oracle Tuxedo 12*c* (12.1.1.0.0) are given in Table 1-1.

Table 1-1 The System Requirements for Oracle Tuxedo 12c 12.1.1.0.0)

Component	Requirement
Platform	Any platform identified in Oracle Tuxedo 12c (12.1.1.0.0) Platform Data Sheets
Hard disk drive	As stated in the data sheet for the target platform in Oracle Tuxedo 12c (12.1.1.0.0) Platform Data Sheets. For additional information about disk space requirements, see "File and Database Management and Disk Space Allocation" on page C-1.
Memory	As stated in the data sheet for the target platform in Oracle Tuxedo 12c (12.1.1.0.0) Platform Data Sheets.

For Oracle Jolt installation preparatory information, see Oracle Tuxedo 12c Release 1 (12.1.1.0.0) Platform Data Sheets"Oracle Jolt 12c (12.1.1.0.0) Overview and Installation Information" on page C-1. No installation preparation is required for Oracle SNMP Agent.

Note: Before installing Oracle Tuxedo software on a UNIX system, Oracle recommends that you repartition your hard disk device in accordance to "File and Database Management and Disk Space Allocation" on page C-1.

Temporary Storage Space Requirements

The Oracle Installation program uses a temporary directory in which it extracts the files from the archive that are needed to install Oracle Tuxedo on the target system. During the installation process, your temporary directory must contain sufficient space to accommodate the compressed Java Runtime Environment (JRE) bundled with the installer and an uncompressed copy of the JRE that is expanded into the temporary directory. The installation program moves the JRE from the temporary directory to the *Oracle Home directory* at the end of the installation process. For information about the Oracle Home directory, see "Oracle Home Directory" on page 1-7.

The amount of temporary storage space needed depends upon the target platform, as stated in the data sheets in Oracle Tuxedo 12c Release 1 (12.1.1.0.0) Platform Data Sheets.

By default, the installation program uses the temporary directories shown in Table 1-2.

Table 1-2 Temporary Directories

Platform	Directory
UNIX	`/tmp`

To ensure there is adequate temporary space, you may want to allocate an alternate directory for use as a temporary directory for the installation. To do so, perform the appropriate step in Table 1-3 before starting the Oracle Installation program.

Table 1-3 Perform This Step

On This Platform…	Perform This Step…
UNIX	Enter the following command at the shell prompt: `export IATEMPDIR=`*`tmpdirname`* Replace *`tmpdirname`* with the name of a temporary directory of your choice.

Interprocess Communication Resources Configuration

Interprocess communication (IPC) is a capability supported by the Windows and UNIX operating systems that allows one process to communicate with another process. The processes can be running on the same computer or on different computers connected through a network.

On a UNIX system, you use methods native to the UNIX system to adjust the IPC parameters and maximize the performance of a Oracle Tuxedo application. Since most UNIX systems are shipped with default values that are too low for a Oracle Tuxedo application, you need to adjust the IPC parameters, using the methods given in Oracle Tuxedo 12c Release 1 (12.1.1.0.0)

Platform Data Sheets. For the recommended IPC values, see "IPC Resource Configuration on a UNIX System" on page D-1.

After installing the Oracle Tuxedo software and creating an application configuration file (`UBBCONFIG` file), you use the `tmloadcf(1)` command to calculate and print a list of the minimum IPC resources needed to support the application. If your Oracle Tuxedo application is distributed, the minimum IPC resources must be available on every machine participating in the application. For more information about calculating IPC resources using the `tmloadcf(1)` command, see "Performing Post-Installation Tasks" on page 6-1.

Note: Before installing Oracle Tuxedo software on a UNIX system, Oracle recommends that you adjust the IPC parameters on the target machine in accordance to "IPC Resource Configuration on a UNIX System" on page D-1.

Oracle Installation Program

The Oracle Tuxedo software is distributed as an installer file, which also contains a copy of the Oracle Installation program. The Oracle Installation program is the Oracle standard tool for installing the Oracle Tuxedo software on Windows or UNIX systems.

Installation Methods

The Oracle Installation program supports three installation methods. You can use any of these methods to install the Oracle Tuxedo product software:

- Graphical user interface (GUI) installation, described in "Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation" on page 2-1.
- Console Installation (UNIX systems only), described in "Installing Oracle Tuxedo Message Queue on UNIX Systems Using" on page 3-1.
- Silent installation, described in "Installing Oracle Tuxedo Message Queue Using Silent Installation" on page 4-1.

Cancelling Installation

Clicking the "Cancel" or the "close' window button in GUI mode, or typing "quit" in console mode creates an incomplete Tuxedo 12*c* (12.1.1.0.0) installation. You will have to re-install Tuxedo 12*c* (12.1.1.0.0).

If you installed Tuxedo 12*c* (12.1.1.0.0) in a previous Tuxedo product directory, and you want to return to your original configuration, you must re-install your previous Tuxedo version.

Install Sets

An install set is a bundle of product software components related by function. Oracle Tuxedo 12*c* (12.1.1.0.0) offers the following seven install sets:

- *Full install set*—consists of all the Oracle Tuxedo 12*c* (12.1.1.0.0) server and client components
- *Server install set*—consists of all the Oracle Tuxedo 12*c* (12.1.1.0.0) server components:
  - Oracle Application-to-Transaction Monitor Interface (ATMI) server, native ATMI client, and ATMI Workstation (/WS) software
  - Oracle Common Object Request Broker Architecture (CORBA) C++ server
  - Oracle Jolt 12*c* (12.1.1.0.0) server software including Oracle Jolt Internet Relay
  - Oracle SNMP Agent 12*c* (12.1.1.0.0) software
  - Oracle Tuxedo Administration Console software
  - Link-Level Encryption (LLE) and Secure Sockets Layer (SSL) encryption software
- *Full client install set*—consists of all the Oracle Tuxedo 12*c* (12.1.1.0.0) client components:
  - Oracle ATMI Workstation (/WS) client software
  - Oracle CORBA C++ client software (C++ client Object Request Broker, or ORB) including environmental objects
  - Oracle Jolt 12*c* (12.1.1.0.0) client software
  - LLE and SSL encryption software
- *ATMI (/WS) client install set*—consists of the following Oracle Tuxedo 12*c* (12.1.1.0.0) components:
  - Oracle ATMI (/WS) client software
  - LLE and SSL encryption software
- *CORBA client install set*—consists of the following Oracle Tuxedo 12*c* (12.1.1.0.0) components:
  - Oracle CORBA C++ client software
  - SSL encryption software

- *Jolt client install set*—consists of the following Oracle Tuxedo 12*c* (12.1.1.0.0) components:
  - Oracle Jolt 12*c* (12.1.1.0.0) client software

In addition to selecting an install set during a Oracle Tuxedo 12*c* (12.1.1.0.0) installation, a user can further customize the installation by selecting (adding) or deselecting (deleting) one or more software components from the install set. Customizing is only possible for GUI-mode and console-mode installation methods described in "Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation" on page 2-1 and "Installing Oracle Tuxedo Message Queue on UNIX Systems Using" on page 3-1; it is not possible for silent-mode installation method.

OTMQ Install

OTMQ is an add-on chargeable product that enhances ATMI /Q functionality.

- If you select FULL Install *and* OTMQ Install, all TUXEDO *and* OTMQ files are installed.
- If you select Server install *and* OTMQ Install, only TUXEDO server *and* OTMQ server files are installed.
- If you select client install (Full client, ATMI (/WS) client, CORBA client, Jolt client) *and* OTMQ Install, only TUXEDO-related *and* OTMQ client files are installed.

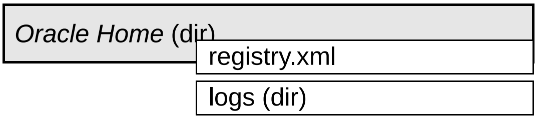
Oracle Home Directory

When you install Oracle Tuxedo, you are prompted to specify an Oracle Home directory. The Oracle Home directory is a repository for common files that are used by multiple Oracle products installed on the same machine. For this reason, the Oracle Home directory can be considered a "central support directory" for the Oracle products installed on your system.

The files in the Oracle Home directory are essential to ensuring that the Oracle software operates correctly on your system. These files perform the following types of functions:

- Facilitate checking of cross-product dependencies during installation
- Facilitate service pack installation

The structure of the Oracle Home directory created by the Oracle Tuxedo 12*c* (12.1.1.0.0) installer program is shown in the following illustration.



Choosing a Oracle Home Directory

During the installation of Oracle Tuxedo 12*c* (12.1.1.0.0), you are prompted to choose an existing Oracle Home directory or specify a path for a new Oracle Home directory. If you choose to create a new directory, the Oracle Tuxedo installer program automatically creates the directory for you.

Upon choosing a Oracle Home directory, you are prompted to choose a Oracle product directory for your Oracle Tuxedo installation. You may choose to create your Oracle product directory under the Oracle Home directory, but there is no requirement to do so.

Understanding the Functions of the Oracle Home Directory

The files and directories in an Oracle Home directory created by the Oracle Tuxedo installer are described in Table 1-4.

Table 1-4 Description

Component	Description
`registry.xml` file	An XML registry file that contains a persistent record of all Oracle products installed on the target system for this Oracle Home directory. The registry contains product-related information, such as version level, service pack level, and product installation directory.
`logs` directory	A directory containing a Oracle Home location file and a history file of installation and uninstallation for this Oracle Home directory. For more information on these files, see the "Understanding the Oracle Tuxedo Message Queue Shortcuts on a Windows System" on page 2-11.

Creating More Than One Oracle Home Directory

Although it is possible to create more than one Oracle Home directory, Oracle recommends that you avoid doing so. In almost all situations, a single Oracle Home directory is sufficient. There may be circumstances, however, in which you prefer to maintain separate development and production environments, each containing a separate product stack. With two directories, you can update your development environment (in an Oracle Home directory) without modifying the production environment until you are ready to do so.

Installation Road Map

You are now ready to begin your installation. To install Oracle Tuxedo 12*c* (12.1.1.0.0), see one of the following sections:

- "Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation" on page 2-1
- "Installing Oracle Tuxedo Message Queue on UNIX Systems Using" on page 3-1
- "Installing Oracle Tuxedo Message Queue Using Silent Installation" on page 4-1

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo 9.1 or earlier software, see "Upgrading the Oracle Tuxedo Message Queue to 12c (12.1.1.0.0)" on page 5-1.

If you want to uninstall your Oracle Tuxedo software, see "Performing Post-Installation Tasks" on page 6-1.

Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation

The following sections describe how to install Oracle Tuxedo Message Queue (OTMQ) using graphical user interface (GUI) mode installation on both Windows and UNIX systems:

- What Is GUI-Mode Installation?
- Before You Start
- Starting GUI-Mode Installation on a UNIX System
- Running GUI-Mode Installation
- Assigning File Ownership on a UNIX System
- What Do I Do Next?

What Is GUI-Mode Installation?

The graphical user interface mode installation is the graphics-based method of executing the Oracle Installation program. It can be run on UNIX system.

To run GUI-mode installation, the console attached to the machine on which you are installing the software must support a Java-based GUI. Not all consoles for UNIX support Java-based GUIs.

Note: To install Oracle Tuxedo Message Queue on a UNIX system with a non-graphics console, see“Installing Oracle Tuxedo Message Queue on UNIX Systems Using” on page 3-1.

Before You Start

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo Message Queue 9.1 or earlier software, see "Upgrading the Oracle Tuxedo Message Queue to 12c (12.1.1.0.0)" on page 5-1 and follow the instructions given there. Then return here to continue your Oracle Tuxedo Message Queue installation.

If you choose to remove the older software, refer to "Upgrading the Oracle Tuxedo Message Queue to 12c (12.1.1.0.0)" on page 5-1, follow the instructions to remove the older software from your machine, and then restart the installation.

If you choose *not* to remove the older software, install the Oracle Tuxedo Message Queue 12*c* (12.1.1.0.0) software to a different product directory than the product directory where the Oracle WebLogic Enterprise software or the older version of the Oracle Tuxedo Message Queue software is installed. See also "Installation Methods" on page 1-5

Starting GUI-Mode Installation on a UNIX System

To start the GUI-mode installation process on a UNIX system, follow these steps:

1. Select a UNIX system that meets the hardware and software requirements described in "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1
2. Log in to the UNIX system as the Oracle Tuxedo Message Queue administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo Message Queue installation.

   For disk space requirements, see "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1. For important disk space allocation information, see "File and Database Management and Disk Space Allocation" on page C-1
4. Install Oracle Tuxedo Message Queue by downloading from the Oracle FTP site:
   a. Download the Oracle Tuxedo 12c (12.1.1.0.0) installation file specific to your platform (Message Queue is an Oracle Tuxedo add-on product).

   b. Go to the directory where you downloaded the installer and invoke the installation procedure by entering the following command:

      prompt> `sh` *`filename`*`.bin`

      where *`filename`* is the name of the Oracle Tuxedo Message Queue installer file.

Note: GUI mode is the default for Tuxedo 12*c* (12.1.1.0.0) installation. If a GUI interface is not available on your Unix platform, you will receive an error message and the installation process is aborted.

If a GUI interface is not available on your Unix platform, you must use silent or console-mode installation. For more information, see What Is Silent Installation? and What Is Console-Mode Installation?

5. Proceed to "Running GUI-Mode Installation" on page 2-3.

Running GUI-Mode Installation

The Oracle Tuxedo Message Queue installer program is currently available in English and Japanese. By checking the value of a certain language-related environment variable on the target platform, the installer program automatically determines (1) the language displayed during the installation and (2) the language-specific message catalog installed during the installation. If the language-related environment variable is not set or is set to a value unknown to the installer program, both the language displayed and the message catalog installed default to English.

The Oracle Tuxedo Message Queue installer program prompts you to enter specific information about your system and configuration. For instructions on responding to the prompts during installation, see Table 2-1.

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Introduction	Click Next to proceed with the installation. You may cancel the installation at any time by clicking Cancel.
Choose Install Set	Select the install set that you want installed on your system. The following seven choices are available: • Full Install (the default)—all Tuxedo server and client software components • Server Install—Tuxedo server software components only • Full Client Install—Tuxedo client software components only • Jolt Client Install—Jolt client software components only • ATMI (/WS) Client Install—Tuxedo ATMI client software components only • CORBA Client Install—Tuxedo CORBA client software components only • Custom Install—select specific Tuxedo server and client software components. The following table entry provides a summary of options for the Custom Install. For a detailed list of software components for each install set, see "Install Sets" on page 1-6. Select an install set and then click Next to launch the Choose Oracle Home Directory window—your selection includes the appropriate encryption software for Link-Level Encryption (LLE) and/or Secure Sockets Layer (SSL). Or select an install set and then click Customize to launch the Customize Install window.

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Customize Install	Select (add) or deselect (clear) one or more software components from the selected install set, or choose one of the other five install sets or Custom Set from the drop-down list menu and customize its software components. For a description of the JRLY component, see "Jolt Internet Relay" on page C-5. Observe the following software component mappings: • *Server*—contains ATMI server software; CORBA C++ server software; Oracle Jolt server software; Oracle SNMP Agent software, and Oracle Tuxedo Message Queue Administration Console software • *ATMI Client*—contains Oracle ATMI Workstation (/WS) client software • *CORBA Client*—contains Oracle CORBA C++ client software (C++ client ORB) including environmental objects • *Jolt JRLY*—contains Oracle Jolt Relay software • *Jolt Client*—contains Oracle Jolt client software After selecting or deselecting one or more software components from the selected install set, click Next to continue with the installation. The appropriate encryption software for LLE and/or SSL is automatically included.
OTMQ Install	OTMQ is an add-on chargeable product. By default it is not installed.
Choose Oracle Home Directory	Specify the Oracle Home directory that will serve as the central support directory for all Oracle products installed on the target system. If you already have a Oracle Home directory on your system, you can select that directory (recommended) or create a new Oracle Home directory. If you choose to create a new directory, the Oracle Tuxedo Message Queue installer program automatically creates the directory for you. For details about the Oracle Home directory, see "Oracle Home Directory" on page 1-7. Choose a Oracle Home directory and then click Next to continue with the installation.

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Choose Product Directory	Specify the directory in which you want to install the Oracle Tuxedo Message Queue software. You may select the default product directory (`tuxedo12c UNIX`) or create a new product directory. If you choose to create a new directory, the Oracle Tuxedo Message Queue installer automatically creates the directory for you. **WARNING:** Running a previous Tuxedo version in conjunction with Tuxedo 12*c* (12.1.1.0.0) on a Windows system requires that the product directories for both versions *do not have the same name*. Choose a product directory and then click Install to start the Oracle Tuxedo Message Queue software installation.
Install Samples	Click the check box to install samples. Click Next to proceed with the installation.
Pre-Installation Summary	Review your installation selections. Clicking Previous allows you to modify installation selections. Otherwise click Install to continue.
Installing Tuxedo. . .	No user input is required here. The installation program is installing Oracle Tuxedo Message Queue in the user-specified product directory. **Note:** It is normal for the installation progress bar to stop for several minutes, especially at the end. The Oracle Tuxedo Message Queue installer is still working when this occurs.

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
Configure tlisten Service	This window appears only if you have installed the Server Install set. Set the tlisten port value. Enter a `tlisten` password of your choice. Your password must be a string of alphanumeric characters in clear-text format that is no more than 80 characters in length. You use this password to log in to the Oracle Tuxedo Message Queue Administration Console. **Note:** Machines on the same domain must have the same tlisten password defined. Oracle Tuxedo Message Queue uses the `tlisten` password to protect the local machine from administrative requests and operations that are *not* authorized. Whenever administrative communications arrive on the local machine through `tlisten(1)` or `wlisten(1)` gateway processes, Oracle Tuxedo Message Queue authenticates them by means of the `tlisten` password. Enter and re-enter a `tlisten` password and then click Next to continue.
Tlisten Encryption Method	Specify the tlisten encryption method: LLE or SSL. The default is LLE.
SSL Parameter Input	If the SSL encryption method is chosen, you must enter Security Principal name, location, and password.
Choose Encryption Bit	Specify tlisten minimum and maximum encryption bit. The default minimum value is 0. The default maximum value is 256.

Table 2-1 Instructions

In This Window . . .	Perform the Following Action . . .
SSL Installation Choice	This window appears only if you have installed Server, ATMI and/or CORBA Client install set components. Oracle Tuxedo Message Queue provides a Lightweight Directory Access Protocol (LDAP) based certificate retrieval mechanism that supports certificate-based authentication when using SSL. The Oracle Tuxedo Message Queue certificate retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server. Choose whether or not you want to install SSL support. If you intend to use SSL encryption in your application, select Yes and then click Next to launch the LDAP Configuration window. If you do *not* intend to use SSL encryption in your application, select No.
LDAP Configuration for SSL Support	Enter the following LDAP configuration information and then click Next to continue: • LDAP Service Name: Fully qualified domain name of the LDAP server (for example, `pcwiz.mydomain.com`) • LDAP PortID: Port number through which the local machine communicates with the LDAP server (for example, `389`) • LDAP BaseObject: Distinguished name of the base object for search in the LDAP server (for example, `o=beasys.com`) • LDAP Filter File Location: (*`tux_prod_dir`*`/udataobj/security/bea_ldap_filter.dat` by default, where *`tux_prod_dir`* represents the product directory in which you installed the Oracle Tuxedo Message Queue software)
Install Complete	Click Done to exit the installation program.

Congratulations! Your installation of the Oracle Tuxedo Message Queue software is complete!

Assigning File Ownership on a UNIX System

If you installed the Oracle Tuxedo Message Queue software on a UNIX system, Oracle strongly recommends that you create a separate user account for the Oracle Tuxedo Message Queue administrator and give ownership of the Oracle Tuxedo Message Queue files to that account. To change ownership of the Oracle Tuxedo Message Queue software on a UNIX machine, enter the following command:

```
chown -R adm_account full_path_of_tux_prod_dir
chgrp -R adm_group full_path_of_tux_prod_dir
```

Replace *adm_account* with the Oracle Tuxedo Message Queue administrator account, replace *adm_group* with the Oracle Tuxedo Message Queue administrator group ID, and replace *full_path_of_tux_prod_dir* with the full pathname of the product directory in which you installed the Oracle Tuxedo Message Queue software.

Installing Oracle Tuxedo Message Queue on UNIX Systems Using Console-Mode Installation

The following sections describe how to install Oracle Tuxedo Message Queue (OTMQ) using console-mode installation:

What Is Console-Mode Installation?

Console-mode installation is the text-based method of executing the Oracle Installation program. It can be run only on UNIX systems and is intended for UNIX systems with non-graphics consoles. Console-mode installation offers the same capabilities as graphics-based installation

Before You Start

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo Message Queue 9.1 or earlier software, see "Upgrading the Oracle Tuxedo Message Queue to 12c (12.1.1.0.0)" on page 5-1 and follow the instructions given there. Then return here to continue your Oracle Tuxedo Message Queue installation.

Starting Console-Mode Installation

To start the console-mode installation process, follow these steps:

1. Select a UNIX system that meets the hardware and software requirements described in "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1.
2. Log in to the UNIX system as the Oracle Tuxedo Message Queue administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo Message Queue installation.

   For disk space requirements, see "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1. For important disk space allocation information, see "File and Database Management and Disk Space Allocation" on page C-1.
4. Install Oracle Tuxedo Message Queue by downloading from the Oracle FTP site:
   1. Download the Oracle Tuxedo 12c (12.1.1.0.0) installation file specific to your platform (Message Queue is an Oracle Tuxedo add-on product).
   2. Go to the directory where you downloaded the installer and invoke the installation procedure by entering the following command:

      ```
      prompt> sh filename.bin -i console
      ```

      where *filename* is the name of the Oracle Tuxedo Message Queue installer file.
5. Proceed to "Running Console-Mode Installation" on page 3-3.

Running Console-Mode Installation

To complete the console-mode installation process, which is currently available in English and Japanese, respond to the prompts in each section with one of the following actions.

- Enter the number associated with your choice or press `<Enter>` to accept the default.
- Enter `back` or `previous` at the prompt to review or change your selection.
- Enter `quit` in response to any prompt to exit the installation process.

In This Section . . .	Perform the Following Action . . .

Choose Locale	Choose a language by entering the number associated with that language. Your choice determines both the language displayed during the installation and the language-specific message catalog installed during the installation.
	`====================================================`
	`Choose Locale...`
	`  ->1- English`
	`CHOOSE LOCALE BY NUMBER:`
	`====================================================`
	Launching console-mode installation on a system that utilizes a Japanese language environment adds a second choice, 2- Japanese, at the `CHOOSE LOCALE BY NUMBER` prompt.

Introduction	Read the Introduction and press Enter to continue with the installation process.

Choose Install Set

Select the install set that you want installed on your system. The following choices appear:

```
=====================================================

Choose Install Set
------------------
Please Choose the Install Set to be installed by this
installer.

  ->1- Full Install
    2- Server Install
    3- Full Client Install
    4- Jolt Client Install
    5- ATMI Client Install
    6- CORBA Client Install

    7- Customize ...
ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER>
TO ACCEPT THE DEFAULT:
=====================================================
```

For a list of software component for each install set, see "Install Sets" on page 1-6.

Your selection includes the appropriate encryption software for Link-Level Encryption (LLE) and/or Secure Sockets Layer (SSL).

Custom Install

Select custom features.

```
====================================================

Choose Product Features
------------------
Enter a comma-separated list of numbers representing the
features you want to select or deselect. To view a
feature’s description, enter ‘?<number>’. Press <RETURN>
when you are done.

    1-[X] Server
    2-[X] ATMI Client
    3-[X] CORBA Client
    4-[X] Jolt JRLY
    5-[X] Jolt Client

SELECT FEATURES FOR CUSTOM INSTALLATION:
====================================================
```

When you first see the Custom Install screen, all Tuxedo components are selected. Entering a number or series of numbers deselects the components.

Note: Entering `1,2,3,4,5` and pressing `<RETURN>` deselects all the Tuxedo install components.

If you `press <RETURN>` and enter `‘back’` to return to the Custom Install screen, all `‘X’`s are removed. You must enter a number to select a component.

Need To Select a Component

If you did not select a component, this screen appears.

```
====================================================

WARNING: Product Feature Selection
----------------------------------
You must select a product feature,
otherwise installation will not continue
  ->1- Re-Select
    2- Quit
====================================================
```

Note: Selecting `‘1- Re-Select’` returns you to the Custom Install screen and all `‘X’`s are removed. You must enter a number to select a component.

OTMQ install	OTMQ Installation Choice == OTMQ is a add-on chargeable product. OTMQ Installation Choice ------------------------------ Would you like to install OTMQ ? 1- Yes ->2- No ENTER THE NUMBER FOR YOUR CHOICE, SELECTION, OR PRESS <ENTER> TO ACCEPT THE DEFAULT: ==

Choose Oracle Home Directory

Specify the Oracle Home directory that will serve as the central support directory for all Oracle products installed on the target system. If you already have a Oracle Home directory on your system, you can select that directory (recommended) or create a new Oracle Home directory. If you choose to create a new directory, the Oracle Tuxedo Message Queue installer program will automatically create the directory for you. For details about the Oracle Home directory, see “Oracle Home Directory” on page 1-7.

Enter 1 to create a new Oracle Home directory or, if you already have a Oracle Home directory on your system, enter 2 to select an existing Oracle Home directory. Be sure to use the full pathname when specifying a new Oracle Home directory.

For example:

```
====================================================

Choose Oracle Home
-------------------------
    1- Create a New Oracle Home
    2- Use Existing Oracle Home
Enter a number: 2
    1- /home3/oracle
    2- /home2/oraclehome
Existing Oracle Home: 1
====================================================
```

In this example, we enter 2 to display the Oracle Home directories that have already been created on this system. At the Existing Oracle Home prompt, we enter 1 to specify /home3/Oracle as the Oracle Home directory for this installation. Make sure that you enter the number associated with the Oracle Home directory instead of the directory name.

If a previous Tuxedo installation is in the existing directory you selected, the following screen appears:

```
====================================================

A previous installation of Tuxedo 12c
has been detected at /home3/Oracle.
The following options are available:
    1- Continue to install over /home3/oracle
    2- Cancel to select another Oracle Home Directory:

CHOOSE A NUMBER:
====================================================
```

Choose Product Directory

Specify the directory in which you want to install the Oracle Tuxedo Message Queue software. You may select the default product directory (`tuxedo12c`) or create a new product directory. If you choose to create a new directory, the Oracle Tuxedo Message Queue installer will automatically create the directory for you.

The initial default installs the product directory under the Oracle Home directory you specified in the previous section. To accept this default, enter `2` at the prompt.

The choices displayed in this section are as follows:

```
===================================================
Choose Product Directory
------------------------
    1- Modify Current Selection
(/home3/oracle/tuxedo12c)
    2- Use Current Selection (/home3/oracle/tuxedo12c)
Enter a number:
===================================================
```

- Enter `1` to select an alternate directory. The following text appears:

  ```
  Specify Product Installation Directory:
  ```

  Specify a product directory using its full pathname. For example:

  `/home3/mytux10g.R3`.

  When you press Enter, your modified selection appears as the default.

  ```
      1- Modify Current Selection (/home3/mytux10g.R3)
      2- Use Current Selection (/home3/mytux10g.R3)
  Enter a number:
  ```

- Enter `2` to accept the current selection.

  If you enter `2` at the initial prompt, you accept the default product directory (`/home3/oracle/tuxedo12c` in this example).

Install Samples

```
Enter “Y” to install sample
Enter “N” not to install samples
```

Pre-Installation Summary

```
Review your installation selections. Enter back or
previous to modify installation selection. Otherwise
press <ENTER> to continue.
====================================================

Pre-Installation Summary
-------------------------
Please Review the Following Before Continuing:
Product Name:
    Tuxedo 12c
Install Folder:
    /home3/oracle/tuxedo12c
Link Folder:
    /home3/oracle
Disk Space Information (for Installation target):
    Required:  303,288,858 bytes
    Available: 19,094,890,496 bytes

PRESS <ENTER> TO CONTINUE:
====================================================
```

Ready To Install

The choices displayed in this section are as follows:

```
====================================================
Ready To Install
------------------------
InstallAnywhere is now ready to install Tuxedo 12c onto
your system at the following location:
    /home3/oracle/tuxedo12c

PRESS <ENTER> TO INSTALL:
====================================================
```

Installing . . .

No user input is required here. The installation program is installing Oracle Tuxedo Message Queue in the user-specified product directory.

It is normal for the installation progress bar to stop for a fairly long time, especially at the end. The Oracle Tuxedo Message Queue installer is still working when this occurs.

<table>
<tr>
<td>Configure tlisten Service</td>
<td>

Enter a `tlisten` password of your choice. Your password must be a string of alphanumeric characters in clear-text format that is no more than 80 characters in length. You use this password to log in to the Oracle Tuxedo Message Queue Administration Console.

Oracle Tuxedo Message Queue uses the `tlisten` password to protect the local machine from administrative requests and operations that are *not* authorized. Whenever administrative communications arrive on the local machine through `tlisten(1)` or `wlisten(1)` gateway processes, Oracle Tuxedo Message Queue authenticates them by means of the `tlisten` password.

```
===================================================

Configure tlisten Service
-------------------------
Password:
Verify Password:
Good Password! Press "Enter" to continue.
===================================================
```

</td>
</tr>
<tr>
<td>SSL Installation Choice</td>
<td>

```
===================================================

SSL Installation Choice
------------------------
Would you like to install SSL Support?
    ->1- Yes
      2- No
ENTER THE NUMBER FOR YOUR CHOICE, SELECTION, OR PRESS
<ENTER> TO ACCEPT THE DEFAULT:
===================================================
```

</td>
</tr>
</table>

Enter Your LDAP Settings for SSL Support

This section appears only if you have installed CORBA server and/or client software components. Oracle Tuxedo Message Queue provides a Lightweight Directory Access Protocol (LDAP) based certificate retrieval mechanism that supports certificate-based authentication when using SSL. The Oracle Tuxedo Message Queue certificate retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.

```
===================================================

Enter Your LDAP Settings for SSL Support
------------------------
LDAP Service Name: a
LDAP PortID: b
LDAP BaseObject: c
LDAP Filter File Location:
Press <ENTER> to accept the default,
/home3/oracle/tuxedo12c/
  udataobj/security/bea_ldap_filter.dat
or enter your own:
===================================================
```

Installation Complete

```
===================================================

Installation Complete
---------------------
Congratulations.Tuxedo 12c has been successfully
installed to:
    /home3/oracle/tuxedo12c
PRESS <ENTER> TO EXIT THE INSTALLER.
===================================================v
```

Congratulations! Your installation of the Oracle Tuxedo Message Queue software is complete!

What Do I Do Next?

To prepare for the post-installation tasks, see “Assigning File Ownership on a UNIX System” on page 2-8. To configure your Oracle Tuxedo Message Queue software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo software, see “Performing Post-Installation Tasks” on page 6-1.

Installing Oracle Tuxedo Message Queue Using Silent Installation

The following sections describe how to install Oracle Tuxedo Message Queue (OTMQ) using silent installation on UNIX systems:

What Is Silent Installation?

Silent installation reads the settings for your configuration from a text file that you create prior to beginning the installation. Manual intervention is not necessary during the installation process. Silent installation works on UNIX systems.

Silent installation is a way of setting installation configurations only once and then using those configurations to duplicate the installation on many machines.

Note: Using silent installation implies your consent to the License Agreement. You neither see a copy of the Software License Agreement nor have any means to accept the terms of the agreement.

Before You Start

If you are upgrading from Oracle WebLogic Enterprise or Oracle Tuxedo Message Queue 9.1 or earlier software, see "Upgrading the Oracle Tuxedo Message Queue to 12c (12.1.1.0.0)" on page 5-1 and follow the instructions given there. Then return here to continue your Oracle Tuxedo Message Queue installation.

If you are planning to use Oracle Tuxedo Message Queue 12*c* (12.1.1.0.0) in conjunction with a previous Tuxedo version on a Windows system, see "Oracle Installation Program" on page 1-5.

Using Silent Installation: Main Steps

The silent installation process has two primary steps:

1. Create a template file that contains the configuration settings, such as Oracle Home directory, product directory, and install set appropriate for your installation.

   For the detailed procedure, see "Creating a Template File" on page 4-3. A sample template file is provided in "UNIX Template File" on page 4-7.

2. Invoke the installation process using the values specified in the template files.

   For the detailed procedure, see "Invoking the Silent Installation Process on a Windows System" on page 4-6 and "Invoking the Silent Installation Process on a UNIX System" on page 4-6.

Creating a Template File

To create a template file for use in the silent installation process, follow these steps:

1. Display the template file specific to your platform. The following template is available in:
   - UNIX template file content at “UNIX Template File” on page 4-7.
2. Copy the contents of the template file and save it as a text file named `installer.properties`.
3. In the `installer.properties` file, modify the values for the keywords shown in the following table to create your desired configuration.

The installer.properties file is shown in Table 4-1

Table 4-1 The installer.properties File

For This Keyword . . .	Enter the Following Value . . .
`INSTALLER_UI=`	The mode of installation. The default is `silent`; do not modify this value.
`ORACLEHOME=`	The full pathname of the Oracle Home directory of your choice. For details about the Oracle Home directory, see “Oracle Home Directory” on page 1-7.
`USER_INSTALL_DIR=`	The full pathname of the product directory of your choice. **WARNING:** Running a previous Tuxedo version in conjunction with Tuxedo 12*c* (12.1.1.0.0) on a Windows system requires that the product directories for both versions *do not have the same name.*
`INSTALL_SAMPLES`	`“yes”` installs Tuxedo 12c samples. `“no”` does not install samples.
`LDAP_HOSTNAME=`	The fully qualified domain name of the Lightweight Directory Access Protocol (LDAP) server of your choice; for example, `pcwiz.mydomain.com`. (See Note at end of table.)

Table 4-1 The installer.properties File (Continued)

For This Keyword . . .	Enter the Following Value . . .
`LDAP_PORTID=`	The port number through which the local machine communicates with the LDAP server; for example, `389`.* (See Note at end of table.)
`TLISTEN_PORT`	The port number through which the local machine communicates with the TLISTEN server; for example, 1357.
`LDAP_BASE_OBJECT=`	The distinguished name of the base object for search in the LDAP server; for example, `o=beasys.com`. (See Note at end of table.)
`LDAP_FILTER_FILE=`	The full pathname of the LDAP filter file on your machine; for example, `C:\oracle\tuxedo12c\udataobj\security\bea_ldap_filter.dat`. (See Note at end of table.)
`TLISTEN_PASSWORD=`	A `tlisten` password of your choice. The password must be a string of alphanumeric characters in clear-text format that is no more than 80 characters in length. You will be prompted to enter this password when logging in to the Oracle Tuxedo Message Queue Administration Console.

Table 4-1 The installer.properties File (Continued)

For This Keyword . . .	Enter the Following Value . . .
`CHOSEN_INSTALL_SET=`	The chosen installation set. You must set this value to one of the following options: **Notes:** The valid options are case sensitive and should be coded exactly as defined below for proper installation. Use of invalid options will result in improper installation. Spaces should not be included after the install set name. If there are spaces after the install set name, the installation process uses the default mode `Full Install`. • `Full` for a full installation • `Server` for server-only installation • `Client` for full client installation • `ATMI` for ATMI-client-only installation, • `CORBA` for CORBA-client-only installation • `Jolt` for Jolt-client-only installation No customizing of install sets is possible with silent-mode installation. For a description of install sets, see "Install Sets" on page 1-6.
`INSTALL_QPP=`	"yes" installs OTMQ. "no" does not install OTMQ.

Notes: Keywords: `LDAP_HOSTNAME`, `LDAP_PORTID`, `LDAP_BASE_OBJECT`, `LDAP_FILTER_FILE` are optional.

Include these keywords and their values only if you want to use Secure Sockets Layer (SSL) encryption. Otherwise, comment these lines out by preceding them with hash marks (#).

Keywords: `TLISTEN_PORT`, and `TLISTEN_PASSWORD are also optional`

Include these keywords and their values only if you select `Server` (server-only installation).

In the template files, comment lines are preceded by hash marks (#). The installer program does not read any line beginning with a hash mark.

Invoking the Silent Installation Process on a UNIX System

Installing the software using silent installation takes the same amount of time as a standard installation. During silent installation, a startup message appears followed by an "Installing . . ." message, indicating that the installation has started.

To invoke the silent installation process on a UNIX system, follow these steps:

1. Select a UNIX system that meets the hardware and software requirements described in "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1.
2. Log in to the UNIX system as the Oracle Tuxedo Message Queue administrator.
3. Ensure that you have enough free space for the Oracle Tuxedo Message Queue installation.

   For disk space requirements, see "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1. For important disk space allocation information, see "File and Database Management and Disk Space Allocation" on page C-1.
4. Install Oracle Tuxedo Message Queue by downloading from the Oracle FTP site:

   a. Download the Oracle Tuxedo 12c (12.1.1.0.0) installation file specific to your platform (Message Queue is an Oracle Tuxedo add-on product).

   b. Go to the directory where you downloaded the installer and invoke the installation procedure by entering the following command:

      prompt> `sh` *`filename`*`.bin -f` *`path/`*`installer.properties`

      where *`filename`* is the name of the Oracle Tuxedo Message Queue installer file, and *`path`* is the full or relative pathname of the `installer.properties` file.

      Note: If a file named `installer.properties` is present in the same directory as the installer (for example, `tuxedo81_sol.bin`), it will be supplied automatically in silent mode. However, if `installer.properties` is in a different directory, you may use the `-f` command line option to specify the properties file. For example, *`filename.exe`* `-f` *`path/`*`installer.properties`
5. To verify that the software installed successfully, see "Performing Post-Installation Tasks" on page 6-1.

UNIX Template File

This sample UNIX template file (`installer.properties`) in Listing 4-1 applies to a silent installation of Oracle Tuxedo Message Queue 12*c* (12.1.1.0.0).

Listing 4-1 UNIX installer.properties Template File

```
INSTALLER_UI=silent
### Locale ###
USER_LOCALE=en
### Installation Mode ###
### Input New Install ###
INSTALL_MODE=New Install
### Oracle Home Directory ###
ORACLEHOME=/home/user/oracle/tuxedo12c
### Product Installation Directory ###
USER_INSTALL_DIR=/home/user/oracle/tuxedo12c1
### LDAP Service Name ###
LDAP_HOSTNAME=pcwiz.mydomain.com
### LDAP PortID ###
LDAP_PORTID=389
### LDAP BaseObject ###
LDAP_BASE_OBJECT="o=beasys.com"
### LDAP Filter File Location ###
LDAP_FILTER_FILE=/home/user/oracle/tuxedo12c/udataobj/security/
bea_ldap_filter.dat
### Tlisten Password ###
TLISTEN_PASSWORD=luckluck
### Valid Install Sets, select one:###
### CHOSEN_INSTALL_SET=Full###
### CHOSEN_INSTALL_SET=Server###
### CHOSEN_INSTALL_SET=Client###
### CHOSEN_INSTALL_SET=ATMI###
### CHOSEN_INSTALL_SET=CORBA###
### CHOSEN_INSTALL_SET=Jolt###
### NOTE: Spaces should not be included after the install set name. ###
### If there are spaces after the install set name, the installation ###
### process uses the default mode Full Install. ###
CHOSEN_INSTALL_SET=Full Install
CHOSEN_INSTALL_SET=Full Install
### Install Samples Y/N###
INSTALL_SAMPLES=Yes
### Install OTMQ yes/no###
INSTALL_QPP=yes
```

What Do I Do Next?

To prepare for the post-installation tasks, read the following sections:

- "Assigning File Ownership on a UNIX System" on page 2-8

To configure your Oracle Tuxedo Message Queue software, verify that your software is installed correctly, or to uninstall Oracle Tuxedo software, see "Performing Post-Installation Tasks" on page 6-1.

Upgrading the Oracle Tuxedo Message Queue to 12*c* (12.1.1.0.0)

The following sections provide procedures for upgrading your Oracle Tuxedo or Oracle WebLogic Enterprise application to Oracle Tuxedo 12*c* (12.1.1.0.0) using simple and hot upgrade procedures:

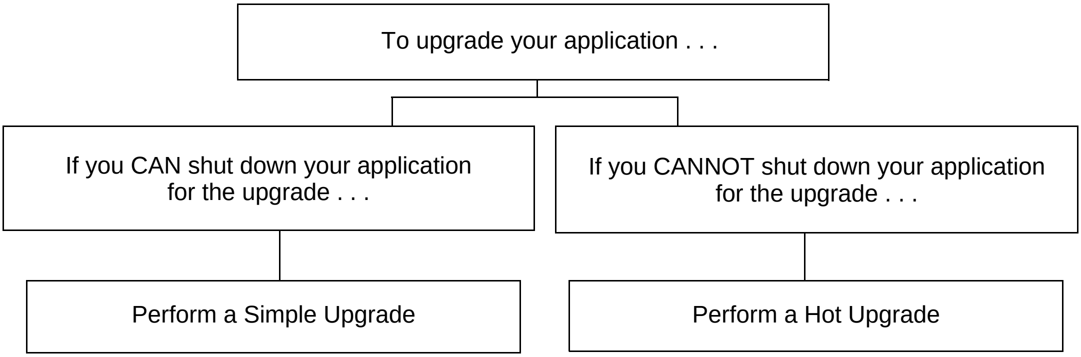
- Preparing Your Machine for an Upgrade
- Selecting an Upgrade Procedure
- Backing Up Files
- Performing a Simple Upgrade
- Upgrading from Tuxedo 6.5 to Tuxedo 12c (12.1.1.0.0)
- Performing a Hot Upgrade
- Rebuilding an Application
- Installing Oracle Jolt 12c (12.1.1.0.0) with Oracle WebLogic Server

Preparing Your Machine for an Upgrade

Before you install the Oracle Tuxedo 12*c* (12.1.1.0.0) software on a machine, make sure that the required version of the operating system and the compiler have been installed. For operating system and compiler requirements for each supported platform, see "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1.

Selecting an Upgrade Procedure

If you are installing Oracle Tuxedo 12*c* (12.1.1.0.0) on a platform that contains an earlier release of Oracle Tuxedo software or the Oracle WebLogic Enterprise software, be sure to read the following procedures carefully.


To upgrade your application . . .
If you CAN shut down your application for the upgrade . . .
If you CANNOT shut down your application for the upgrade . . .
Perform a Simple Upgrade
Perform a Hot Upgrade

If you can shut down your application for the upgrade, then shut down your application and perform a simple upgrade. For instructions on performing a simple upgrade, see "Performing a Simple Upgrade" on page 5-3.

If you cannot shut down your application for the upgrade, you can perform a hot upgrade for any of the following software products to Oracle Tuxedo 12*c* (12.1.1.0.0) without having to shut down your enterprise:

- Versions 6.5, 7.1, 8.0, 8.1, 9.0, 9.1, 10.0, 10.3, 11.1.1.1.0 of Oracle Tuxedo
- Version 5.1 of WebLogic Enterprise

You can add the Oracle Tuxedo 12*c* (12.1.1.0.0) software to existing Oracle Tuxedo or Oracle WebLogic Enterprise domains without shutting down the existing applications, and you can add new Oracle Tuxedo 12*c* (12.1.1.0.0) application servers to those domains without shutting down the existing applications. For instructions on performing a hot upgrade, see "Performing a Hot Upgrade" on page 5-4.

Backing Up Files

There are certain files that you need to back up prior to the installation and then restore after the installation is complete. Any modifications that you made to these files will be overwritten when the new Oracle Tuxedo software is installed.

To avoid having to modify these files again, follow these steps:

1. Back up the `RM` file to a temporary location. The `RM` file contains database vendor-specific settings that are used by commands such as `buildtms` and `buildXAJS`. It is located in the *`tux_prod_dir`*`/udataobj` directory, where *`tux_prod_dir`* is the directory in which you installed the Oracle Tuxedo or Oracle WebLogic Enterprise software.
2. Back up the `tuxwsvr.ini` and `webgui.ini` files to a temporary location. These files contain customized initialization settings for the Oracle Tuxedo Administration Console server and GUI. The `tuxwsvr.ini` file is located in the *`tux_prod_dir`*`/udataobj` directory, and the `webgui.ini` file is located in the *`tux_prod_dir`*`/udataobj/webgui` directory.
3. Back up the `CATNAMES`, `jrepository`, and `jrly.config` files. The `CATNAMES` file is located in the *`tux_prod_dir`*`/locale` directory. The `jrepository` and `jrly.config` files are located in the *`tux_prod_dir`*`/udataobj/jolt` directory structure.
4. Move any files that you or your coworkers added to the *`tux_prod_dir`* directory to a temporary location.
5. After the installation is complete, restore these files to their original locations.

Performing a Simple Upgrade

A simple upgrade consists of the following basic steps:

1. Shut down the application.
2. Back up files as described in "Backing Up Files" on page 5-3.
3. Update the text version of the configuration file on the `MASTER` machine by running the `tmunloadcf(1)` command.
4. Back up the old binary version of the configuration file.
5. If upgrading a Oracle Tuxedo 9.1 machine, uninstall the Oracle Tuxedo system software as described in "Uninstalling Oracle Tuxedo" on page 6-25.

6. Install the Oracle Tuxedo 12*c* (12.1.1.0.0) software on at least the `MASTER` and backup `MASTER` machines.

7. Reload the text version of the configuration file by running `tmloadcf(1)` on the `MASTER` machine.

8. If necessary, recompile and relink your application programs.

9. Reboot the application.

10. As time permits, shut down and upgrade the other machines in your configuration.

Upgrading from Tuxedo 6.5 to Tuxedo 12*c* (12.1.1.0.0)

If you are upgrading from Tuxedo 6.5 to Tuxedo 12*c* (12.1.1.0.0), please note the following:

- In order to allow a Tuxedo 12*c* (12.1.1.0.0) server to interoperate with Tuxedo 6.5 software, the Tuxedo 12*c* (12.1.1.0.0) server `servopts -t` option must be specified. The server may be a workstation listener (WSL) process (where the `servopts -t` option allows interoperability for all of its workstation handler—WSH—processes), a domain gateway (GWTDOMAIN) process, or a system or application server process. For more information, see `GWTDOMAIN(5)` in the *Oracle Tuxedo File Formats, Data Descriptions, MIBs, And System Processes Reference*.

- In order to allow a domain gateway (GWTDOMAIN) in Tuxedo 12*c* (12.1.1.0.0) to interoperate with a domain gateway running on Tuxedo 6.5. you must upgrade to Tuxedo 6.5 patch level 446.

  If you have not upgraded to patch level 446, then you must set the `TM_GWT_OLDSECCHECK` environment variable of the domain gateway in this Tuxedo 12*c* (12.1.1.0.0). For more Oracle Tuxedo environment variable information, see `tuxenv(5)` in the *Oracle Tuxedo File Formats, Data Descriptions, MIBs, And System Processes Reference*.

- If your Tuxedo 6.5 COBOL program calls `TPINITIALIZE()`, you must set the `TM_CBL_IGNORE_CONTEXT` environment variable to `"Y"` when you run the COBOL application. For more Oracle Tuxedo environment variable information, see `tuxenv(5)` in the *Oracle Tuxedo File Formats, Data Descriptions, MIBs, And System Processes Reference*.

Performing a Hot Upgrade

Before performing a hot upgrade, back up files as described in "Backing Up Files" on page 5-3.

During the hot upgrade, you must shut down the MASTER and backup MASTER machines *serially*. The remainder of the configuration is unaffected.

Pre-Upgrade Notes

Once you have finished migrating the MASTER to Oracle Tuxedo 12*c* (12.1.1.0.0) (via a hot upgrade), you will not be able to migrate it back to the acting backup until you have upgraded the backup to Oracle Tuxedo 12*c* (12.1.1.0.0). In other words, migrating the MASTER is an irreversible procedure; once you have completed it, you cannot change the MASTER back to the release of the Oracle Tuxedo software on which it was running previously.

When you run `tmadmin(1)` on a Oracle Tuxedo 12*c* (12.1.1.0.0) machine with an old-release MASTER, the UPGRADE file shown in "Hot-Upgrade Example" on page 5-6 must reside in the current directory. To make sure that the files are in the current directory, you might, for example, run `tmadmin` from the Tuxedo application directory: %APPDIR% (Windows)/ $APPDIR (UNIX) directory.

During migration, keep in mind the following guidelines regarding the Oracle Tuxedo data storage areas:

- You will be instructed to remove the old TUXCONFIG file. The TUXCONFIG file is propagated automatically when the backup machine is run; therefore, you do not need to do anything.
- Create the TLOG when you load the software. If you already have a TLOG, remove it.
- Do not do anything with application queue spaces and queues.
- If you have a Domains configuration (BDMCONFIG) file, leave it as is.

To minimize downtime, Oracle recommends installing Oracle Tuxedo 12*c* (12.1.1.0.0) on the MASTER machine and backup MASTER, and rebuilding all clients and servers in the directory on each machine that is defined by %APPDIR%/ $APPDIR, in parallel with the directory in which the old-release clients and servers were created. In other words, both an old-release application and a Oracle Tuxedo 12*c* (12.1.1.0.0) application should be available on each machine. *This advice is not a requirement, but a recommendation.*

The Oracle Tuxedo software supports device list path entries of up to 256 bytes in length (in previous releases the limit was 64 bytes in length.) Queue Spaces, TLOG files, and other configuration files originally created with an earlier version of Oracle Tuxedo are understood and supported with this release. However, such device lists originally created with Oracle Tuxedo 8.0 or earlier are restricted to 64-byte path names. Therefore, if the administrator wants to add a new extent using a path name greater than 64 bytes in length to a queue space in the future and such queue space is empty when the upgrade is performed, the administrator may want to create the queue space device list and the queue space at the time of the hot upgrade. Many applications need to preserve queue spaces across an upgrade and will not be able to recreate them. Furthermore, most applications will not need to add queue space extent in the future or will be able to do so using path names no longer than the ones they are currently using. *This queue space recreation is completely optional and mentioned for completeness.*

Hot-Upgrade Procedure

This section provides an overview of the tasks for a “hot upgrade.”

1. Make sure your environment is set up as follows:
   – The configuration must have a backup `MASTER` machine.
   – The `MODEL` option in the configuration must be set to `MP`.
   – The `MIGRATE` option in the configuration must be set.
2. If upgrading a Oracle Tuxedo 9.1 machine, uninstall the Oracle Tuxedo system software as specified in “Uninstalling Oracle Tuxedo” on page 6-25.
3. Shut down the backup `MASTER` machine.
4. Install the Oracle Tuxedo 12*c* (12.1.1.0.0) software on the backup `MASTER` machine and reboot the backup machine.
5. Migrate the `MASTER` machine running the old release to the backup `MASTER` machine running Oracle Tuxedo12*c* (12.1.1.0.0).
6. Shut down the machine that is now acting as the backup `MASTER`.
7. Install Oracle Tuxedo 12*c* (12.1.1.0.0) on the old `MASTER`.
8. Using Oracle Tuxedo 12*c* (12.1.1.0.0), reboot the machine that you shut down in step 7.

For more information about hot upgrades, refer to the following section, “Hot-Upgrade Example.”

Hot-Upgrade Example

The following example describes a sample scenario for a hot upgrade and the sequence of steps to perform the hot upgrade.

Suppose you have the following on a UNIX platform:

- A machine called `MACH1` that is configured as the `MASTER` and that runs an old release
  - A corresponding application directory called `$APPDIR1`
  - A corresponding configuration file called `$TUXCONFIG1`
- A machine called `MACH2` that is configured as the backup
  - A corresponding application directory called `$APPDIR2`
  - A corresponding configuration file called `$TUXCONFIG2`

Refer to the following instruction sequence in Listing 5-1 to perform a hot upgrade for your application.

Listing 5-1 Typical Hot Upgrade

```
MACH1> tmshutdown -B MACH2 -l MACH2
MACH1> Run tmconfig to reset the APPDIR, TUXCONFIG, TUXDIR for MACH2
       MACH2> Kill tlisten process
       MACH2> Load Tuxedo 12c if it is not already loaded
       MACH2> cd $APPDIR2
       MACH2> rm $TUXCONFIG2
       MACH2> >UPGRADE    # Indicator that upgrade being done
       MACH2> Reset PATH, LD_LIBRARY_PATH, TUXCONFIG, APPDIR
       MACH2> Rebuild clients and servers, create TLOG
       MACH2> Start Tuxedo 12c tlisten
MACH1> >UPGRADE    # Indicator that upgrade being done
MACH1> tmboot -B MACH2 -l MACH2
       MACH2> tmadmin
              master
              y
              psr
              q
       MACH2> tmshutdown -B MACH1 -l MACH1
       MACH2> Run tmconfig to reset the APPDIR, TUXCONFIG, TUXDIR for MACH1
MACH1> Kill tlisten process
MACH1> Load Tuxedo 12c if it is not already loaded
MACH1> cd $APPDIR1
MACH1> rm $TUXCONFIG1
MACH1> Reset PATH, LD_LIBRARY_PATH, TUXCONFIG, APPDIR
MACH1> Rebuild clients and servers, create TLOG
MACH1> Start Tuxedo 12c tlisten
MACH1> rm UPGRADE  # Remove indicator that upgrade being done
```

```
MACH2> rm UPGRADE  # Remove indicator that upgrade being done
MACH2> tmboot -B MACH1 -l MACH1
```

Note: If you are performing the upgrade illustrated in Listing 5-1 from a Windows platform, be sure to stop the Oracle Tuxedo `procMGR` (called IPC Helper in Oracle Tuxedo 6.5 or Oracle WebLogic Enterprise 5.1) service after stopping the `tlisten` process and to start the Oracle Tuxedo `procMGR` service *before* starting the `tlisten` process.

Rebuilding an Application

Now that you have successfully installed the Oracle Tuxedo software, you need to rebuild all application clients and servers that you want to execute on a system upgraded to Oracle Tuxedo 12*c* (12.1.1.0.0)

If your Oracle Tuxedo system applications are distributed, the `MASTER` and backup `MASTER` machines must run the highest release of the Oracle Tuxedo software of any machines in the configuration. Other machines, including Workstation clients, may continue to use executables from an earlier release. For example, even if your `MASTER` and backup `MASTER` machines are running Oracle Tuxedo 12*c* (12.1.1.0.0), Oracle Tuxedo 6.5 or later may still be used on other machines.

There is one important exception to the “highest release” rule. If you wish to install Oracle Tuxedo 12*c* (12.1.1.0.0) only on Workstation clients, and you do not want to upgrade any of your Oracle Tuxedo system server machines to Oracle Tuxedo 12*c* (12.1.1.0.0), you may do so. In other words, with any release level beginning with Oracle Tuxedo 6.5, Workstation clients may be intermixed freely in a configuration in which the `MASTER` machines is running Oracle Tuxedo 6.5 or later.

Installing Oracle Jolt 12*c* (12.1.1.0.0) with Oracle WebLogic Server

This section provides supplementary installation instructions for customers using Oracle Jolt 12*c* (12.1.1.0.0) with Oracle WebLogic Server 6.1, 7.0, or 8.1 and Oracle Tuxedo 6.5, 7.1, 8.0, 8.1, 9.0, 9.1, or 10.0. You can install the Jolt client personality software known as “Oracle Jolt for Oracle WebLogic Server” on an Oracle WebLogic Server 6.1, 7.0, or 8.1 machine by copying the Jolt client personality software from a Oracle Tuxedo 12*c* (12.1.1.0.0) machine to the target Oracle WebLogic Server machine. Of course, if the Oracle Tuxedo 12*c* (12.1.1.0.0) machine and the target Oracle WebLogic Server machine are the same machine, you need only include the Jolt client library class files in your `CLASSPATH` variable setting.

The “Oracle Jolt for Oracle WebLogic Server” client software consists of three JAR files: `jolt.jar`, `joltjse.jar`, and `joltwls.jar`. Include these files in your `CLASSPATH` variable setting.

For instructions on configuring the Jolt connection pool on Oracle WebLogic Server, see *Using Oracle Jolt with Oracle WebLogic Server*.

CHAPTER 6

Performing Post-Installation Tasks

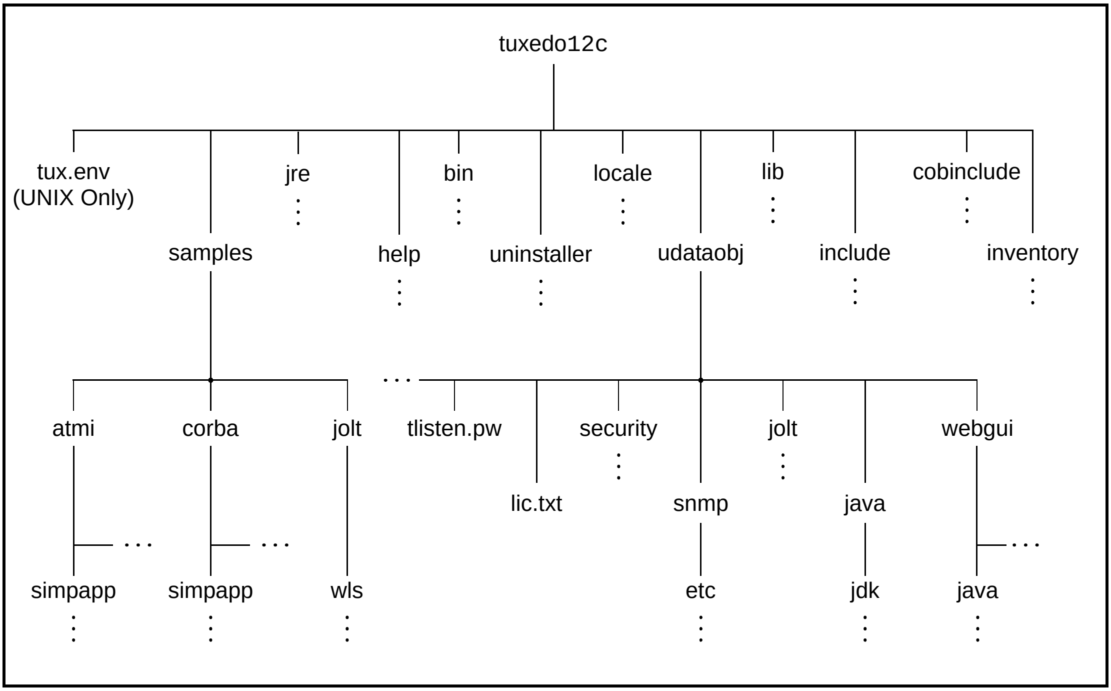
The following sections describe the tasks you perform after installing Oracle Tuxedo:

Understanding the Oracle Tuxedo Directory Structure

During the Oracle Tuxedo software installation, the installer program creates the following directory structure for a *full installation*. A full installation contains all the Oracle Tuxedo server and client software components plus the Link-Level Encryption (LLE) and Secure Sockets Layer (SSL) encryption software packages.

Oracle Tuxedo 12*c* (12.1.1.0.0) Directory Structure is shown in Figure 6-1.

Figure 6-1 Oracle Tuxedo 12*c* (12.1.1.0.0) Directory Structure



The product directory shown here, `tuxedo12`*c* (12.1.1.0.0), is the default for Oracle Tuxedo 12*c* (12.1.1.0.0). The default name can be changed during installation.

The top-level directories and files of the Oracle Tuxedo directory structure are briefly described in Table 6-1.

Table 6-1 Top-level Directories and Files of the Oracle Tuxedo Directory Structure

Directory Name	Description
`samples` **Note:** Samples are not installed unless they are selected during Tuxedo 12c installation.	Contains sample code and resources designed to help you learn how to develop your own applications using Oracle Tuxedo. The samples directory contains the following subdirectories: • `atmi` A collection of simple applications that demonstrate the many features of the Oracle Tuxedo Application-to-Transaction Monitor Interface (ATMI) server software. • `corba` A collection of simple applications that demonstrate the many features of the Oracle Tuxedo Common Object Request Broker Architecture (CORBA) C++ server software. • `jolt` A collection of simple applications that demonstrate the many features of the Oracle Jolt server software.
`help`	Contains online help files for the Oracle Tuxedo Administration Console.
`jre` directory	A directory containing the 1.5x version of the Java Runtime Environment (JRE). JRE 1.5.x provides the Java Virtual Machine, or JVM, required by the Oracle Tuxedo installation program. This version of the JRE is included in the Oracle Tuxedo 12*c* (12.1.1.0.0) distribution. It is automatically installed in the Oracle Home directory when you install Oracle Tuxedo. **Note:** The JRE cannot be used for development. To develop Oracle Tuxedo Java client applications, you must ensure that the Java Development Kit (JDK) is installed on your system.

Table 6-1 Top-level Directories and Files of the Oracle Tuxedo Directory Structure

Directory Name	Description
`bin`	Contains Tuxedo, Jolt, and SNMP Agent executable programs.
`uninstaller`	Contains code required to uninstall the Oracle Tuxedo software.
`locale`	Contains subdirectories to support the localization of system messages. `C` subdirectory contains message catalogs for the default locale (U.S. English).
`cobinclude`	Contains `.cbl` file entries for use in COBOL programs.

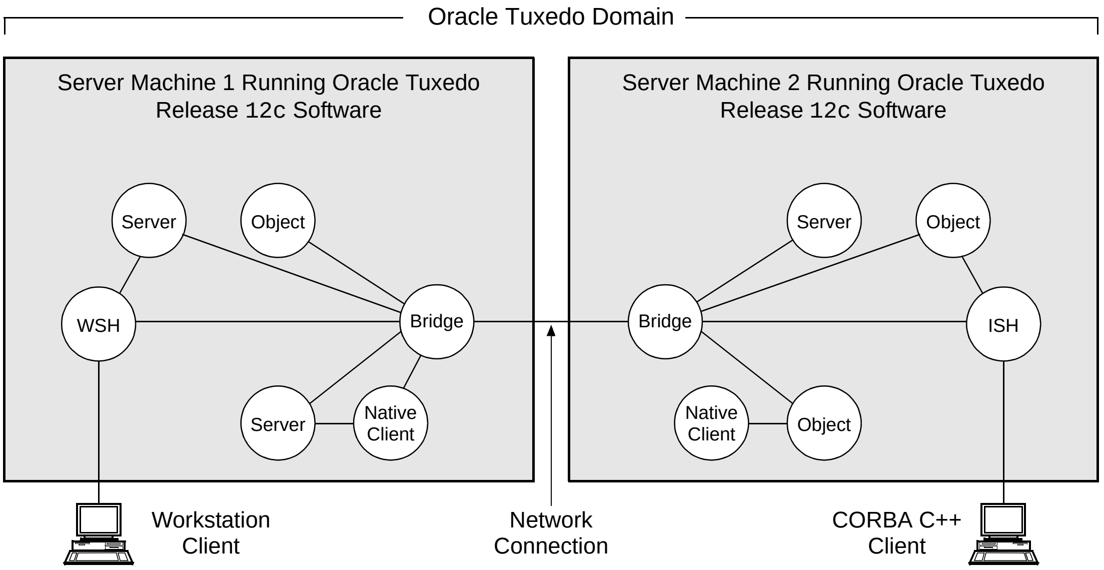
Table 6-1 Top-level Directories and Files of the Oracle Tuxedo Directory Structure

Directory Name	Description
`lib`	Contains compiled object files, including dynamic shared libraries (for platforms on which Oracle Tuxedo uses dynamic shared libraries) and other object files needed to build Oracle Tuxedo clients and servers.
`include`	Contains C and C++ language header files, as well as OMG IDL files. Includes subdirectories `unicode` and `xercesc`; may include subdirectory `rpc`, depending on the platform.
`udataobj`	Contains other directories and files required by Oracle Tuxedo. The udataobj directory contains the following subdirectories and files: • `security` Contains the default Lightweight Directory Access Protocol (LDAP) filter file (`bea_ldap_filter.dat`) and LLE-related and SSL-related encryption files. • `jolt` Contains the files for the Oracle Jolt software components that you selected to install. • `snmp` Contains the `etc` directory, which in turn contains the SNMP Agent configuration files, MIB files, and other miscellaneous files. • `java` Contains the classes and Java archive files needed to run Java applications. • `webgui` Contains the Java and image files for the Oracle Tuxedo Administration Console. • `tlisten.pw` (file) Contains the `tlisten` administrative password that you entered during the installation.
tux.env (file)	UNIX systems only: Contains Oracle Tuxedo environment variables for UNIX installations and serves as a model for setting those variables.

Understanding the Oracle Tuxedo Architecture

Figure 6-2 shows an Oracle Tuxedo *domain*, which is the basis of the Oracle Tuxedo architecture.

Figure 6-2 Simplified View of Oracle Tuxedo Architecture



A Tuxedo domain, also known as a Tuxedo application, is a business software program, built upon the Tuxedo system, that is defined and controlled by a single configuration file—the `UBBCONFIG` file. A 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.

Other important terms and concepts that you need to know about Oracle Tuxedo before performing post-installation checks are briefly described in the following sections:

- UBBCONFIG File
- MASTER Machine
- TUXCONFIG File
- TUXCONFIG Environment Variable

- TUXDIR Environment Variable

There is no need to fully understand these terms now; rather, use the sections as a reference. As you come upon these terms during the post-installation procedures, refer back to these sections to understand exactly what these terms mean.

UBBCONFIG File

Each Tuxedo domain is controlled by a configuration file in which installation-dependent parameters are defined. The text version of the configuration file is referred to as UBBCONFIG, although the configuration file may have any name, as long as the content of the file conforms to the format described in reference page UBBCONFIG(5) in *OracleTuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*.

The UBBCONFIG file for a Tuxedo domain contains all the information necessary to boot the application, such as lists of its resources, machines, groups, servers, available services, and so on. It consists of nine sections, five of which are required for all configurations: RESOURCES, MACHINES, GROUPS, SERVERS, and SERVICES.

MASTER Machine

The MASTER machine, or MASTER node, for a Tuxedo domain contains the domain's UBBCONFIG file, and is designated as the MASTER machine in the RESOURCES section of the UBBCONFIG file. Starting, stopping, and administering a Tuxedo domain is done through the MASTER machine.

In a multi-machine Tuxedo domain running different releases of the Tuxedo system software, the MASTER machine must run the highest release of the Tuxedo system software in the domain.

TUXCONFIG File

The TUXCONFIG file is a binary version of the UBBCONFIG file. It is created by running the tmloadcf(1) command, which parses UBBCONFIG and loads the binary TUXCONFIG file to the location referenced by the TUXCONFIG environment variable. As with UBBCONFIG, the TUXCONFIG file may be given any name.

The MASTER machine for a Tuxedo domain contains the master copy of the TUXCONFIG file. Copies of the TUXCONFIG file are propagated to all other server machines—referred to as non-MASTER machines—in a Tuxedo domain whenever the Tuxedo system is booted on the MASTER machine.

TUXCONFIG Environment Variable

The TUXCONFIG environment variable defines the location on the MASTER machine where the tmloadcf(1) command loads the binary TUXCONFIG file. It must be set to an absolute pathname ending with the device or system file where TUXCONFIG is to be loaded.

The TUXCONFIG pathname value is designated in the MACHINES section of the UBBCONFIG file. It is specified for the MASTER machine *and* for every other server machine in the Tuxedo domain. When copies of the binary TUXCONFIG file are propagated to non-MASTER machines during system boot, the copies are stored on the non-MASTER machines according to the TUXCONFIG pathname values.

TUXDIR Environment Variable

The TUXDIR environment variable defines the product installation directory of the Oracle Tuxedo software on the MASTER machine. It must be set to an absolute pathname ending with the name of the product installation directory.

The TUXDIR pathname value is designated in the MACHINES section of the UBBCONFIG file. It is specified for the MASTER machine *and* for every other server machine in the Tuxedo domain.

Setting Up Your Environment

You need to set several environment variables before using Oracle Tuxedo to build and run Oracle Tuxedo applications. Table 6-2, Table 6-3, Table 6-4, Table 6-5 list and define many of those environment variables.

Table 6-2 Oracle Tuxedo Core Environment Variables (Sheet 1 of 2)

Environment Variable	Description
TUXDIR	Absolute pathname of the product directory in which you installed the Oracle Tuxedo software on this machine. TUXDIR must be set on both server and client-only machines.
APPDIR	Absolute pathname of the application directory in which application and administrative servers will be booted on this server machine. APPDIR may be set to more than one application directory.

Table 6-2 Oracle Tuxedo Core Environment Variables (Sheet 2 of 2)

Environment Variable	Description
TUXCONFIG	Absolute pathname of the device or system file where the binary TUXCONFIG file is found on this server machine. The TUXCONFIG file is created by running the tmloadcf(1) command on the UBBCONFIG configuration file.
WEBJAVADIR	Absolute pathname of the Java and image files for the Oracle Tuxedo Administration Console on this server machine.

Table 6-3 Oracle Tuxedo Client-Only Environment Variables

Environment Variable	Description
WSENVFILE	Tuxedo ATMI Workstation (/WS) client: Name of the file in which all environment variables are set for this workstation. There is no default for this variable.
TOBJADDR	Tuxedo remote CORBA client: Address of the Tuxedo server machine’s listener; must match exactly (including case) the host and port specified for the server machine in the UBBCONFIG file.

Table 6-4 COBOL Environment Variables

Environment Variable	Description
COBCPY	Directories that contain a set of the COBOL COPY files to be used by the compiler.
COBOPT	Arguments that you may want to use on the compile command line.

Table 6-5 Java Environment Variables

Environment Variable	Description
`JAVA_HOME`	Absolute pathname of the Java Development Kit (JDK) 1.5.x installation directory on this machine; needed to build and run Java applications on this machine.*
`JDKDIR`	Set to `JAVA_HOME` value.
`CLASSPATH`	Absolute pathnames to classes and Java archive files on this machine; needed to run Java applications on this machine.

* The Oracle Tuxedo 12*c* (12.1.1.0.0) distribution does not include a JDK.

Setting Environment Variables on a UNIX System

On a UNIX server machine, set and export the following environment variables shown in Table 6-6 to set up your environment:

Table 6-6 Environment Variables

`TUXDIR=`*`pathname_of_Oracle_Tuxedo_product_directory`*
`APPDIR=`*`pathname_of_Oracle_Tuxedo_application_directory`*
`TUXCONFIG=`*`pathname_of_TUXCONFIG_file`*
`WEBJAVADIR=$TUXDIR/udataobj/webgui/java`
`PATH=$APPDIR:$TUXDIR/bin:/bin:$PATH`
`LD_LIBRARY_PATH=$APPDIR:$TUXDIR/lib:/lib:/usr/lib:$LD_LIBRARY_PATH` **Note:** For HP-UX systems only, use `SHLIB_PATH` instead of `LD_LIBRARY_PATH`. For AIX systems only, `LIBPATH` instead of `LD_LIBRARY_PATH`.
`export TUXDIR APPDIR TUXCONFIG WEBJAVADIR PATH LD_LIBRARY_PATH`

Examples of `TUXDIR`, `APPDIR`, and `TUXCONFIG` are:

```
TUXDIR=/home/Oracle/tuxedo12c
```

```
APPDIR=/home/me/simpapp
TUXCONFIG=$APPDIR/tuxconfig
```

The TUXDIR, APPDIR, and TUXCONFIG environment variables must match the values of the TUXDIR, APPDIR, and TUXCONFIG parameters in the MACHINES section of the UBBCONFIG file. A Bourne shell script named tux.env, located in the Oracle Tuxedo product directory, serves as a model for setting these and other environment variables on a UNIX system.

Starting the tlisten Process

You, as the application administrator, must start a tlisten process on each machine of a networked Oracle Tuxedo application before the application is booted. The tlisten process enables you and the Oracle Tuxedo software running on the MASTER machine to start, shut down, and administer Oracle Tuxedo processes running on the non-MASTER machines. For example, tmboot(1) can start Oracle Tuxedo system servers on the non-MASTER machines. Generally, one tlisten process is required for each Oracle Tuxedo application running on a server machine.

In addition to the installer program starting a tlisten process on port 3050 during the installation of Oracle Tuxedo, a tlisten process may be started as shown in tlisten Process Table 6-7.

Table 6-7 tlisten Process

On This Machine . . .	By This Administrator . . .	Using This Method . . .
UNIX server	UNIX system administrator	As part of a UNIX initialization (boot) script
	Oracle Tuxedo application administrator	As a cron job
		Manually starting a tlisten process from a command-line shell

tlisten Invocation

In all cases, the same basic syntax is used to invoke `tlisten`.

`$TUXDIR/bin/tlisten` [`-d` *`devname`*] `-l` *`nlsaddr`* [`-u` *`appuid`*] (UNIX)

The `-l` option is required. The argument to `-l` must match the value of the `NLSADDR` parameter in the `NETWORK` section of the `UBBCONFIG` file. For information on determining the value of `NLSADDR`, see reference page `UBBCONFIG(5)` in *OracleTuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*.

The value of *`devname`* is the device name of the network provider; for example, Starlan. If the `tlisten` process is operating with Sockets, the `-d` option is not needed.

The value of *`appuid`* is the user identifier (UID), or login name, of the Oracle Tuxedo application administrator. It must match the value of the `UID` parameter in the `RESOURCES` section of the `UBBCONFIG` file.

Note: To obtain the UID on a UNIX system, run the `id` command.

On a UNIX machine, use the `-u` *`appuid`* option when the command is part of an installation script run by user `root` to run the `tlisten` process with the effective UID of the owner of the Oracle Tuxedo software installation on this machine. If `tlisten` is started by the Oracle Tuxedo application administrator, either as a `cron` job or manually, the `-u` option is unnecessary since the job is already owned by the correct account.

For more information about the `tlisten` command, see reference page `tlisten(1)` in *Oracle Tuxedo Command Reference*. For details on starting the `tlisten` process on a Windows 2003 server machine, see "Configuring tlisten Processes to Start Automatically" in *Using Oracle Tuxedo ATMI on Windows*.

tlisten Password

Oracle Tuxedo uses the administrative password that you specified during the installation to protect the machine on which Oracle Tuxedo is installed from administrative requests and operations (such as `tmboot(1)`) that are not authorized. Whenever administrative communications arrive on this machine through `tlisten(1)` or `wlisten(1)` gateway processes, Oracle Tuxedo authenticates them by means of the password.

A `tlisten` password must be a string of alphanumeric characters in clear-text format. It may contain no more than 80 characters.

A common password is required for two machines in a Oracle Tuxedo application to communicate successfully. For this reason, you must use the same password whenever you install

Oracle Tuxedo on multiple machines for a single application. If during the Oracle Tuxedo installation process you use a different password for one machine, you must add that password using the tlistpwd(1)$TUXDIR utility to the tlisten.pw file on each machine that communicates with eachother.

For these reasons, you may have more than one administrative password in your tlisten.pw file. A single password file may contain no more than 20 passwords, with one password per line.

Note: The tlisten.pw file is system-encrypted. To add or change passwords, you must use the tlistpwd(1) utility.

Editing a UBBCONFIG File

Each Oracle Tuxedo application is controlled by a configuration file in which installation-dependent parameters are defined. In the Oracle Tuxedo documentation, this file is referred to as UBBCONFIG, but you can give your configuration file any name you like as long as the content of the file conforms to the format described in reference page UBBCONFIG(5) in *OracleTuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*. Typical configuration filenames begin with the string ubb, followed by a mnemonic string, such as simple in the filename ubbsimple.

As an example, consider ubbsimple, the UBBCONFIG file for the rudimentary ATMI-based simpapp application delivered with the Oracle Tuxedo installation. On a UNIX system, it is found in the directory $TUXDIR/samples/atmi/simpapp.

Listing 6-1 shows an example ubbsimple file. The example has been modified from the ubbsimple file delivered on a Windows or UNIX system to include example pathname values for both Windows and UNIX systems.

Listing 6-1 ubbsimple for the ATMI-based simpapp Application

```
#ident “@(#)apps:simpapp/ubbsimple   $Revision: 1.3 $

#Skeleton UBBCONFIG file for the Tuxedo Simple Application.
#Replace the <bracketed> items with the appropriate values.

*RESOURCES
IPCKEY          <Replace with a valid IPC Key>

#Example:
#IPCKEY         123456
```

```
 DOMAINID        simpapp
 MASTER          simple
 MAXACCESSERS    10
 MAXSERVERS      5
 MAXSERVICES     10
 MODEL           SHM
 LDBAL           N

 *MACHINES
 DEFAULT:
                 APPDIR=“<Replace with the current directory pathname>”
                 TUXCONFIG=“<Replace with your TUXCONFIG Pathname>”
                 TUXDIR=“<Directory where Tuxedo is installed>”
 #UNIX
 #Example:
 #               APPDIR=“/home/me/simpapp”
 #               TUXCONFIG=“/home/me/simpapp/tuxconfig”
 #               TUXDIR=“/home/oracle/tuxedo12c”


 <Machine-name>  LMID=simple

 #Example:
 #beatux         LMID=simple

 *GROUPS
 GROUP1
           LMID=simple GRPNO=1 OPENINFO=NONE

 *SERVERS
 DEFAULT:
                 CLOPT=“-A”
simpserv        SRVGRP=GROUP1 SRVID=1

 *SERVICES
 TOUPPER
```

In the configuration file for your application, you must replace the strings enclosed in angle brackets with values specific to your application. Listing 6-8 provides a sample of the parameters that must be defined in every configuration file.

Table 6-8 Parameter Sample

This Parameter . . .	Specifies . . .
`IPCKEY`	A numeric key that identifies the shared memory segment where the structures used by your application are located. The value must be greater than 32,768 and less than 262,143.
`machine_name`	The node name of the machine. To obtain the node name on a UNIX system, run the `uname -n` command.
`APPDIR` = *`string`*	A list of one or more directories in which application and administrative servers will be booted on this machine. For UNIX, the value of *`string`* is the absolute pathname of one directory, optionally followed by a colon-separated list of pathnames for other directories on the machine being defined.
`TUXCONFIG` = *`string`*	The absolute pathname of the device or system file where the binary `TUXCONFIG` file is to be created on this machine. The `TUXCONFIG` file is created by running the `tmloadcf(1)` command on the `UBBCONFIG` file.
`TUXDIR` = *`string`*	The absolute pathname of the product directory of the Oracle Tuxedo software on this machine.

You must define `APPDIR`, `TUXCONFIG`, and `TUXDIR` for every machine in your Oracle Tuxedo application. If you need to look up other parameters when editing your `UBBCONFIG` file, see reference page `UBBCONFIG(5)` in *OracleTuxedo File Formats, Data Descriptions, MIBs, and System Processes Reference*.

You must edit your `UBBCONFIG` file before running `tmloadcf(1)` to verify the IPC requirements in the section that follows. If you run `tmloadcf` without first editing the `UBBCONFIG` file, the command will fail with syntax errors.

Using the TYPE Parameter in UBBCONFIG

The `TYPE` parameter in the `MACHINES` section of a configuration file specifies the invocation of the External Data Representation (XDR) encode/decode routines when messages are passed between unlike machines. The term “unlike” applies even to machines of the same type if the compiler on each machine is different. In such a case, give each machine a unique `TYPE` string to force every message to go through the encode/decode routines.

Checking IPC Requirements

The Oracle Tuxedo system uses interprocess communications (IPC) resources heavily. On many systems, the default values for the parameters that control the size and quantity of the various IPC resources are below the minimums needed to run even a modest Oracle Tuxedo application. Therefore, you may need to reset some parameters. After editing your `UBBCONFIG` file, you should determine whether you have enough IPC resources for your application.

To perform this task, enter the following `tmloadcf(1)` command, specifying your edited `UBBCONFIG` file as input:

```
tmloadcf -c UBBCONFIG
```

With the `-c` option, the `tmloadcf` program prints a list of the minimum IPC resources required for your application, but does not create or update the `TUXCONFIG` file.

Listing 6-2 is an output report based on the values in `ubbsimple`.

Listing 6-2 Output Produced by tmloadcf -c

```
Ipc sizing (minimum /T values only)...
                   Fixed Minimums Per Processor
SHMMIN: 1
SHMALL: 1
SEMMAP: SEMMNI
                   Variable Minimums Per Processor
        SEMUME,           A                             SHMMAX
        SEMMNU,           *                               *
Node    SEMMNS  SEMMSL  SEMMSL  SEMMNI  MSGMNI  MSGMAP  SHMSEG
------  ------  ------  ------  ------  ------  ------  ------
sftuxe     17        5      12   A + 1      13      26     75K

where 1 <= A <= 8.

The number of expected application clients per processor should be added to each
MSGMNI value.
```

The output report identifies IPC resources by their traditional UNIX names. To map the traditional names to the names specific to a UNIX platform, see the data sheet for that platform in "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1.

The example output report indicates that to run `simpapp`, your system must have `SEMUME`, `SEMMNU`, and `SEMMNS` set to no less than 17. `SEMMSL` must be at least 5, and `SEMMNI` and `SEMMAP` at least 4 (assuming the value of A is 3). `MSGMNI` must be at least 13, and `MSGMAP` at least 26. Finally, the product of `SHMMAX` and `SHMSEG` must be at least 75K bytes.

The IPC values are application-dependent, and the numbers in this example reflect a very small configuration. If other client or server applications that use IPC resources are running on the same system with a Oracle Tuxedo application, then the requirements of both applications must be satisfied. Keep in mind also that every machine participating in an application must have sufficient IPC resources available.

If the current IPC resources are inadequate, you must increase the values of the associated IPC parameters.For instructions on changing the current IPC values for a UNIX system, see the data sheet for your platform in "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1.

Creating the Universal Device List and the Transaction Log

You create the Universal Device List (UDL) and define a UDL entry for the global transaction log (TLOG) on each machine in your application that will use global transactions. The TLOG is a log file in which information about transactions is kept until the transaction is completed.

Defining the TLOG

Before creating the UDL and defining UDL entries for TLOG, you must set the following parameters in the `MACHINES` section of the `UBBCONFIG` file for each machine in your application that will use global transactions.

Table 6-9 lists the parameters.

Table 6-9 Parameter

This Parameter . . .	Specifies . . .
`TLOGDEVICE` = *`string`*	The Oracle Tuxedo filesystem containing the distributed transaction processing (DTP) TLOG for this machine. If not specified, it is assumed that this machine has no TLOG.
`TLOGOFFSET` = *`offset`*	The numeric offset in pages (from the beginning of the device) to the start of the Oracle Tuxedo filesystem containing the DTP transaction log for this machine. The default is 0.
`TLOGNAME` = *`string`*	The name of the DTP transaction log for this machine. If not specified, the default is "TLOG."
`TLOGSIZE` = *`size`*	The numeric size, in pages, of the DTP transaction log for this machine. If not specified, the default is 100 pages.

Because the TLOG seldom needs to be larger than 100 blocks (pages) and because disk partitions are always substantially larger than that, it may make sense to use the same device for both the `TUXCONFIG` file and the TLOG. If so, the pathname of the device needs to be specified by both the `TUXCONFIG` and the `FSCONFIG` environment variables.

Creating the UDL and UDL Entries for TLOG

You must manually create a UDL entry for the `TLOGDEVICE` on each machine where a TLOG is needed. You may create these entries either before or after you have loaded `TUXCONFIG`, but you must create these entries before booting the application.

To access the create device list command, `crdl`, you invoke `tmadmin -c` with the application inactive. The `-c` option invokes `tmadmin` in configuration mode.

To create the UDL and a UDL entry for TLOG on each machine in your application that will use global transactions, follow these steps:

1. Log in as the application administrator on the MASTER machine.
2. Enter the following command:

   ```
   tmadmin -c
   crdl -z config -b blocks
   ```

   Here `-z config` specifies the full pathname of the device on which the UDL should be created (that is, where the TLOG will reside), and `-b blocks` specifies the number of blocks to be allocated on the device. The value of *config* should match the value of the TLOGDEVICE parameter in the MACHINES section of the UBBCONFIG file. The blocks must be larger than the value of TLOGSIZE. If -z is not specified, the value of *config* defaults to the value of the FSCONFIG environment variable.
3. Log in as the application administrator on each remaining non-MASTER machine that will use global transactions and repeat step 2.

If the TLOGDEVICE is mirrored between two machines, step 3 is not required on the paired machine. To be recoverable, the TLOG should reside on a device that can be mirrored.

Running simpapp to Verify Your Installation

One of the ways to verify that your Oracle Tuxedo software is installed correctly is to run one or more of the sample applications included with the installation. The sample applications demonstrate the capabilities of the ATMI and CORBA clients, and the ATMI and CORBA C++ servers.

The following sections provide procedures for verifying both the ATMI and CORBA C++ parts of your Oracle Tuxedo installation:

- Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation
- Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation

The simpapp application is a non-distributed application, meaning that it runs on a single machine. It is designed so that it can be up and running within minutes after the Oracle Tuxedo software is installed.

The simpapp application offers a single service called TOUPPER, which converts strings from lowercase to uppercase. The client is invoked with a single argument: a lowercase string to be converted to uppercase. The server returns the converted string to the client, and the client prints the converted string.

For example, the invocation

```
simpcl “hello world”
```

results in the output

```
Returned string is: HELLO WORLD
```

Two versions of simpapp exist: an ATMI version and a CORBA version. The ATMI version consists of an ATMI server, an ATMI client, and a UBBCONFIG file. The CORBA version consists of a CORBA C++ server, a CORBA C++ client.

Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation

To verify that you have successfully installed the Oracle Tuxedo ATMI software on your system, run the ATMI version of the simpapp application, which on a UNIX system can be found in the directory $TUXDIR/samples/atmi/simpapp.

Note: Samples are not installed unless they are selected to be installed during Tuxedo 12c installation.

The procedure presented in the following two sections is also provided in the README file in the simpapp directory, and in “Tutorial for simpapp, a Simple C Application” in *Tutorials for Developing Oracle Tuxedo ATMI Applications*.

Running simpapp to Verify the Oracle Tuxedo ATMI Software Installation on a UNIX System

To configure and run the ATMI version of simpapp on a UNIX system, follow these steps:

1. Log in to the target machine as the Oracle Tuxedo application administrator and open a command-line shell.

2. Create a working directory for your sample application and change to it:

   ```
   cd /home/me
   mkdir atmi
   cd atmi
   ```

3. Set and export the environment variables used by the Oracle Tuxedo system, as explained in “Setting Environment Variables on a UNIX System” on page 6-10. Set APPDIR and TUXCONFIG as follows:

   ```
   APPDIR=/home/me/atmi
   TUXCONFIG=$APPDIR/tuxconfig
   export APPDIR TUXCONFIG
   ```

Note: You do not have to set the WEBJAVADIR environment variable.

4. Copy the simpapp files to your working directory. You will need to edit one of them: the configuration file, ubbsimple. Make sure that the client and server files, simpcl and simpserv, are executable, and that the configuration file, ubbsimple, is writable. For example:

   ```
   cp $TUXDIR/samples/atmi/simpapp/* .
   chmod 755 simpserv simpcl
   chmod 644 ubbsimple
   ```

5. Compile the simpapp client and server programs by entering the following commands:

   ```
   buildclient -o simpcl -f simpcl.c
   buildserver -o simpserv -f simpserv.c -s TOUPPER
   ```

6. In the sample configuration file, ubbsimple, replace the strings shown in angle brackets with values appropriate to your Oracle Tuxedo system installation. Comments in ubbsimple explain how to customize the file. Set the following parameters in the ubbsimple file:

   – Set IPCKEY to a valid IPC key. This value must be greater than 32,768 and less than 262,143.

   – Set APPDIR to ″/home/me/atmi″.

   – Set TUXCONFIG to the literal pathname corresponding to $APPDIR/tuxconfig (in our example, ″/home/me/atmi/tuxconfig″).

   – Set TUXDIR to the absolute pathname of the product directory of the Oracle Tuxedo software on this machine (for example, ″/home/oracle/tuxedo12c″).

   – Set MACHINE-NAME to the name of your system.

     To determine the name of your system on a UNIX machine enter the command:

     ```
     uname -n
     ```

   Note: The APPDIR, TUXCONFIG, and TUXDIR parameter settings in the ubbsimple file must match the APPDIR, TUXCONFIG, and TUXDIR environment variable settings.

7. Create the binary version of your edited configuration file by invoking tmloadcf(1), which produces a file named tuxconfig. This file, referenced by the TUXCONFIG environment variable, provides the Oracle Tuxedo system with a description of the application configuration at run time:

   ```
   tmloadcf -y ubbsimple
   ```

8. Boot simpapp by typing the following command:

   ```
   tmboot -y
   ```

If the boot succeeds, output similar to the following appears and you can proceed to step 10.

Listing 6-3 shows the output produced by tmboot -y.

Listing 6-3 Output Produced by tmboot -y

```
Booting all admin and server processes in /home/me/atmi/tuxconfig
INFO: Oracle Tuxedo(r) System Release 10.3.0.0
INFO: Serial #: 000102-9125503751, Maxusers 25
Booting admin processes ...
exec BBL -A:
     process id=24180 ... Started.
Booting server processes ...
exec simpserv -A :
     process id=24181 ... Started.
2 processes started.
```

9. If the boot fails, examine the log named `ULOG.`*`mmddyy`* in your application directory (`$APPDIR`, `/home/me/atmi`). The string *`mmddyy`* is a placeholder for the date (digits representing the current month, day, and year) that will make up the end of the filename. If you see a message such as the following, near the end of the log

    ```
    can't create enough semaphores for BB
    ```

    then the interprocess communication (IPC) resources configured in your operating system are not adequate for running `simpapp`.

    To confirm this hypothesis, invoke the Oracle Tuxedo system command `tmloadcf(1)` and specify the name of your configuration file, as shown in the following example:

    ```
    tmloadcf -c $APPDIR/ubbsimple
    ```

    If the current value of any IPC parameter configured in your operating system is less than a minimum (either variable or fixed) listed in the `tmloadcf` output, you must increase the value of that parameter. For instructions on determining and changing the current IPC values for your platform, see the data sheet for your platform in "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1.

10. If the boot succeeded, you can invoke the client. For example, enter the following command:

```
simpcl “hello world”
```

The following is displayed:

```
Returned string is: HELLO WORLD
```

11. When you have finished, shut down simpapp with the following command:

```
tmshutdown -y
```

Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation

To verify that you have successfully installed the Oracle Tuxedo CORBA C++ software on your system, run the CORBA version of the simpapp application, which on a UNIX system can be found in the directory $TUXDIR/samples/corba/simpapp. The procedure presented in the following two sections is also provided in the README file in the simpapp directory, and in “Tutorial for simpapp, a Simple C Application” in *Tutorials for Developing Oracle Tuxedo ATMI Applications*.

Running simpapp to Verify the Oracle Tuxedo CORBA C++ Software Installation on a UNIX System

To configure and run the CORBA version of simpapp on a UNIX system, follow these steps:

1. Log in to the target machine as the Oracle Tuxedo application administrator and open a command-line shell.

2. Create a working directory for your sample application and change to it:

```
prompt> cd /home/me
prompt> mkdir corba
prompt> cd corba
```

3. Make sure that the product directory in which you installed the Oracle Tuxedo software is set in the TUXDIR environment variable. For example, if you installed the software in the /home/oracle/tuxedo12c directory, set and export TUXCONFIG as follows:

```
prompt> TUXDIR=/home/oracle/tuxedo12c
prompt> export TUXDIR
```

4. Copy the simpapp files to your working directory and change the permissions on all files to allow full access. For example:

```
prompt> cp $TUXDIR/samples/corba/simpapp/* .
prompt> chmod 777 *
```

5. Ensure that `make` is in your path.

6. To run `simpapp` automatically, enter `../runme.ksh`. The `simpapp` application runs and prints the following messages:

   ```
   Testing simpapp
        cleaned up
        prepared
        built
        loaded ubb
        booted
        ran
        shutdown
        saved results
    PASSED
   ```

7. To run `simpapp` manually to observe the processes starting and stopping, follow these steps:

   a. prompt> `ksh`

   b. prompt> `../results/setenv.ksh`

   c. prompt> `tmboot -y`

      The application starts several processes.

   d. prompt> `simple_client`

      The prompt `String?` is displayed.

   e. prompt> *`enter_a_word_in_lowercase_letters`*

      The application converts the word to uppercase and then to lowercase letters and displays the results.

   f. prompt> `tmshutdown -y`

   g. The application shuts down the processes.

8. To restore the directory to its original state, follow these steps:

   a. prompt> `../results/setenv.ksh`

   b. prompt> `make -f makefile.mk clean`

Running buildtms for Oracle Tuxedo Applications That Use XA Resource Managers

For Oracle Tuxedo applications that use distributed transactions and XA-compliant resource managers, you must use the `buildtms` command to construct a transaction manager server load module. This requirement exists on UNIX systems. When the module has been created, it must reside in `$TUXDIR/bin` on UNIX systems.

If you run the CORBA C++ University sample applications, each sample's makefile creates the TMS load module for you and calls it `tms_ora.exe`. Therefore, running `buildtms` as a separate step is necessary only if you do not plan to run any of these sample applications.

For information about the `buildtms` command with Oracle Tuxedo applications, see reference page `buildtms(1)` in *Oracle Tuxedo Command Reference*.

Uninstalling Oracle Tuxedo

Uninstalling Oracle Tuxedo12*c* (12.1.1.0.0), 11.0, 10.0, 9.1, 9.0, 8.1 or 8.0 does not remove the Oracle Home directory associated with the installation but does remove all Oracle Tuxedo components installed by the installer program. The uninstallation also removes the product directory associated with the installation unless one of the following is true:

- The product directory contains customer-created configuration or application files—the uninstallation does not delete customer-created configuration or application files.
- The uninstall was invoked from within the product directory structure—specifically, within the `uninstaller` directory.

To uninstall Oracle Tuxedo 12*c* (12.1.1.0.0), complete the procedures for the appropriate platform, provided in Table 6-10. To uninstall Oracle Tuxedo 9.1 or 10.0, substitute "Tuxedo 9.1 or 10.0" for "Tuxedo 12*c* (12.1.1.0.0)" in Table 6-10.

Table 6-10 Uninstall Oracle Tuxedo

To Uninstall Oracle Tuxedo on This Platform . . .	Perform the Following Procedure . . .
UNIX	1. Shut down any Oracle Tuxedo servers that are running. For instructions on using the `tmshutdown` command to shut down Oracle Tuxedo applications, see reference page `tmshutdown(1)` in *Oracle Tuxedo Command Reference*. 2. Go to the `tux_prod_dir/uninstaller` directory, where `tux_prod_dir` represents the product directory in which you installed the Oracle Tuxedo software. 3. Choose one of two methods for uninstalling the software: – To use the GUI-mode installation program, go to step 4. – To use the console-mode procedure, go to step 5. 4. (GUI-mode method) Enter the `sh Uninstall_Tuxedo_12c` command at the prompt. In the Uninstaller window, click Uninstall to start the uninstall program, then click Exit in the Uninstall Complete window to complete the uninstallation. 5. (Console-mode method) Enter the `sh Uninstall_Tuxedo_12c -i console` command at the prompt. When the uninstall process is complete, press Enter to exit the uninstaller.

Reinstalling Oracle Tuxedo

When you start the Oracle Tuxedo Installation program on system that already has a copy of Oracle Tuxedo installed, the installation program detects the existing Oracle Tuxedo home directory and asks if you want to:

- Copy the new installation directory over the existing one
- Create a new installation directory

Table 6-11 shows the reinstallation screen.

Table 6-11 Reinstalling Oracle Tuxedo

Click . . .	To . . .
Continue	Close the warning window and continue with the installation. This option overwrites the previous installation.
Cancel	Return to the Choose Oracle Home Directory window. To continue installing the software using a different Oracle Home directory, select an existing Oracle Home directory that does not contain the release 12*c* (12.1.1.0.0) software or create a new Oracle Home directory.
Exit	Exit the installation program and uninstall the previous installation. You can invoke the uninstall program as described in "Uninstalling Oracle Tuxedo" on page 6-25, and reinstall the software as described in one of the following: • "Installing Oracle Tuxedo Message Queue Using GUI-Mode Installation" on page 2-1 • "Installing Oracle Tuxedo Message Queue on UNIX Systems Using" on page 3-1 • "Installing Oracle Tuxedo Message Queue Using Silent Installation" on page 4-1

APPENDIX A

Oracle Tuxedo Message Queue 12*c* (12.1.1.0.0) Platform Data Sheets

The following sections provide platform-specific information for the platforms on which the Oracle Tuxedo 12*c* (12.1.1.0.0) system software is supported:

- Supported Platforms
  - Platform Support Policy
- Supported Platform Data Sheets
  - IBM AIX 6.1 (64-bit) on IBM PowerPC
  - Oracle Linux 5.6 (64-bit) on x86_64
  - Oracle Solaris 10 (64-bit) on SPARC
- Platforms Supporting Threads

Supported Platforms

Vendor	Operating System	Release/Version
IBM	AIX	IBM AIX 6.1 (64-bit) on IBM PowerPC
Oracle	Linux	Oracle Linux 5.6 (64-bit) on x86_64 Oracle Solaris 10 (64-bit) on SPARC

Note: Oracle Jolt 12*c* (12.1.1.0.0) and Oracle SNMP Agent 12*c* (12.1.1.0.0) are supported on the same platforms that are supported by Oracle Tuxedo 12*c* (12.1.1.0.0).

A data sheet is provided for each platform. Each data sheet includes the following platform-specific information:

- A list of available Oracle Tuxedo install sets
- Hardware, software, network, and disk space requirements
- Instructions for mounting and unmounting the Oracle Tuxedo software DVD
- Tunable parameters

Platform Support Policy

The Oracle policy regarding discontinuance of operating system platform support for Oracle Tuxedo and related products is as follows:

- x86 or x86-64 denotes various CPUs based on x86 or x86-64 architecture.
- Oracle Tuxedo 32-bit binaries are supported on 64-bit operating systems, such as on 64-bit versions of HP-UX, Solaris, AIX and Windows. This support is based on OS vendor compatibility assurance that 32-bit applications can run on 64-bit OS without any changes. For more 32-bit application support on 64-bit OS information, please refer to respective OS vendor documentation.
- Oracle Tuxedo is considered supported in virtualization software environments provided by the operating system vendor (such as Solaris Containers, HP-UX VM and AIX6 WPARs). Even though Oracle has performed limited testing in these environments, the support is based on the OS vendor compatibility statement that software applications will run in a virtualization software environment without any changes.

  Other virtualization software environments (such as VMWare), are also considered supported, provided that the virtualization software environment is supported by the operating system vendor and the support criteria specified in the operating system data sheets are met.

  For more information, see Supported Platform Data Sheets.
- The Oracle policy is that when the operating system provider discontinues supporting a specific release, Oracle support is also discontinued.

- Oracle will not be able to notify customers and users when support for a specific operating system release is discontinued. As a user of the operating system release, we expect you to find out from the operating system provider of discontinued support and plan accordingly.

Platform End of Life Expectancy Table

The following is the End-of Life expectancy table for Oracle Tuxedo 12*c* (12.1.1.0.0) supported platforms.

Note: OS EOL dates may have changed since time of this publication. Check with vendor for current OS EOL dates.

Platforms	Release Date	OS EOL Date
IBM AIX 6.1 (64-bit) on IBM PowerPC	2010.08.31	TBD
Oracle Linux 5.6 (64-bit) on x86_64	2011.02.21	TBD
Oracle Solaris 10 (64-bit) on SPARC	2010.08.31	TBD

EOL information for all Oracle products can be found at:

http://www.oracle.com/support/library/brochure/lifetime-support-middleware.pdf.

Tunable Parameters

You probably need to reconfigure the platform kernel before running Oracle Tuxedo software because the default values of some tunable parameters are too low.

To adjust the tunable parameters, follow these steps:

1. Determine whether the current values are adequate.

   For instructions about determining whether the current tunable parameter values are adequate, see "IPC Resource Configuration on a UNIX System" on page E-1 and "Checking IPC Requirements" on page 6-18.

2. Reset the tunable parameters as necessary.

Supported Platform Data Sheets

IBM AIX 6.1 (64-bit) on IBM PowerPC

The following sections list Oracle Tuxedo 12*c* (12.1.1.0.0) requirements for AIX 6.1 (64-bit) systems on IBM PowerPC.

Installation

- Available Oracle Tuxedo 12*c* (12.1.1.0.0) Install Sets for IBM AIX 6.1 (64-bit)
  - Full install set
  - Server install set
  - Full client install set
  - ATMI (/WS) client install set
  - CORBA client install set
  - Jolt client install set

  For descriptions of Oracle Tuxedo install sets, see "Install Sets" on page 1-9.

- OTMQ install for IBM AIX 6.1 (64-bit)

Hardware Requirements for IBM AIX 6.1 (64-bit)

- IBM PowerPC
- 1 GB of RAM minimum
- At least 5MB RAM for each Oracle Tuxedo system server

Software Requirements for IBM AIX 6.1 (64-bit)

Component	Requirement
OS Version (Patches)	IBM AIX 6.1 64-bit
C/C++ and COBOL compilers	C/C++: IBM XL C/C++ Enterprise Edition V9.0 for AIX (PTF October 2007) or later compatible versions COBOL: Micro Focus 5.1, or other compatible COBOL compiler

Component	Requirement
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0
Non-Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0 (or higher) Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle 12c
SSL certificate authorities	Verisign
Lightweight Directory Access Protocol (LDAP) directory servers	iPlanet Directory Server; needed to retrieve X.509v3 digital certificates for SSL

Additional Notes:

- If you use GAUTHSVER with JDK1.6 or later, please specify the following: `JAVA_OPTS=-Djavax.xml.stream.XMLInputFactory=com.bea.xml.stream.MXParserFactory` in your environment before booting GAUTHSVR.
- When using the Oracle MQ Adapter for Oracle Tuxedo in an AIX version 6.1 64-bit environment, if the `dltmqlnk` command is not executed to remove the symbolic link to the 32-bit WebSphere MQ libraries, you must export the `TUXEDO_LIBPATH_PREPEND` environment variable and set its value to the 64-bit WebSphere MQ libraries path
- When using Oracle, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- When developing CORBA C++ client/server applications, you must use `ifdef` directives to include standard C++ headers or old C headers, otherwise compiler will fail.

The applications use the arco "`OBB_ANSI_CPP`" macro to distinguish the two types of headers.

  - If "`OBB_ANSI_CPP`" is defined, the standard C++ headers are included.
  - If "`OBB_ANSI_CPP`" is undefined, the old C headers are included.

- For Oracle Tuxedo 56-bit, 128-bit, and 256-bit encryption:
  - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).

    LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).

    LLE and SSL support connections between machines and domains.
  - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
  - Oracle Tuxedo CORBA C++ clients support SSL 3.0. SSL connectivity between these clients, and the Oracle Tuxedo ISL/ISH has been certified.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPLanet Directory Server.
- For compiler compatibility:

  Look for documentation from the compiler vendor that provides assurance that the version of compiler you want to use is compatible with the tested version of the compiler. The compatibility assurance must be provided for the following:
  - Source Code:

    C/C++ source code that was compiled and linked using the tested version will compile with the newer version of complier without requiring any changes.
  - Executables and libraries:

    Libraries and executables built using the tested version will run with libraries and executables that are built with the newer version of compilers.

Network Requirements for IBM AIX 6.1 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for IBM AIX 6.1 (64-bit)

For all install sets and components, 449,800KB is the minimum disk space requirement for installation on IBM AIX 6.1 (64-bit) IBM PowerPC systems. This requirement assumes installation of the default components for the selected install set.

Mounting and Unmounting the CD for AIX 6.1 (64-bit)

To mount a CD, examine the file `/etc/filesystems` to determine whether there is a standard place in which to mount a CD. If there is, enter the `mount` command and specify the directory named in the `/etc/filesystems` entry.

For example, to mount a CD when an entry in `/etc/filesystems` specifies `/cd` as the mount point, enter:

```
su
/usr/sbin/mount /cd
```

If `/etc/filesystems` does not contain a CD entry, enter:

```
su
mkdir /cd
/usr/sbin/mount -v cdrfs -r cd_device /cd
```

In the latter command line, *`cd_device`* represents the name of the CD device file, typically `/dev/cd0`.

Alternatively, you can use the System Management Interface Tool (SMIT) to perform the mount. To use SMIT, enter the following:

```
smit mount
```

To unmount the CD, enter the following command:

```
unmount /cd
```

In this command line `cd` represents the mount point.

Tunable Parameters for IBM AIX 6.1 (64-bit)

No IPC configuration is required for AIX. To *check* the value of a kernel tuning parameter (`maxuproc` only) enter the following command: `lsattr -El sys0 -a maxuproc`.

To *change* the value of a kernel tuning parameter (`maxuproc` only) enter the following command: `chdev -l syso -a maxuproc=″,,,″` or can enter `smit chgsys` and select “maximum number of processes allowed per user.”

Oracle Linux 5.6 (64-bit) on x86_64

The following sections list Oracle Tuxedo 12*c* (12.1.1.0.0) requirements for Oracle Linux 5.6 (64-bit) on x86_64 .

Available Oracle Tuxedo 12*c* (12.1.1.0.0) Install Sets for Oracle Linux 5.6 (64-bit)

Installation

- Available Oracle Tuxedo 12*c* (12.1.1.0.0) Install Sets for Oracle Linux 5.6 (64-bit)
  - Full install set
  - Server install set
  - Full client install set
  - ATMI (/WS) client install set
  - CORBA client install set
  - Jolt client install set

  For descriptions of Oracle Tuxedo install sets, see "Install Sets" on page 1-9.
- OTMQ install for Oracle Linux 5.6 (64-bit)

Hardware Requirements for Oracle Linux 5.6 (64-bit)

- x86_64
- 1 GB of RAM minimum
- At least 4MB RAM for each Oracle Tuxedo system server

Software Requirements for Oracle Linux 5.6 (64-bit)

Component	Requirement
OS Version (Patches)	Oracle Linux 5.6 (64-bit)
C/C++ and COBOL compilers	C/C++: gcc/g++ 4.1.2 COBOL: Micro Focus 5.1, or other compatible COBOL compiler.
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0

Additional Notes

- The Kerberos Security Plug-in is not supported on this platform.
- ATMI users need a C, C++ or COBOL compiler.
- CORBA C++ users need a C++ compiler and linker.
- For compiler compatibility:

  Look for documentation from the compiler vendor that provides assurance that the version of compiler you want to use is compatible with the tested version of the compiler. The compatibility assurance must be provided for the following:

  – Source Code:

    C/C++ source code that was compiled and linked using the tested version will compile with the newer version of complier without requiring any changes.

  – Executables and libraries:

    Libraries and executables built using the tested version will run with libraries and executables that are built with the newer version of compilers.

Network Requirements for Oracle Linux 5.6 (64-bit)

- TCP/IP, using the Sockets network interface

Disk Space Requirements for Oracle Linux 5.6 (64-bit)

For all install sets and components, 192,864 KB is the minimum disk space requirement for installation on Oracle Linux 5.6 (64-bit) systems. This requirement assumes installation of the default components for the selected install set.

Mounting and Unmounting the CD for Oracle Linux 5.6 (64-bit)

Mount CD-ROM automatically.

Tunable Parameters for Oracle Linux 5.6 (64-bit)

See sysctl manpage.

Oracle Solaris 10 (64-bit) on SPARC

The following sections list Oracle Tuxedo 12*c* (12.1.1.0.0) requirements for Oracle Solaris 10 (64-bit) systems on SPARC.

Available Oracle Tuxedo 12*c* (12.1.1.0.0) Install Sets for Oracle Solaris 10 (64-bit)

Installation

- Available Oracle Tuxedo 12*c* (12.1.1.0.0) Install Sets for Solaris 10 (64-bit) on SPARC
  - Full install set
  - Server install set
  - Full client install set
  - ATMI (/WS) client install set
  - CORBA client install set
  - Jolt client install set

  For descriptions of Oracle Tuxedo install sets, see “Install Sets” on page 1-9.
- OTMQ install for Oracle Solaris 10 (64-bit) on SPARC)

Hardware Requirements for Oracle Solaris 10 (64-bit) on SPARC

- UltraSPARC

- 1 GB of RAM minimum
- At least 7MB RAM for each Oracle Tuxedo system server

Software Requirements for Oracle Solaris 10 (64-bit) on SPARC

Component	Requirement
OS Version (Patches)	SunOS 5.10 (64-bit)
C/C++ and COBOL compilers	C/C++: Sun Studio 12 COBOL: Micro Focus 5.1, or other compatible COBOL compiler
Java 2 JRE for the Java run-time environment	Tested with Java 2 JRE 1.6.0_10
Java 2 Software Development Kit (SDK) for the Java development environment	Tested with JDK 1.6.0_10
Non Oracle CORBA Java clients	Tested with Java 2 SDK 1.6.0_10 Interface Definition Language (IDL) ORB (run time)
Database for CORBA C++ applications	Tested with Oracle 12c
SSL certificate authorities	Verisign
Lightweight Directory Access Protocol (LDAP) directory servers	iPlanet Directory Server; needed to retrieve X.509v3 digital certificates for SSL
IBM WebSphere MQ for Oracle Tuxedo MQ Adapter	Tested with WebSphere MQ 7.0.0.1

Additional Notes

- When Oracle is used, Programmer/2000 Pro*C/C++ version 10.1.0.3.0 is required to build the Oracle Tuxedo University sample applications.
- ATMI users need a C, C++, or COBOL compiler.

Note: To use Micro Focus 5.1, C/C++ compilers should use Sun Studio 12 or higher.

- CORBA C++ users need a C++ compiler and linker.
- For Oracle Tuxedo 56-bit, 128-bit, and 256-bit encryption:
  - LLE and SSL are available for Oracle Tuxedo ATMI client (/WS) connections to the Oracle Tuxedo Workstation Listener (WSL) or Workstation Handler (WSH).

    LLE and SSL are available for Oracle Jolt client connections to the Oracle Tuxedo Jolt Server Listener (JSL) or Jolt Server Handler (JSH).

    LLE and SSL support connections between machines and domains.
  - The Oracle Tuxedo Internet Inter-ORB Protocol (IIOP) Listener (ISL) and IIOP Handler (ISH) support SSL 3.0 for IIOP connections.
  - Oracle Tuxedo CORBA C++ client supports SSL 3.0. SSL connectivity between these clients, and the Oracle Tuxedo ISL/ISH has been certified.
- To support certificate-based authentication when using SSL, Oracle Tuxedo provides an LDAP-based certificate retrieval mechanism. This retrieval mechanism has been certified for use with the LDAP Directory server included with iPlanet Directory Server.
- For compiler compatibility:

  Look for documentation from the compiler vendor that provides assurance that the version of compiler you want to use is compatible with the tested version of the compiler. The compatibility assurance must be provided for the following:
  - Source Code:

    C/C++ source code that was compiled and linked using the tested version will compile with the newer version of complier without requiring any changes.
  - Executables and libraries:

    Libraries and executables built using the tested version will run with libraries and executables that are built with the newer version of compilers.

Network Requirements for Oracle Solaris 10 (64-bit) on SPARC

TCP/IP using the Sockets network interface.

Disk Space Requirements for Oracle Solaris 10 (64-bit) on SPARC

For all install sets and components, 447,686KB is the minimum disk space requirement for installation on Oracle Solaris 10 (64-bit) SPARC systems. This requirement assumes installation of the default components for the selected install set.

Mounting and Unmounting the CD for Oracle Solaris 10 (64-bit) on SPARC

The Oracle Solaris Volume Management software automatically mounts CDs on `/cdrom/cdrom0/s0`.

It is not necessary to unmount CDs on Oracle Solaris systems. However, it is necessary to issue a command to open the CD reader. To open the CD reader, `cd` to `/` (root) and enter `eject`.

Tunable Parameters for Oracle Solaris 10 (64-bit) on SPARC

Use the `mdb` tool to change the kernel parameters.

The tunable parameters currently set on your system are in the kernel configuration file located in the `/etc` directory (`/etc/`*`system`*).

Platforms Supporting Threads

The following platforms support threads:

- IBM AIX 6.1 (64-bit) on IBM PowerPC
- Oracle Linux 5.6 (64-bit) on x86_64
- Oracle Solaris 10 (64-bit) on SPARC

Note: If threads are not supported on your platform, your application must either: (1) exclude threads or (2) serialize threaded access through all Oracle Tuxedo system calls.

Starting the Oracle Tuxedo Message Queue Administration Console

The following sections provide the system requirements for the Oracle Tuxedo Administration Console and explain how to start and exit the Console.

What is the Oracle Tuxedo Administration Console?

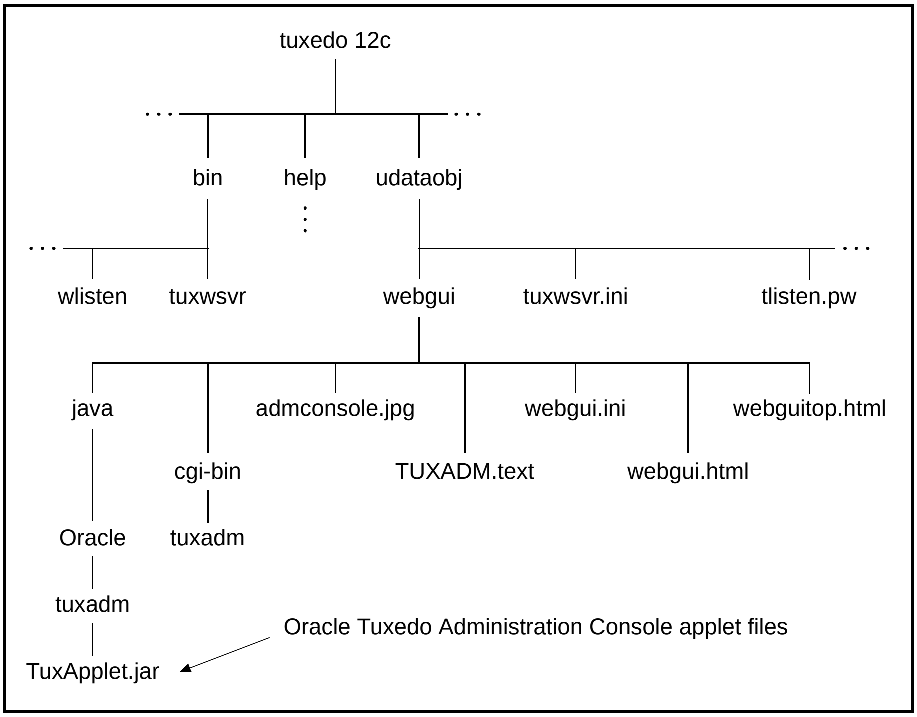
The Oracle Tuxedo Administration Console is a graphical user interface that enables administrators to perform most administration and configuration tasks for ProductName applications. It is implemented as a set of Java applets, which can run on most platforms that support a Java-capable Web browser.

The server-side components of the Oracle Tuxedo Administration Console reside on one of the server machines in a ProductName application. To use the Console, you must enter the URL of the server and download the Java applets.

Administration Console File Tree

During the ProductName software installation, the installer program places most of the directories and files for the Oracle Tuxedo Administration Console in the `webgui` directory, as shown in Figure 0-1.

Figure 0-1 Oracle Tuxedo Administration Console File Tree



The installer program installs the following HTML files:

- An HTML template file named webgui.html, which is used by the common gateway interface (CGI) program tuxadm as the basis for many screens displayed during a Oracle Tuxedo Administration Console session.
- An HTML file named webguitop.html, which displays legal notices and warnings when the Oracle Tuxedo Administration Console is first brought up on the screen.
- The HTML documentation files for the ProductName Administration Console, which are installed in a top-level directory named help.

The installer program installs the class files for the Java applet in the java directory, and installs the tuxadm program in the cgi-bin directory. The installer assigns an alias pathname for tuxadm, which is used by Web clients to access tuxadm. The alias pathname is \cgi-bin on a Windows 2003 Server system and /cgi-bin on a UNIX system.

Server Requirements

The Oracle Tuxedo Administration Console server for ProductName **Product Version:** is supported on the following platforms:

- IBM AIX 5.3 (32-bit) on IBM PowerPC
- IBM AIX 5.3 (64-bit) on IBM PowerPC
- IBM AIX 6.1 (32-bit) on IBM PowerPC
- IBM AIX 6.1 (64-bit) on IBM PowerPC
- Novell SUSE Linux Enterprise Server 10 (64-bit) on x86-64
- Oracle Enterprise Linux 5.0 (64-bit)
- Oracle Solaris 10 (32-bit) on SPARC
- Oracle Solaris 10 (64-bit) on x86-64
- Oracle Solaris 10 (64-bit) on SPARC

Browser Requirements

The encryption level for the Oracle Tuxedo Administration Console is set with the ENCRYPTBITS parameter in the webgui.ini file for the Console. In the following webgui.ini file, the encryption level is set to 56 bits.

```
# Web GUI initialization file.
# Created Sep 12, 2002 2:02:07 PM by Oracle software installation program.
#
TUXDIR=C:\oracle\tuxedo12c
INIFILE=C:\oracle\tuxedo12c\udataobj\webgui\webgui.ini
NADDR=//foo5:4003
DEVICE=/dev/tcp
CODEBASE=/java
DOCBASE=/doc
SNAPDIR=C:\oracle\tuxedo12c\udataobj\webgui\java\snapshot
SNAPBASE=/java/snapshot
ENCRYPTBITS=56
```

The ENCRYPTBITS parameter specifies the strength of encryption used in communication between the GUI applet and the Oracle Tuxedo Administration Console server. The ENCRYPTBITS parameter can be set to 0, 56, or 128. The default value is 128.

Table 0-1 lists the platforms and browsers supported for ProductName **Product Version:** when the Oracle Tuxedo Administration Console is configured for 56-bit or 128-bit encryption.

Table 0-1 Platforms and Browsers

On this platform . . .	Oracle tested these browsers . . .	Using Java plug-in . . .
IBM AIX 5.3 (32-bit) on IBM PowerPC		
Oracle Solaris 10(32-bit) on SPARC		

If the Oracle Tuxedo Administration Console is configured with no encryption (0-bit encryption), Internet Explorer 6.0 is supported on any Windows platform regardless of the Java plug-in version.

Hardware Requirements

To run the Oracle Tuxedo Administration Console, you need a color display device that meets the following requirements:

- Resolution: 800 by 600 or more is required; 1024 by 768 or more is recommended.
- Colors: A minimum of 256 colors is recommended.

Setting Up Your Environment for the Oracle Tuxedo Administration Console

To run the Oracle Tuxedo Administration Console, first set up your environment as described in "Setting Up Your Environment" on page 6-8. Ensure that your `TUXDIR`, `WEBJAVADIR`, and `PATH` environment variables are set correctly. Then set up the following two server processes:

- `tuxwsvr(1)`

  A Web server provided with the ProductName system software. You are not required to use this server; you may, if you prefer, use your own commercial Web server.

- `wlisten(1)`

  A server required to administer the Oracle Tuxedo Administration Console. It must be run on the `MASTER` machine in a Tuxedo multi-machine configuration.

After starting the `tuxwsvr` and `wlisten` server processes, you can start the Oracle Tuxedo Administration Console to monitor the `tuxwsvr` server and the ProductName application.

Starting tuxwsvr

The tuxwsvr process is a Web server process that can be used to support the ProductName Administration Console GUI by customers who do not have a commercial Web server or a public-domain Web server on the machine on which the ProductName Web GUI processes are running. The tuxwsvr process places itself in the background when invoked unless otherwise specified, and continues running until the machine shuts down or the tuxwsvr process is killed using an operating system command.

Table 0-2 lists the commands for starting tuxwsvr.

Table 0-2 Starting tuxsvr

To Start Tuxwsvr on This Platform . . .	Enter the Following Command . . .
UNIX system	tuxwsvr -l //*machine*:*port* -i $TUXDIR/udataobj/tuxwsvr.ini

The ProductName installer program creates the tuxwsvr.ini file, an example of which is shown below:

```
# tuxwsvr initialization file.
# Created Sep 12, 2008 2:02:07 PM by Oracle software installation program.
#
CGI    /cgi-bin     C:\oracle\tuxedo12c/udataobj/webgui/cgi-bin
HTML   /java    C:\oracle\tuxedo12c/udataobj/webgui/java
HTML   /doc     C:\oracle\tuxedo12c/help
HTML   /        C:\oracle\tuxedo12c/udataobj/webgui
```

Usually, you do not need to edit this file, but under certain circumstances, you may want to do so. For example, you may decide to move your Java files to a non-default directory. In that case, you would need to edit the pathnames in the initialization file appropriately. For details, see reference page tuxwsvr(1) in *Oracle Tuxedo Command Reference*.

Starting wlisten

The wlisten process is a listener process that receives incoming connections from Web GUI applets and starts a Web GUI gateway process (wgated). All wlisten options are taken from an initialization file that is specified by the -i option. If the -i option is not given $TUXDIR/udataobj/webgui/webgui.ini is used as the default initialization file on a UNIX system.

To start `wlisten`, follow these steps:

1. Check the `webgui.ini` file to make sure that the default values assigned to the parameters during installation are appropriate. If they are not, make the necessary changes.

   For example, on a machine called `foo5`, the default port assigned to `wlisten` is 4003. To run `wlisten` with port 6060, edit the `NADDR` parameter as follows:

   ```
   NADDR=//foo5:6060
   ```

   For details about other parameters in the `webgui.ini` file, see reference page `wlisten(1)` in *Oracle Tuxedo Command Reference*.

2. Enter the following command:

   ```
   wlisten
   ```

Starting the Oracle Tuxedo Administration Console

The `tuxadm` process is a CGI process used to initialize the Web GUI from a browser. As shown in the "Synopsis" section of reference page `tuxadm(1)`, this program can be used only as a location, or URL from a Web browser; normally it is not executed from a standard command-line prompt. Like other CGI programs, `tuxadm` uses the `QUERY_STRING` environment variable to parse its argument list.

To start the Oracle Tuxedo Administration Console, complete the following four-step procedure:

1. Start the browser.

2. Enter the following URL:

   `http://`*`machine`*`:`*`port`*`/webguitop.html`

   For example: `http://foo5:4003/webguitop.html`

   Use of this URL depends on the following assumption: You are using `tuxwsvr` with the `tuxwsvr.ini` file. If you are using a commercial browser on the default port (8080), you can use a URL such as `http://ctomsn:8080/webguitop.html`.

   The Oracle Tuxedo Administration Console entry page-displays.

3. To start the Oracle Tuxedo Administration Console, select the Click Here to Run the Oracle Tuxedo Administration Console prompt at the bottom of the screen. The Login window displays.

4. Enter your login name and password in the appropriate fields, and select LOGIN. The password must correspond to one of the entries in the system-encrypted `tlisten.pw` file.

The main window of the Oracle Tuxedo Administration Console displays.

Table 0-3 contains instructions for accessing additional information about the Oracle Tuxedo Administration Console main window.

Table 0-3 Accessing Information About the Oracle Tuxedo Administration Console Main Window

If. . .	Then. . .
The main window is displayed and you want to start working with the GUI	See "Administration Console Tutorial" in *Oracle Tuxedo Administration Console Online Help*.
The main window is displayed and you want to read a description of it	See "A Tour of the Main Window" in *Oracle Tuxedo Administration Console Online Help*.
The main window is not displayed and the `Connect Failed` error message is displayed	1. On the Administration Console *server* machine, verify that the `wlisten` process is running. On a UNIX machine, for example, enter the `ps` command. 2. If `wlisten` is not running, open the `webgui.ini` file and, in the line `NADDR=//foo5:4003`, replace the port number (4003) with a valid port number. 3. Enter `wlisten` again: • `$ wlisten -i $TUXDIR/udataobj/webgui/webgui.ini` (on UNIX) 4. Check that the `tuxwsvr` process is running at the port specified in the URL. 5. Verify the password. It must match one of the entries in the `tlisten.pw` file. 6. Return to step 1.

Limitation(s)

The Oracle Tuxedo Administration Console has not been updated to support any new features introduced after ProductName release 7.1.

Exiting the ProductName Administration Console

To exit the Oracle Tuxedo Administration Console, choose Domain → Exit from the menu bar. This menu option closes the current domain and exits the Oracle Tuxedo Administration Console applet.

File and Database Management and Disk Space Allocation

The following sections describe Oracle Tuxedo file and database management and provide guidelines for allocating disk space for a Oracle Tuxedo application:

- Introduction
- How the Oracle Tuxedo System Manages Files
- Arranging for Raw Disk Space
- How the Oracle Tuxedo Filesystem Is Organized
- Space for Queue Spaces
- Space for Application Servers

Introduction

For the most part, the following discussions apply to both Windows and UNIX systems except for the guidelines to repartition hard disk devices. On a Windows system, input/output (I/O) is buffered by default, but Oracle Tuxedo sets a certain system-level flag to change the default. Thus, all I/O for Oracle Tuxedo processes is *unbuffered*, meaning that you do not need to make any special disk-space arrangements on a Windows system.

How the Oracle Tuxedo System Manages Files

The Oracle Tuxedo system provides a facility called the Disk Management Interface (DMI), which manages logical files within a single disk device or set of devices. The DMI performs tasks such as storing binary configuration tables and the transaction log. You can use it to create, initialize, or destroy entries in the Oracle Tuxedo filesystem. To access the DMI, use the `tmadmin(1)`administrative commands described in the *Oracle Tuxedo Command Reference*.

There are two ways that the logical files managed by the DMI can be stored physically:

- Stored on an operating-system (OS) filesystem
- Stored on disk space, set aside for the Oracle Tuxedo system, that is outside the control of all OS filesystems

Oracle Tuxedo files reside on device special files in the designated space and are managed directly by the DMI disk management software. The DMI supports the notion of a Oracle Tuxedo filesystem distinct from any OS filesystem.

Space outside the OS filesystem is usually referred to as *raw disk space*. Not only is I/O faster when done by system calls reading directly from and writing directly to device special files on raw disks, but a physical `write()` occurs right away. When using an OS filesystem, Oracle Tuxedo cannot predict or control the precise moment at which a `write()` is done. When using raw disk space, however, Oracle Tuxedo has accurate control of the write operation, which is particularly important for entries in the Oracle Tuxedo transaction log. Also, when multiple users are accessing the system, being able to control the write operation is important for assuring database consistency.

Arranging for Raw Disk Space

If you decide to use raw disk space for your Oracle Tuxedo application, and you are using a UNIX system, you may find that the hard disk devices on your system are fully allocated to filesystems such as `/` (root) and `/usr`. If that is the case, you must repartition your hard disk device in order to set aside some partitions for use as non-OS filesystems. For repartitioning instructions, refer to the system administration documentation for your platform.

How the Oracle Tuxedo Filesystem Is Organized

A Oracle Tuxedo filesystem has a Volume Table of Contents (VTOC), which lists the files residing on the devices named in the Universal Device List (UDL). The UDL contains information about the location of the physical storage space for Oracle Tuxedo system tables.

In a Oracle Tuxedo application, all system files might be stored together on the same raw disk slice or OS filesystem. While it is possible to use regular OS filesystem files for the configuration tables, we strongly recommend that you store the transaction log, TLOG, on a raw disk device.

Because the TLOG seldom needs to be larger than 100 blocks (51200 bytes assuming 512-byte blocks), and because disk partitions are always substantially larger than 100 blocks, it may make sense to use the same device for both the configuration files and the TLOG.

Listing C-1 shows a sample VTOC and UDL diagram for a bankapp (sample application) configuration on a single machine.

Listing C-1 VTOC and UDL Diagram

```
Output based on setting FSCONFIG=$TUXCONFIG, and invoking tmadmin:

No bulletin board exists. Entering boot mode.

> livtoc
Volume Table of Contents on /usr2/bank/tuxconfig:
0: VTOC: Device 0 Offset 0 Pages 7
1: UDL: Device 0 Offset 7 Pages 28
2: _RESOURCE_SECT:    Device 0 Offset  35 Pages 6
3: _MACHINES_SECT:    Device 0 Offset  41 Pages 40
4: _GROUPS_SECT:      Device 0 Offset 141 Pages 100
5: _SERVERS_SECT:     Device 0 Offset 241 Pages 150
6: _SERVICES_SECT:    Device 0 Offset 391 Pages 60
7: _ROUTING_SECT:     Device 0 Offset 451 Pages 100
8: _NETWORK_SECT:     Device 0 Offset 551 Pages 20
9: _MIBPERMS_SECT:    Device 0 Offset 571 Pages 2
10: _NETGROUPS_SECT:  Device 0 Offset 573 Pages 2
11: _INTERFACES_SECT: Device 0 Offset 575 Pages 10

# If the TLOG is stored on the same device, there will be an
# entry something like:

12: TLOG:             Device 0 Offset 585 Pages 100
```

The Oracle Tuxedo application administrator must make sure raw disk slices are available as needed on each node participating in an application. Table C-1 lists the size of each element in the Oracle Tuxedo filesystem.

Table C-1 Size of Oracle Tuxedo System Tables

Entity	512-Byte Pages (Blocks)
VTOC	7
TUXCONFIG	550
TLOG	100 (default)
UDL	28
TOTAL	685

The amount of space required for the TUXCONFIG file must be larger if there are more entries in the configuration file, UBBCONFIG, than in the bankapp sample application. The administrator is encouraged to allocate additional space for dynamic reconfiguration and growth of the application. The default block size assumed by the crdl subcommand of tmadmin is 1000 blocks (512000 bytes assuming 512-byte blocks), which should be adequate for the initial installation.

Space for Queue Spaces

If your Oracle Tuxedo application uses /Q for store-and-forward queue management, your queue space can be listed in the same UDL as the one used to store the TUXCONFIG file and the TLOG, and managed by the Oracle Tuxedo VTOC.

Space for Application Servers

As you are calculating your space requirements for the Oracle Tuxedo system, you should also consider the requirements of the servers that perform the work of the application. These requirements are specified by the application; they are unrelated to the requirements for the Oracle Tuxedo system itself (unless otherwise specified).

IPC Resource Configuration on a UNIX System

The following sections describe the interprocess communication (IPC) parameters on a UNIX system and provide guidelines for configuring them:

- Parameter Sets Controlling IPC Resources
- Shared Memory
- Semaphores
- Message Queues and Messages
- Other Kernel Tunables

Parameter Sets Controlling IPC Resources

On a UNIX system, the Oracle Tuxedo system uses the IPC resources provided by the UNIX operating system, which are controlled by the tunable parameters listed in Table D-1.

Table D-1 Tunable Parameters

Tunable Parameters Starting with This Prefix . . .	Control the . . .
SHM	Amount of shared memory
SEM	Number of semaphores
MSG	Size of message queues and messages

The settings for these parameters are application-dependent. Most UNIX systems are shipped with default values that are too low for a Oracle Tuxedo application.

Because the IPC parameters vary across different versions of the UNIX system, the descriptions provided in the following sections are generic. For the exact parameter names and defaults for each platform and for information on how to change parameter values, see "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1. If you change a parameter value, you will need to rebuild the kernel and reboot the operating system, using standard administrative tools. Consult your operating system administrator or the system administrator's guide for your platform for details.

If your Oracle Tuxedo application is distributed, the minimum IPC resources must be available on every UNIX platform participating in the application.

Shared Memory

In the Oracle Tuxedo environment, shared memory is used for the bulletin board and the control table of the workstation listener (WSL) and the IIOP listener (ISL) processes. An application may also use shared memory for its own purposes.

The following shared memory parameters may need to be adjusted:

`SHMMAX`

Maximum size, in bytes, of a shared memory segment. This number represents the largest shared memory segment that can be allocated. A process can, however, attach to more than one segment of size `SHMMAX`.

`SHMSEG`

Maximum number of shared memory segments per process. For a given configuration, the maximum amount of shared memory to which a process can attach is the product (in bytes) of `SHMMAX * SHMSEG`. A value between 6 and 15 should be adequate.

`SHMMNI`

Maximum number of shared memory identifiers in the system. The Oracle Tuxedo system requires one identifier per bulletin board and an additional identifier for each workstation listener (WSL) and IIOP listener (ISL) that is running.

`SHMMIN`

Minimum size, in bytes, of shared memory segment. This parameter should always be set to 1.

Semaphores

Every process that participates in a Oracle Tuxedo application requires a *semaphore*. A semaphore is a hardware or software flag used to prevent processes from accessing the same shared memory space at the same time. When a process has control of a shared memory resource, all other processes are locked out of the shared memory resource until the process releases the resource.

When the Oracle Tuxedo application is booted, the underlying Oracle Tuxedo system checks the number of semaphores configured in the operating system. If the configured number is not high enough, the boot fails.

The following semaphore parameters may need to be adjusted:

SEMMNS

Maximum number of semaphores in the system. The minimum requirement for SEMMNS is

```
MAXACCESSERS - MAXWSCLIENTS + 13
```

where MAXACCESSERS is the maximum number of Oracle Tuxedo system processes on a particular machine (including servers and native clients) and MAXWSCLIENTS is the maximum number of Oracle Tuxedo remote clients. Both of these parameters are specified in the UBBCONFIG file for the application. For more information about UBBCONFIG, see “Creating the Configuration File” in *Setting Up an Oracle Tuxedo Application* or UBBCONFIG(5) in the *File Formats, Data Descriptions, MIBs, and System Processes Reference*.

SEMMNI

Maximum number of active semaphore sets.

SEMMSL

Maximum number of semaphores per semaphore set. SEMMNI and SEMMSL are commonly chosen so that their product equals SEMMNS. The Oracle Tuxedo system does not perform semaphore operations on semaphore sets; however, it attempts to allocate as many semaphores per semaphore set as possible.

SEMMAP

Size of the control map used to manage semaphore sets. SEMMAP should be equal to SEMMNI.

SEMMNU

Number of undo structures in the system. Because an undo structure is needed for each process that can access the bulletin board, SEMMNU must be at least as large as SEMMNS. (The UNIX operating system uses undo structures to unlock semaphores held by processes that die unexpectedly.)

`SEMUME`

Maximum number of undo entries per undo structure. The value 1 suffices.

Message Queues and Messages

The Oracle Tuxedo system uses UNIX system messages and message queues for client/server communication. Examples of such messages are service requests, service replies, conversational messages, unsolicited notification messages, administrative messages, and transaction control messages.

Every Multiple Servers, Single Queue (MSSQ) set of servers and every individual server has a message queue for receiving requests. Every client has its own queue for receiving replies. Servers that specify the `REPLYQ` parameter also get individual reply queues.

The adjustment of kernel message parameters is important to the proper tuning of an application. Inappropriate values can lead to an inability to boot, or to severe performance degradation.

Several message queue parameters are available to define various characteristics of the queue space, as indicated in Table D-2.

Table D-2 Message Que Parameters

This Parameter. . .	Specifies the . . .
`MSGTQL`	Total number of outstanding messages that can be stored by the kernel
`MSGMNB`	Total number of bytes that can be stored on one queue
`MSGMAX`	Maximum size of an individual message
`MSGSEG`	Total number of message segments that can be outstanding at one time
`MSGSSZ`	Size of each segment

If the limit specified by any of these parameters is exceeded, then a *blocking condition* occurs. There is one exception to this rule: `MSGMAX`. Messages that exceed 75 percent of `MSGMNB`, or that are larger than `MSGMAX`, are placed in a UNIX file. A very small message containing the filename is then sent to the recipient. Because this mode of operation results in a severe reduction in performance, we strongly recommend that you avoid it.

What Is Application Deadlock?

An application deadlock can result if every process is blocked while trying to send a message. For example, when clients fill up the message space with requests, servers that are trying to send replies are blocked. Therefore, no server can read a message and a deadlock results. Occasionally, timeouts can break a deadlock, but no useful work will have been done.

A client that sends its requests with the `TPNOREPLY` flag is especially troublesome. This practice can fill either individual queues or the system message space, depending on the size of the messages. Such applications may have to implement their own flow control to limit the number of outstanding messages.

To summarize, if clients or servers are blocking on their send operations (requesting services or sending replies), there is potential for trouble. It is usually no problem, though, for a single server request queue to remain full, as long as there is space in the system for more messages on other queues.

Performance Implications of Blocking Conditions

There are performance implications to queue blocking conditions, both on the sending side and the receiving side. When waking up blocked processes, the UNIX operating system wakes up all the processes blocked on a particular event, even if only one can proceed. The other processes go back to sleep. This process scheduling overhead can be expensive.

For example, on an empty server request queue on which more than one server (MSSQ) resides, an arriving message wakes up all the idle (blocked) servers on that queue. In the case of a full server request queue, as each request is read by a server, the system wakes up all the blocked clients. Depending on the size of the messages, zero or more clients can place messages on the queue. The rest go back to sleep. Because there may be hundreds of clients in the system, the mass wakeup of all of these clients every time a service request is processed can severely degrade performance.

Tunable Message Parameters

A properly tuned system rarely fills its queues. Enough slack should be left in the queues to handle the natural variability of the message flow. No exact settings can be recommended. Tuning is very application dependent. The UNIX `ipcs(1)` command provides a snapshot of the queues so you can determine whether they are full. You can try setting the `TPNOBLOCK` flag when sending requests. If you do, clients can tell when queues are full, and they can slow down a bit. It might help to increase the scheduling priority of servers with full request queues.

The following message parameters may need to be adjusted:

`MSGMNI`

Number of unique message queue identifiers. Each process participating in a Oracle Tuxedo application on a particular machine typically needs at least one message queue. This number is reduced if MSSQ sets are used, which means that multiple server processes share a single queue. For transaction processing, count an additional queue per server group for transaction manager server (TMS) processes. Thus, the minimum requirement for `MSGMNI` can be determined by the following formula:

```
MSGMNI = MAXACCESSERS + 7
+ (number of servers with REPLYQ)
+ (number of MSSQ sets)
- (number of servers in MSSQ sets)
```

`MSGMAX`

Maximum message size in bytes. `MSGMAX` must be big enough to handle any Oracle Tuxedo application running on this machine.

`MSGMNB`

Maximum message queue length in bytes. This number must accommodate the total size of all messages that are on a queue and have not been taken off by the associated processes. The minimum value for `MSGMNB` is the value of `MSGMAX`. Messages longer than 75% of `MSGMNB` are sent to a file instead of a message queue—a situation that should be avoided because it severely degrades performance.

`MSGMAP`

Number of entries in the control map used to manage message segments. The value of `MSGMAP` should be the number of message segments (specified in `MSGSEG`).

`MSGSSZ`

Size, in bytes, of a message segment. A message can consist of several such segments. The value of `MSGSSZ` should be such that a multiple of `MSGSSZ` is equal to the size (including the Oracle Tuxedo system header) of the most commonly sent message. By dividing messages into segments in this way, you can avoid wasting space.

`MSGSEG`

Number of message segments in the system.

`MSGTQL`

Total number of outstanding messages that can be stored by the kernel. This is the maximum number of unread messages at any given time.

Other Kernel Tunables

Experience with the Oracle Tuxedo system has shown that some other UNIX system tunables may need to be set to higher values. These parameters are very application dependent and do not apply to all applications. "Oracle Tuxedo Message Queue 12c (12.1.1.0.0) Platform Data Sheets" on page A-1 includes information on the defaults for each platform and instructions for changing them.

`ULIMIT`

Maximum file size. `ULIMIT` needs to be large enough so that you can install the Oracle Tuxedo system and build servers. We recommend 4 MB.

`NOFILES`

Maximum number of open files per process. A Oracle Tuxedo server requires a minimum of four file descriptors.

`MAXUP`

Maximum number of processes per non-superuser. The Oracle Tuxedo system processes—servers and administrative processes—run with the `UID` specified in the application's `UBBCONFIG` file. `MAXUP` needs to be large enough to allow all of these processes to run.

`NPROC`

Maximum number of processes (system wide).

`NREGION`

Number of region table entries to allocate. Most processes have three regions: text, data, and stack. Additional regions are needed for each shared memory segment and each shared library (including text and data) that is attached. However, the region table entry for the text of a "shared text" program is shared by all processes executing that program. Each shared memory segment attached to one or more processes uses another region table entry.

`NUMTIM`

Maximum number of STREAMS modules that can be pushed by the Transport Layer Interface (TLI). A typical default value is 16; we recommend setting this parameter to at least 256.

`NUMTRW`

The number of TLI read/write structures to allocate in kernel data space. A typical default value is 16; we recommend setting this parameter to at least 256.